Segunda Apresentação - Engenharia de Software 2

Bruna Zamith, 628093
Leonardo Utida, 628182
Leonardo Tavares, 628174

April 26, 2019

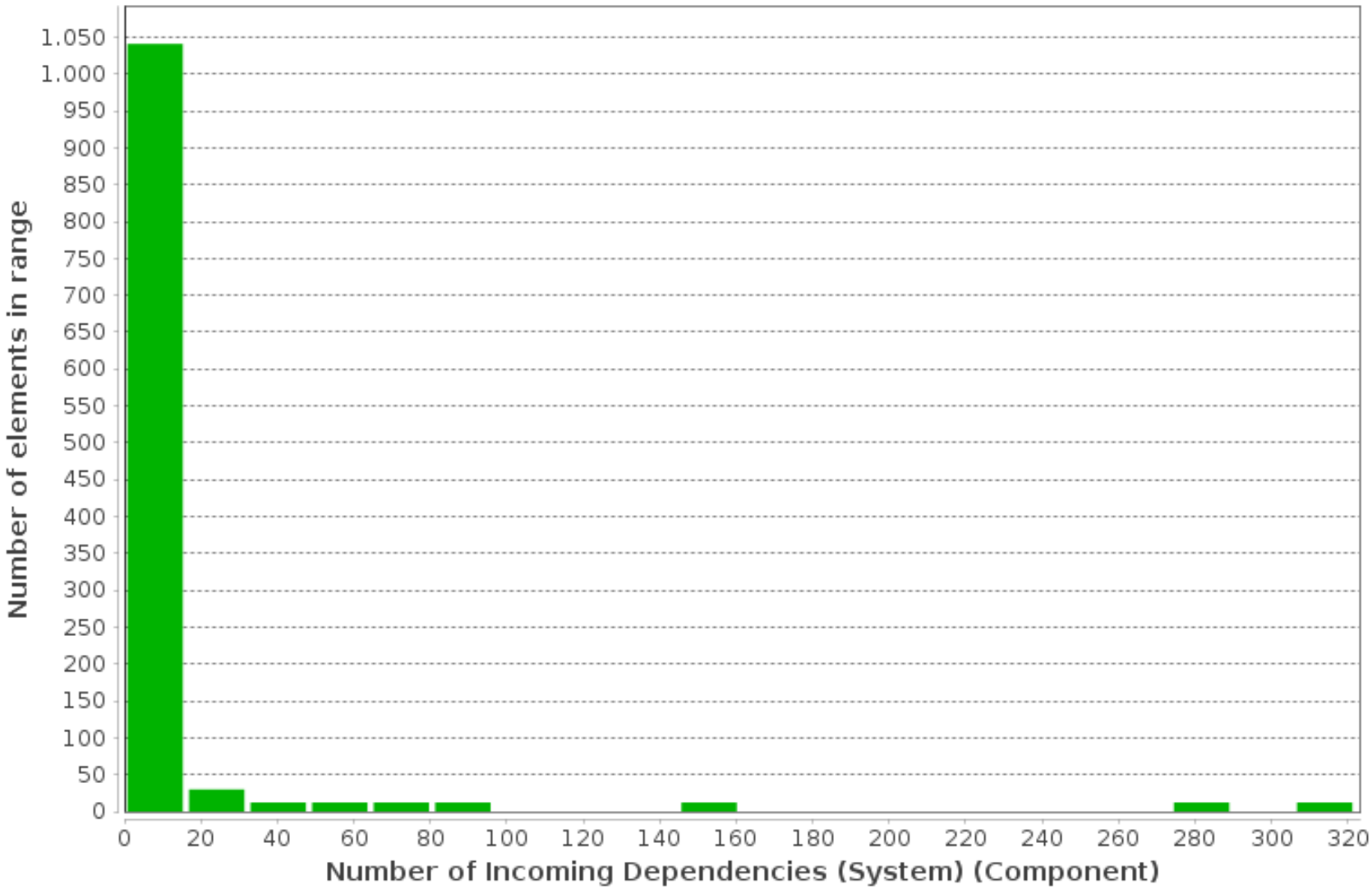
Sistemas

1. Jenkins
2. Apache Flink
3. Eclipse Che

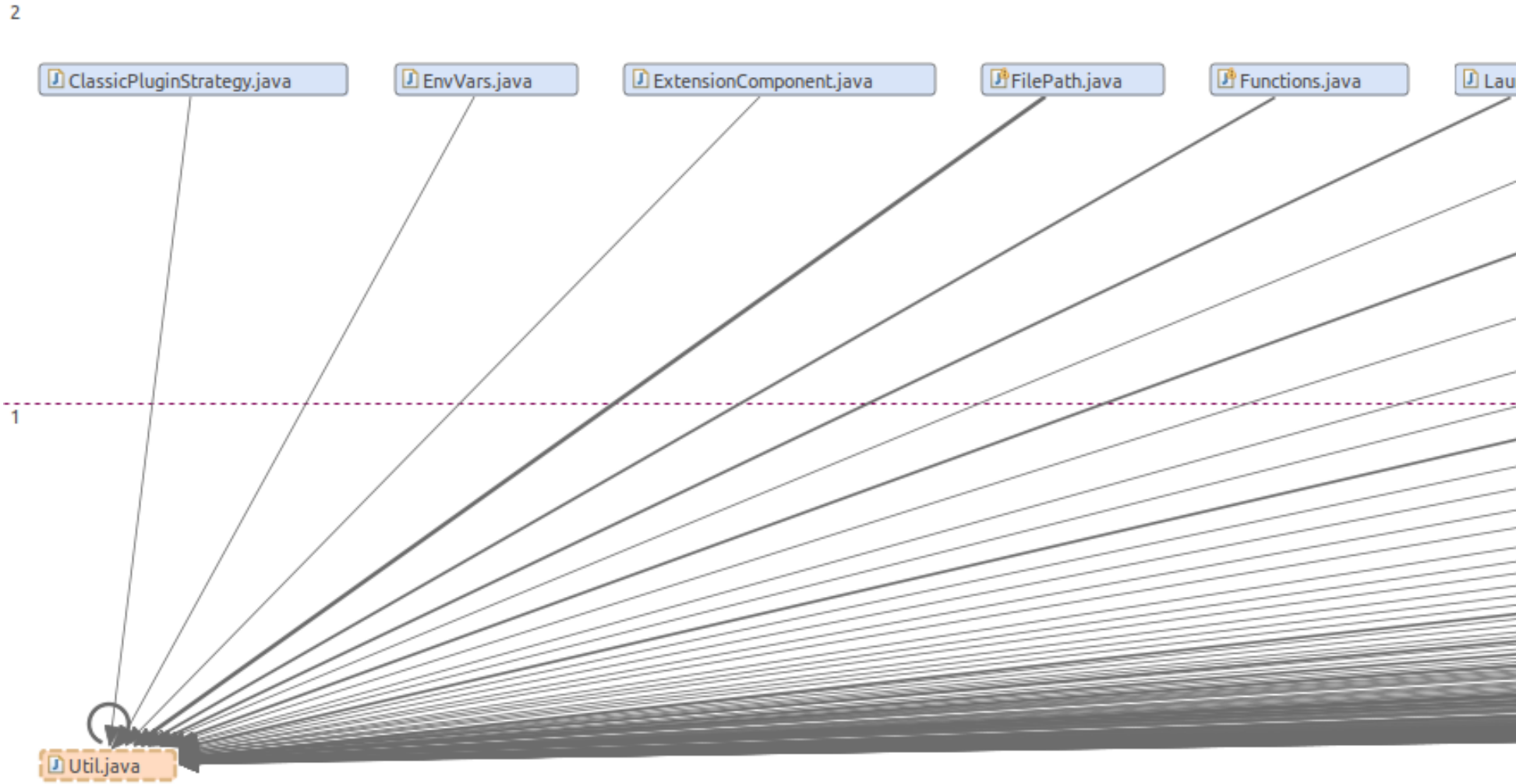
1. Quais as cinco classes em que a métrica de “Number of Incoming Dependencies” é maior?

Classe	**N.I.D**
/jenkins/model/Jenkins.java	322
/hudson/Extension.java	287
/hudson/model/Descriptor.java	153
/hudson/ExtensionList.java	148
/hudson/Util.java	146
/hudson/ExtensionPoint.java	146

Jenkins


Number of elements in range
1.050
1.000
950
900
850
800
750
700
650
600
550
500
450
400
350
300
250
200
150
100
50
0
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
Number of Incoming Dependencies (System) (Component)

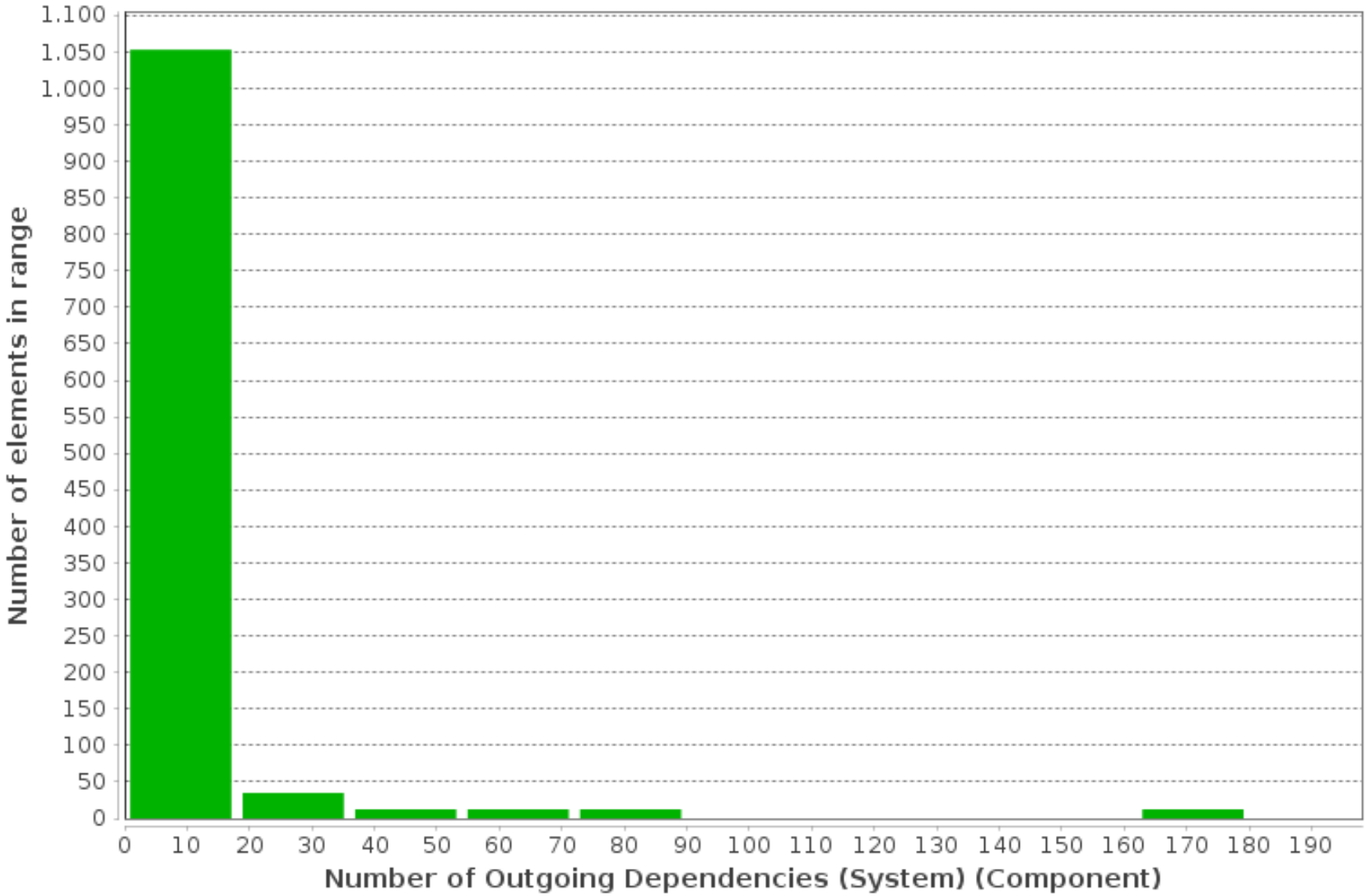
Jenkins


2
ClassicPluginStrategy.java
EnvVars.java
ExtensionComponent.java
FilePath.java
Functions.java
Lau
1
Util.java

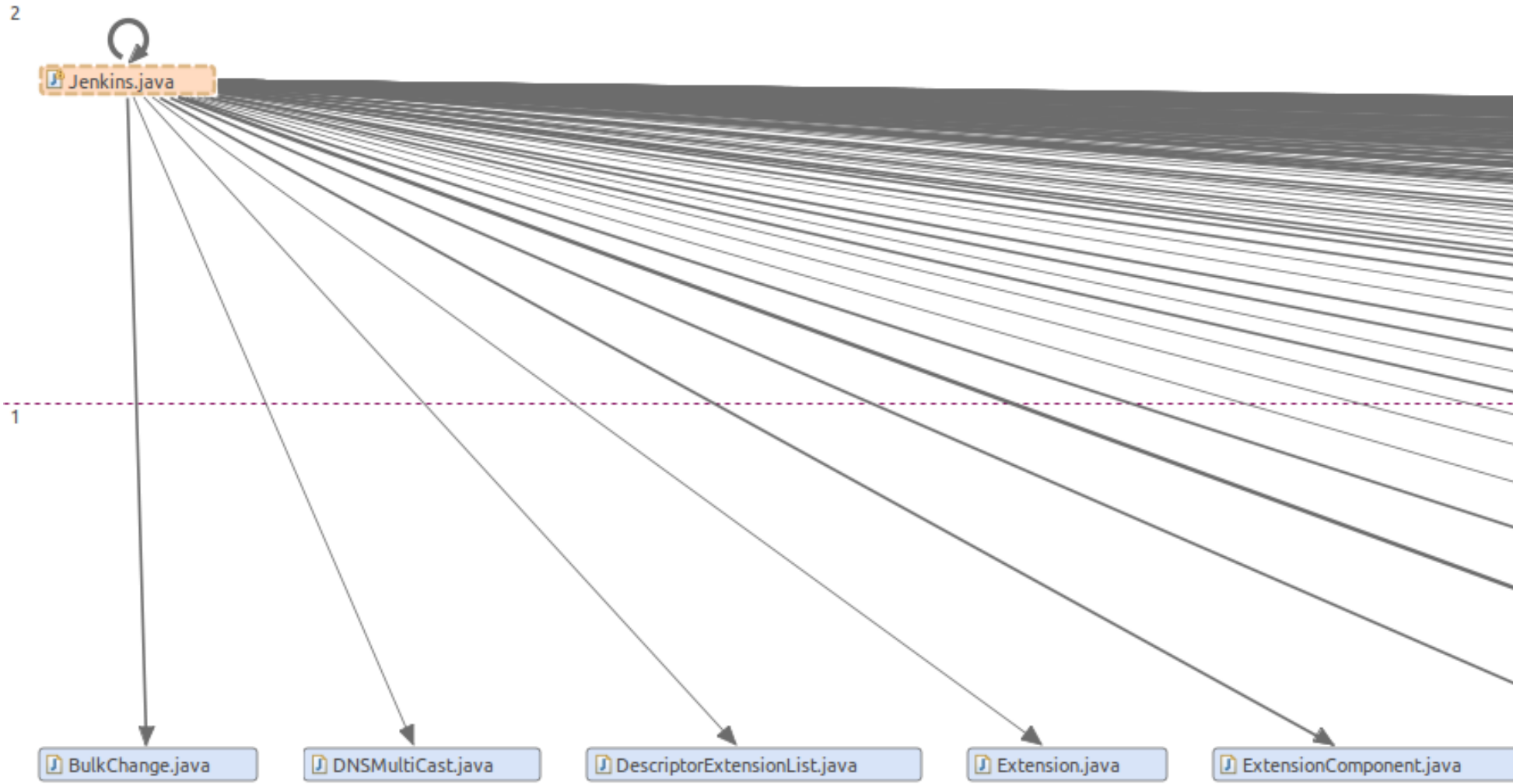
2. Quais as cinco classes em que a métrica de “Number of Outgoing Dependencies” é maior?

Classe	**N.O.D**
/jenkins/model/Jenkins.java	180
/hudson/model/AbstractProject.java	84
/hudson/model/Run.java	76
/hudson/Functions.java	74
/hudson/model/Job.java	72

Jenkins


Number of elements in range
1.100
1.050
1.000
950
900
850
800
750
700
650
600
550
500
450
400
350
300
250
200
150
100
50
0
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190
Number of Outgoing Dependencies (System) (Component)

Jenkins


2
Jenkins.java
1
BulkChange.java
DNSMultiCast.java
DescriptorExtensionList.java
Extension.java
ExtensionComponent.java

3. Mostrar quais são os tipos de dependências existentes as classes anteriores.

Incoming

	Tipos de Dependência
Jenkins	Aggregated, Field, Read Field,Static Method Call
Extension	Aggregated, Has Annotation
Descriptor	Aggregated, Extends, Local Variable, Parameter, Returns, Stastic Method Call,Throws, Type Argument, Virtual Method Call
ExtensionList	Aggregated, Returns, Static Method Call, Virtual Method Call
Util	Aggregated, Read Field, Static Method Call
ExtensionPoint	Aggregated, Extends, Implements

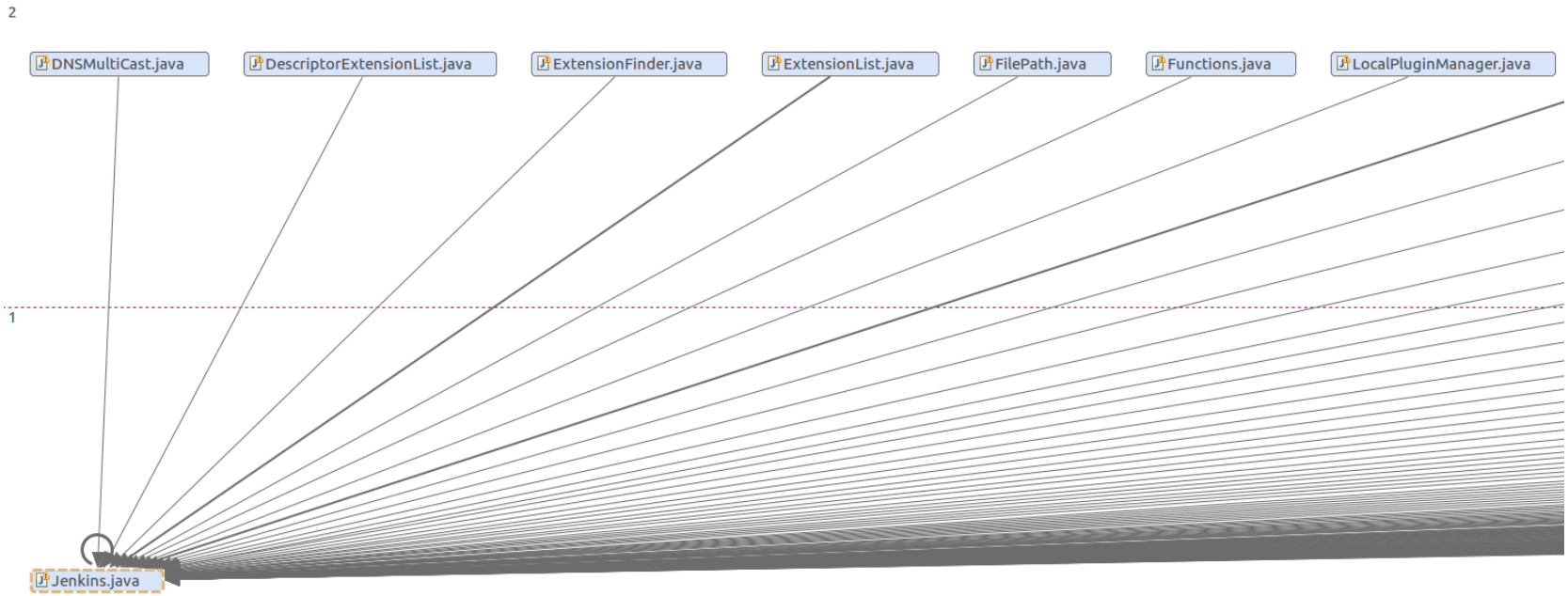
3. Mostrar quais são os tipos de dependências existentes as classes anteriores.

Outgoing

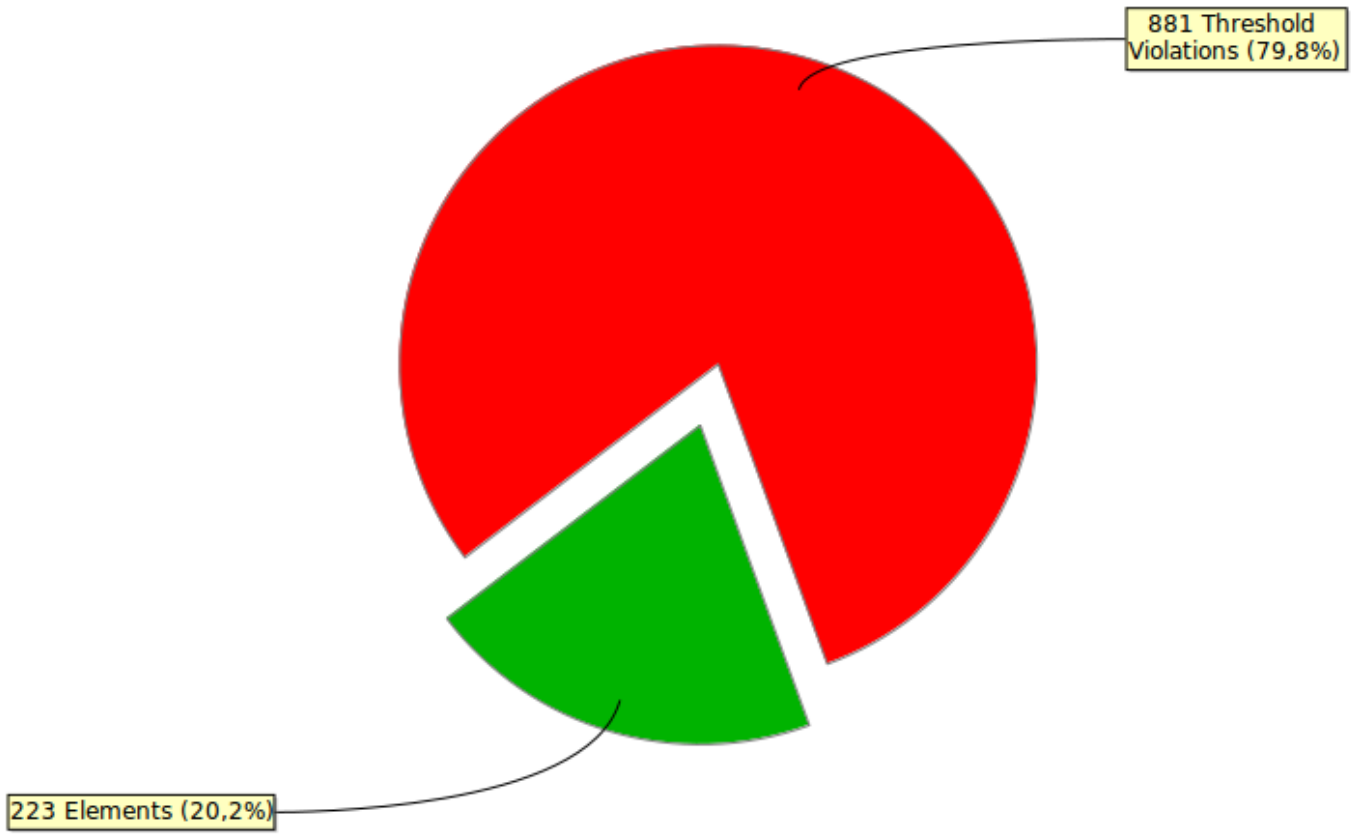
	Tipos de Dependência
Jenkins	Aggregated, Catch, Extends, Field, Has Annotation, Implements, Instance Of, Interface Method Call, Local Variable, New Array, Parameter, Read Field, Returns, Static Method Call, Throws, Uses, Virtual Method Call
AbstractProject	Agreggated, Annotation Value, Catch, Extends, Field, Has Annotation, Implements, Interface Method Call, Parameter, Returns, Static Method Call, Type Argument, Virtual Method Call
Run	Aggregated, Has Annotation, Implements, Interface Method Call, Local Variable, Read Field, Returns, Static Method Call, Throws, Type Argument, Uses, Virtual Method Call
Functions	Aggregated, Catch, Implements, Instance Of, Local Variable, Read Field, Read Field Inline, Static Method Call, Throws, Type Argument, Virtual Method Call
Job	Aggregated, Catch, Extends, Has Annotation, Implements, Instance Of, Interface Method Call, Local Variable, Parameter, Read Field, Returns, Static Method Call, Throws, Virtual Method Call

4. Quais daquelas dependências são do tipo Associação?

- Quando uma classe possui um atributo do tipo da outra.
  - Field ou New
  - Pode ser contido pelo Aggregated



5. Averiguar se há classes cujas “outgoing dependencies” são zero.



Arithmetic average: 5,58
Standard deviation: 9,75
Median: 4,00

Min. value: 0
Max. value: 180

Jenkins

6. Utilize a métrica Abstractness para identificar padrões de projetos.

Element [80]	Abstractness (System)
hudson.security.captcha	1,00
lib.jenkins	1,00
lib	1,00
jenkins.scm	0,50
jenkins.tasks	0,50
hudson.scm	0,43
hudson.tools	0,42
hudson.markup	0,40
hudson.util.jna	0,39
hudson.cli.declarative	0,38
jenkins	0,34
hudson.model.queue	0,34
jenkins.model.queue	0,33
jenkins.model.identity	0,33
jenkins.util.groovy	0,33
jenkins.util.io	0,33
hudson.model.labels	0,33
jenkins.telemetry	0,33
hudson.init	0,33
hudson.model.listeners	0,31
hudson.util.spring	0,30
jenkins.mvn	0,29

7. Design Patterns

- **Singleton**

```
    protected Jenkins(File root, ServletContext context) throws IOException,
InterruptedException, ReactorException {
        this(root, context, null);
    }

    /**
     * @param pluginManager
     *      If non-null, use existing plugin manager.  create a new one.
     */
    @edu.umd.cs.findbugs.annotations.SuppressFBWarnings({
        "SC_START_IN_CTOR", // bug in FindBugs. It flags UDPBroadcastThread.start()
call but that's for another class
        "ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD" // Trigger.timer
    })
    protected Jenkins(File root, ServletContext context, PluginManager pluginManager)
throws IOException, InterruptedException, ReactorException {
        oldJenkinsJVM = JenkinsJVM.isJenkinsJVM(); // capture to restore in cleanUp()
        JenkinsJVMAccess._setJenkinsJVM(true); // set it for unit tests as they will
not have gone through WebAppMain
        long start = System.currentTimeMillis();
        STARTUP_MARKER_FILE = new FileBoolean(new File(root, ".lastStarted"));
        // As Jenkins is starting, grant this process full control
        ACL.impersonate(ACL.SYSTEM);
        try {
```

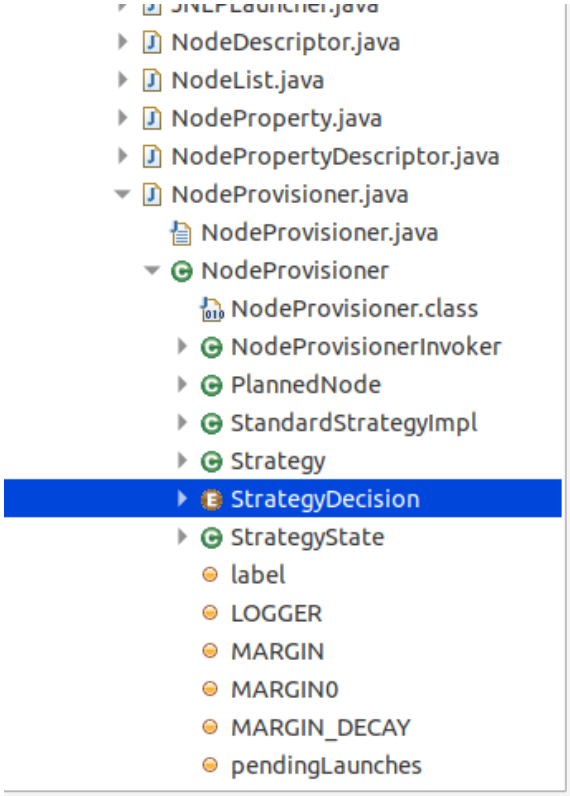
7. Design Patterns

- **Singleton**

```
    /**
     * Gets the {@link Jenkins} singleton.
     * @return {@link Jenkins} instance
     * @throws IllegalStateException for the reasons that {@link #getInstanceOrNull}
might return null
     * @since 2.98
     */
    @Nonnull
    public static Jenkins get() throws IllegalStateException {
        Jenkins instance = getInstanceOrNull();
        if (instance == null) {
            throw new IllegalStateException("Jenkins.instance is missing. Read the
documentation of Jenkins.getInstanceOrNull to see what you are doing wrong.");
        }
        return instance;
    }
```

7. Design Patterns

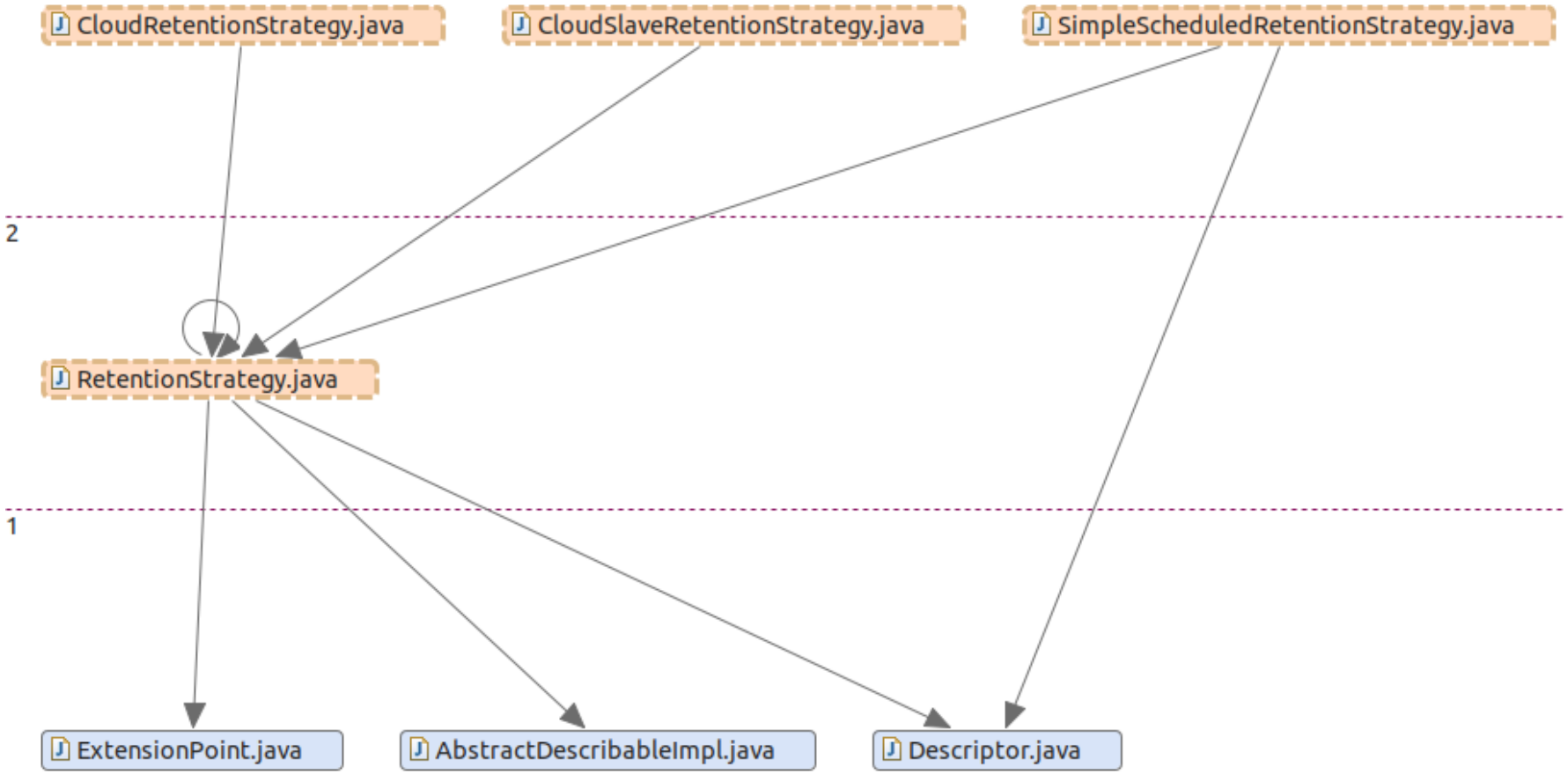
- **Strategy**




```
333  /**
334   * Represents the decision taken by an individual {@link hudson.slaves.NodeProvis
335   * @since 1.588
336   */
337  public enum StrategyDecision {
338      /**
339       * This decision is the default decision and indicates that the {@link hudson.
340       * either could not provision sufficient resources or did not take any action.
341       * will be able to contribute to the ultimate decision.
342       */
343      CONSULT_REMAINING_STRATEGIES,
344      /**
345       * This decision indicates that the {@link hudson.slaves.NodeProvisioner.Strat
346       * action so as to ensure that the required resources are available, and there
347       * to consult the remaining strategies. Only return this decision when you are
348       * need for additional provisioning actions (i.e. you detected an excess workl
349       * for that excess workload).
350       */
351      PROVISIONING_COMPLETED
352  }
353
354  /**
355   * Extension point for node provisioning strategies.
356   * @since 1.588
357   */
358  public static abstract class Strategy implements ExtensionPoint {
```

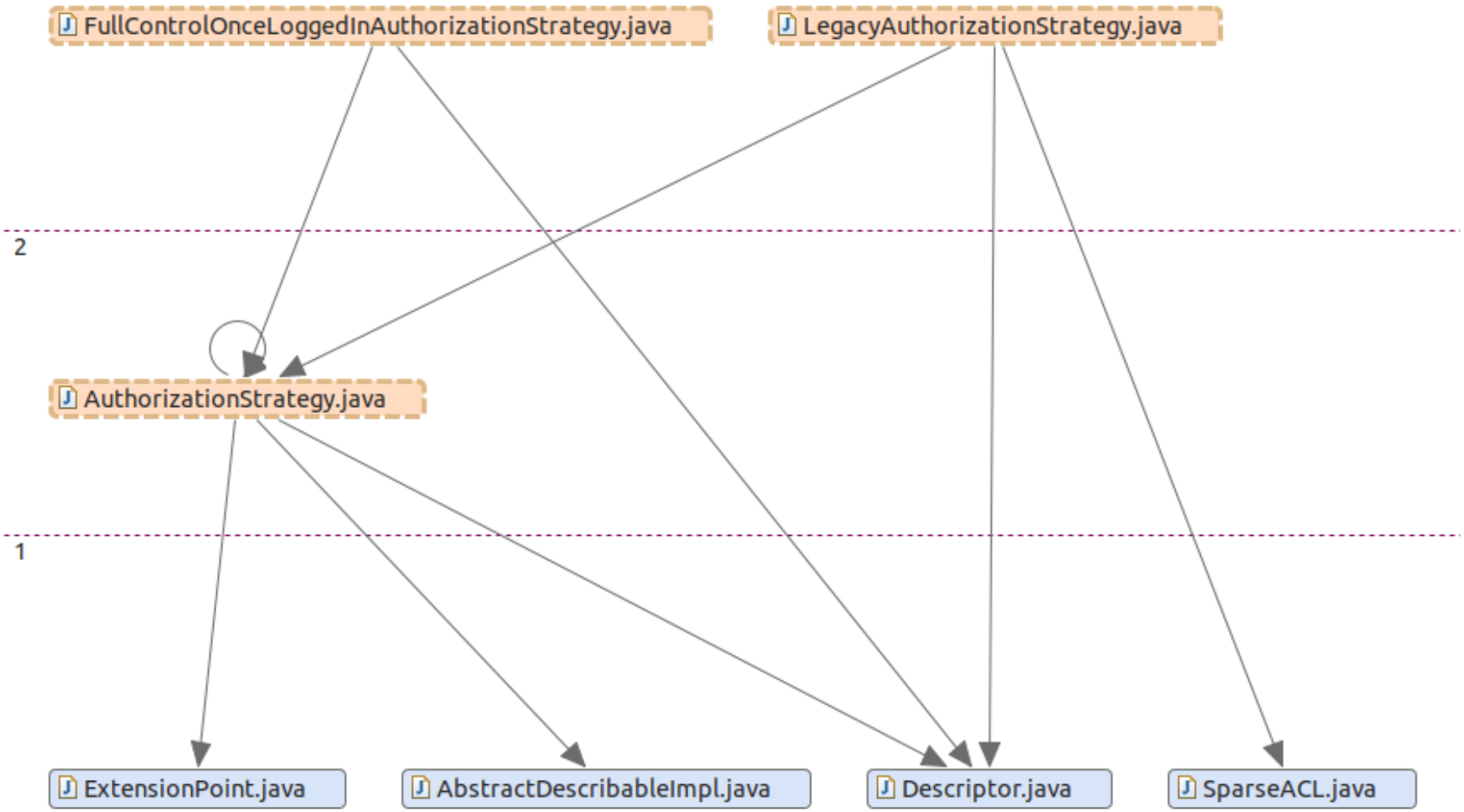
7. Design Patterns

- **Strategy**



7. Design Patterns

- **Strategy**

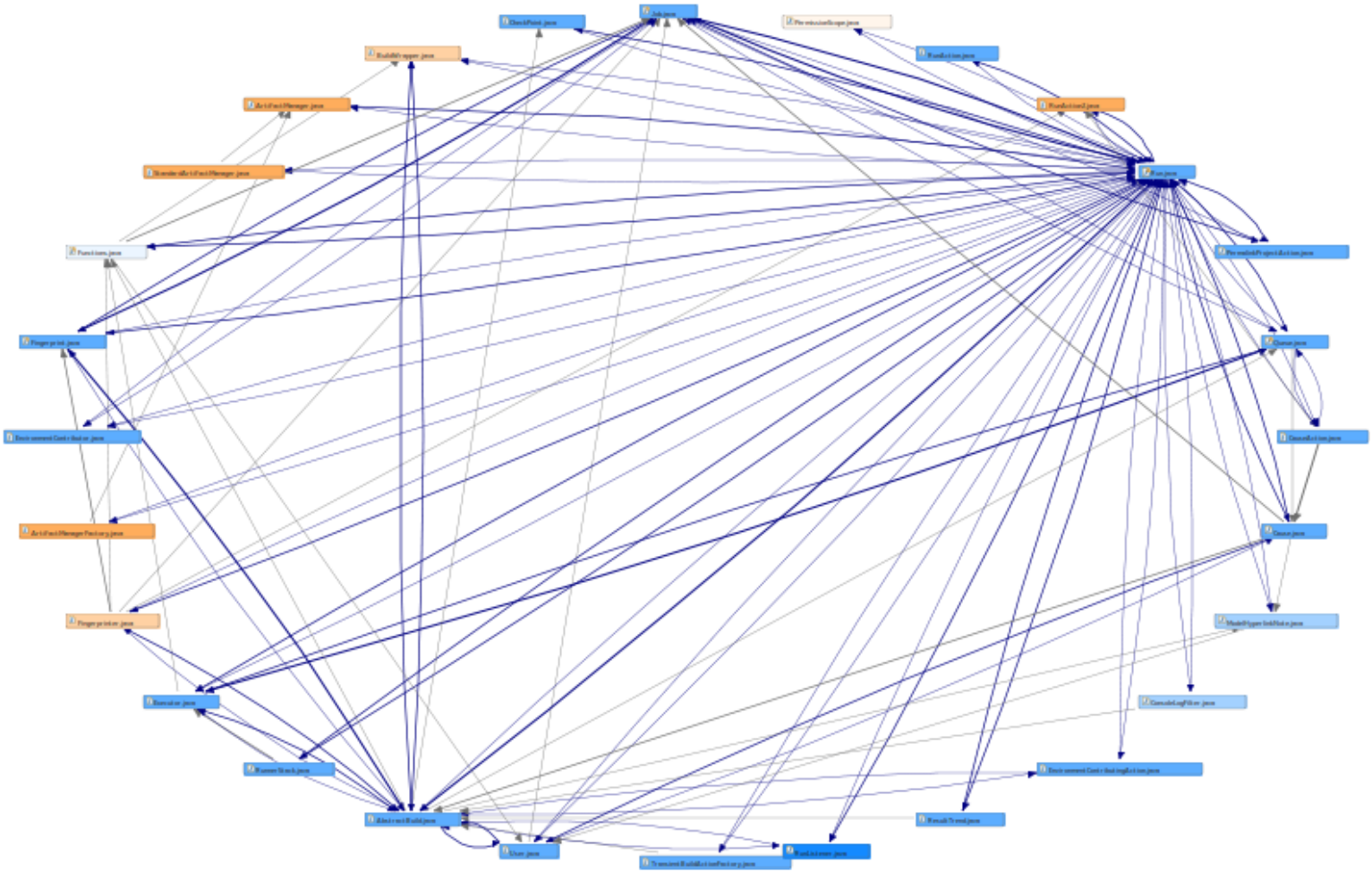


7. Design Patterns
   - **State**
   - Class Run...

```java
/**
 * The current build state.
 */
private volatile transient State state;

private enum State {
    /**
     * Build is created/queued but we haven't started building it.
     */
    NOT_STARTED,
    /**
     * Build is in progress.
     */
    BUILDING,
    /**
     * Build is completed now, and the status is determined,
     * but log files are still being updated.
     *
     * The significance of this state is that Jenkins
     * will now see this build as completed. Things like
     * "triggering other builds" requires this as pre-condition.
     * See JENKINS-980.
     */
    POST_PRODUCTION,
    /**
     * Build is completed now, and log file is closed.
     */
    COMPLETED
}
```

7. Design Patterns

- **State**



7. Design Patterns

- **Factory**

```java
package hudson.search;

import hudson.Extension;
import hudson.ExtensionList;
import hudson.ExtensionPoint;

/**
 * Creates a {@link Search} instance for a {@link SearchableModelObject}.
 *
 * <p>
 * This allows you to plug in different backends to the search, such as full-text search,
 * or more intelligent user-sensitive search, etc. Puts @{@link Extension} annotation
 * on your implementation to have it registered.
 *
 * <p>
 * Right now, there's no user control over which {@link SearchFactory} takes priority,
 * but we may do so later.
 *
 * @author Kohsuke Kawaguchi
 * @since 1.469
 */
public abstract class SearchFactory implements ExtensionPoint {
    /**
     * Creates a {@link Search} object.
     *
     * This method needs to execute quickly (without actually executing any search),
     * since it is created per incoming HTTP response.
     *
```

7. Design Patterns

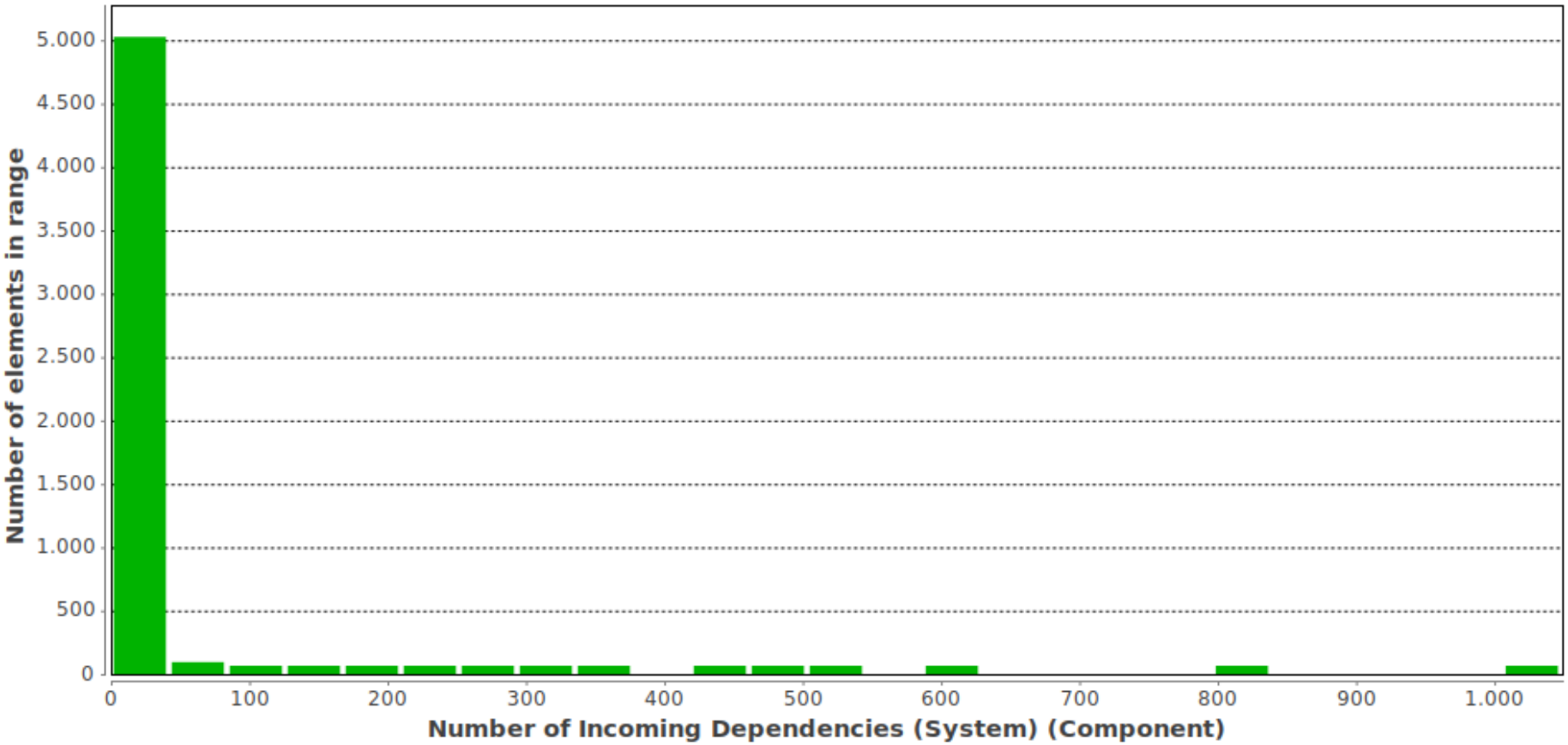
- **Composite**

```java
package jenkins.model.queue;

import hudson.model.TaskListener;
import hudson.model.queue.CauseOfBlockage;
import java.util.List;
import java.util.Map;
import java.util.TreeMap;
import org.apache.commons.lang.StringUtils;
import org.kohsuke.accmod.Restricted;
import org.kohsuke.accmod.restrictions.NoExternalUse;

/**
 * Represents the fact that there was at least one {@link hudson.model.Queue.JobOffer} which rejected a task.
 */
@Restricted(NoExternalUse.class)
public class CompositeCauseOfBlockage extends CauseOfBlockage {

    public final Map<String, CauseOfBlockage> uniqueReasons;

    public CompositeCauseOfBlockage(List<CauseOfBlockage> delegates) {
        uniqueReasons = new TreeMap<>();
        for (CauseOfBlockage delegate : delegates) {
            uniqueReasons.put(delegate.getShortDescription(), delegate);
        }
    }

    @Override
    public String getShortDescription() {
        return StringUtils.join(uniqueReasons.keySet(), "; ");
    }
```

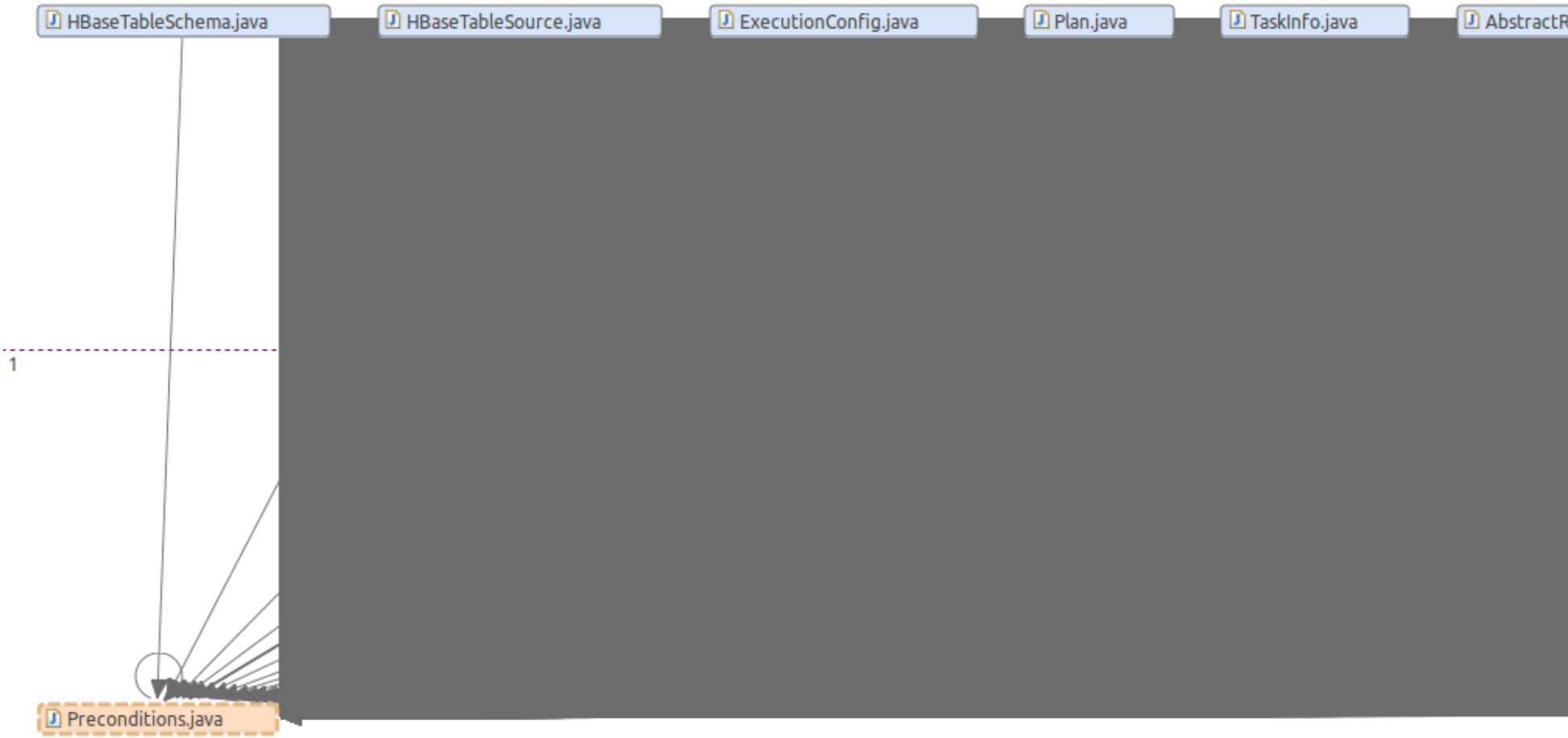
1. Quais as cinco classes em que a métrica de “Number of Incoming Dependencies” é maior?

Classe	**N.I.D**
/apache/flink/util/Preconditions.java	1048
/apache/flink/annotation/Internal.java	809
/apache/flink/api/common/typeinfo/TypeInformation.java	600
/apache/flink/annotation/PublicEvolving.java	542
/apache/flink/api/common/typeutils/TypeSerializer.java	490

Apache Flink



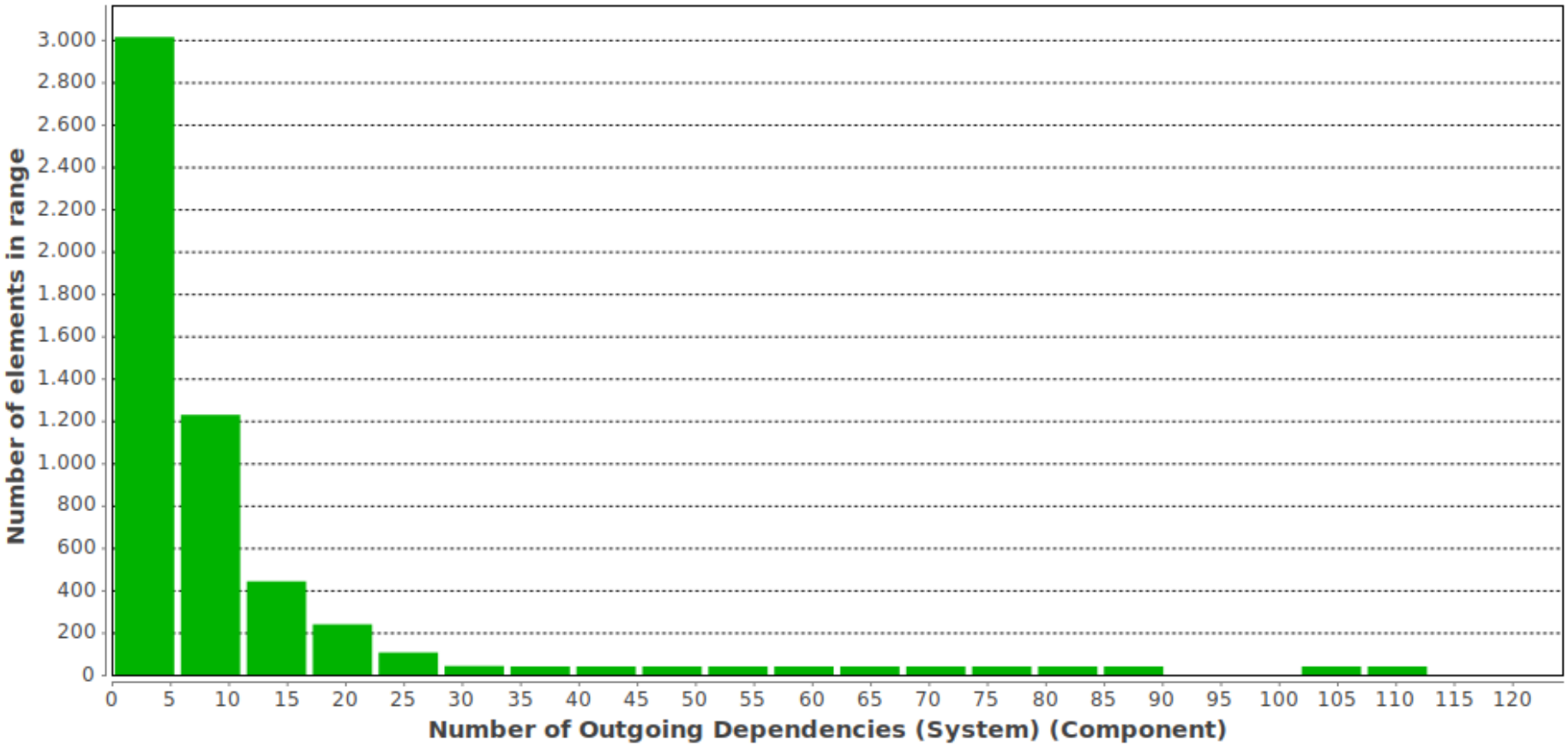
Apache Flink



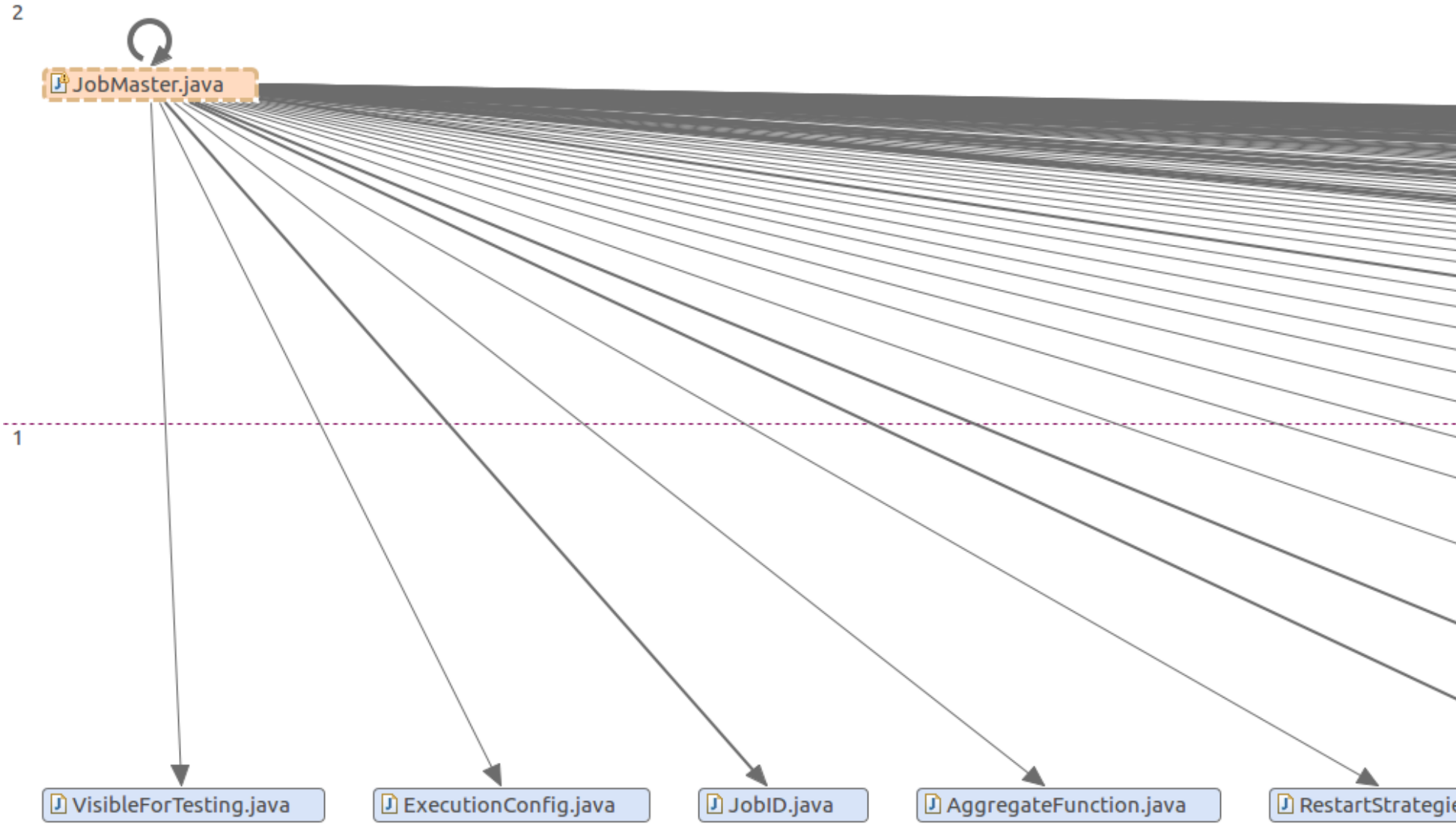
2. Quais as cinco classes em que a métrica de “Number of Outgoing Dependencies” é maior?

Classe	**N.O.D**
/apache/flink/runtime/jobmaster/JobMaster.java	113
/apache/flink/runtime/taskexecutor/TaskExecutor.java	111
/apache/flink/runtime/webmonitor/WebMonitorEndpoint.java	107
/apache/flink/runtime/taskmanager/TaskManager.scala	102
/apache/flink/client/program/rest/RestClusterClient.java	89

Apache Flink



Apache Flink



3. Mostrar quais são os tipos de dependências existentes as classes anteriores.

Incoming

	Tipos de Dependência
Preconditions	Aggregated, Static Method Call
Internal	Aggregated, Has Annotation
TypeInformation	Aggregated, Extends, Field, Local Variable, New Array, Parameter, Returns, Static Method Call, Virtual Method Call
PublicEvolving	Aggregated, Has Annotation
TypeSerializer	Aggregated, Extends, Field, Local Variable, New Array, Parameter, Returns, Type Argument, Virtual Method Call

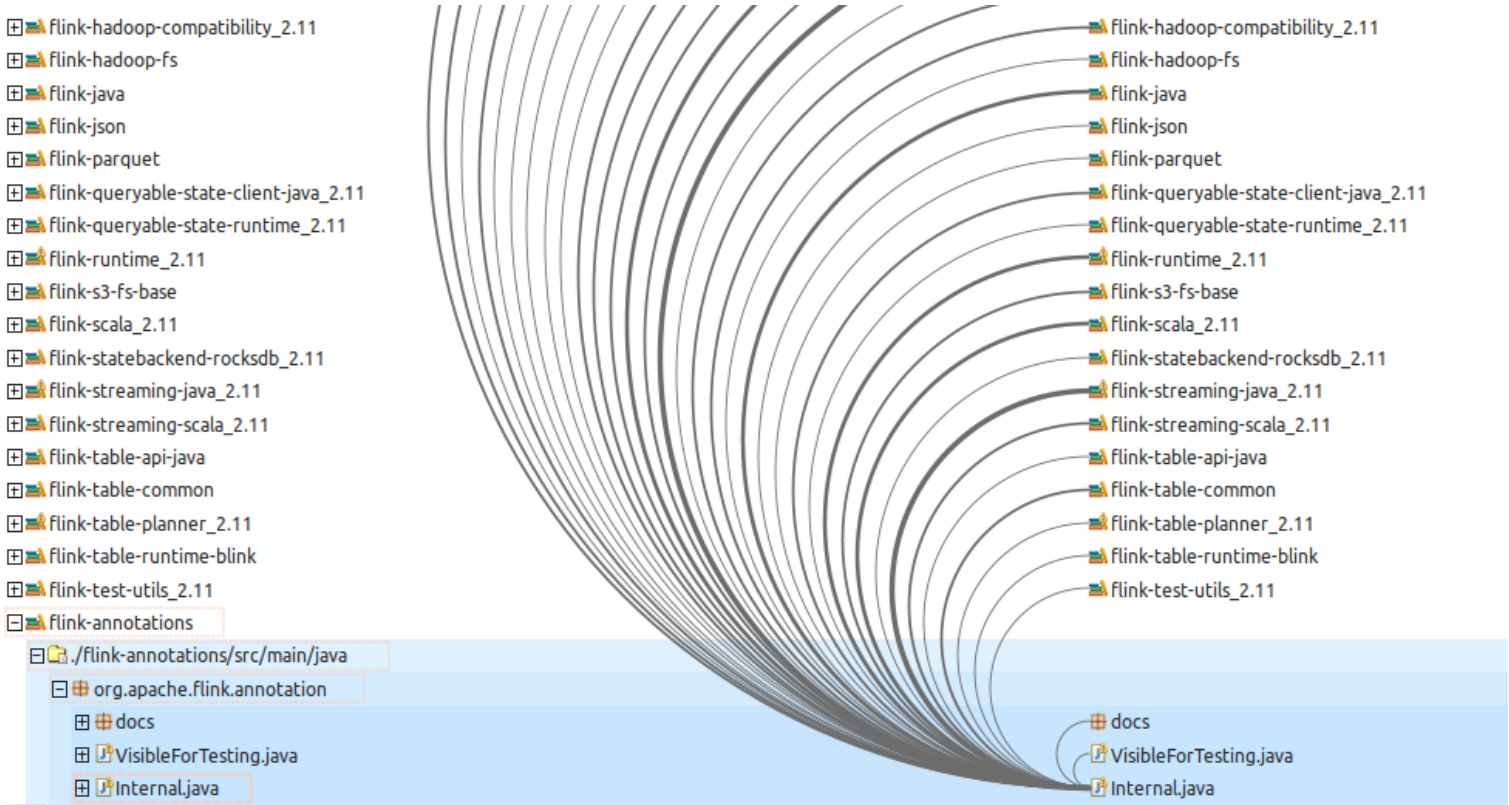
3. Mostrar quais são os tipos de dependências existentes as classes anteriores.

Outcoming

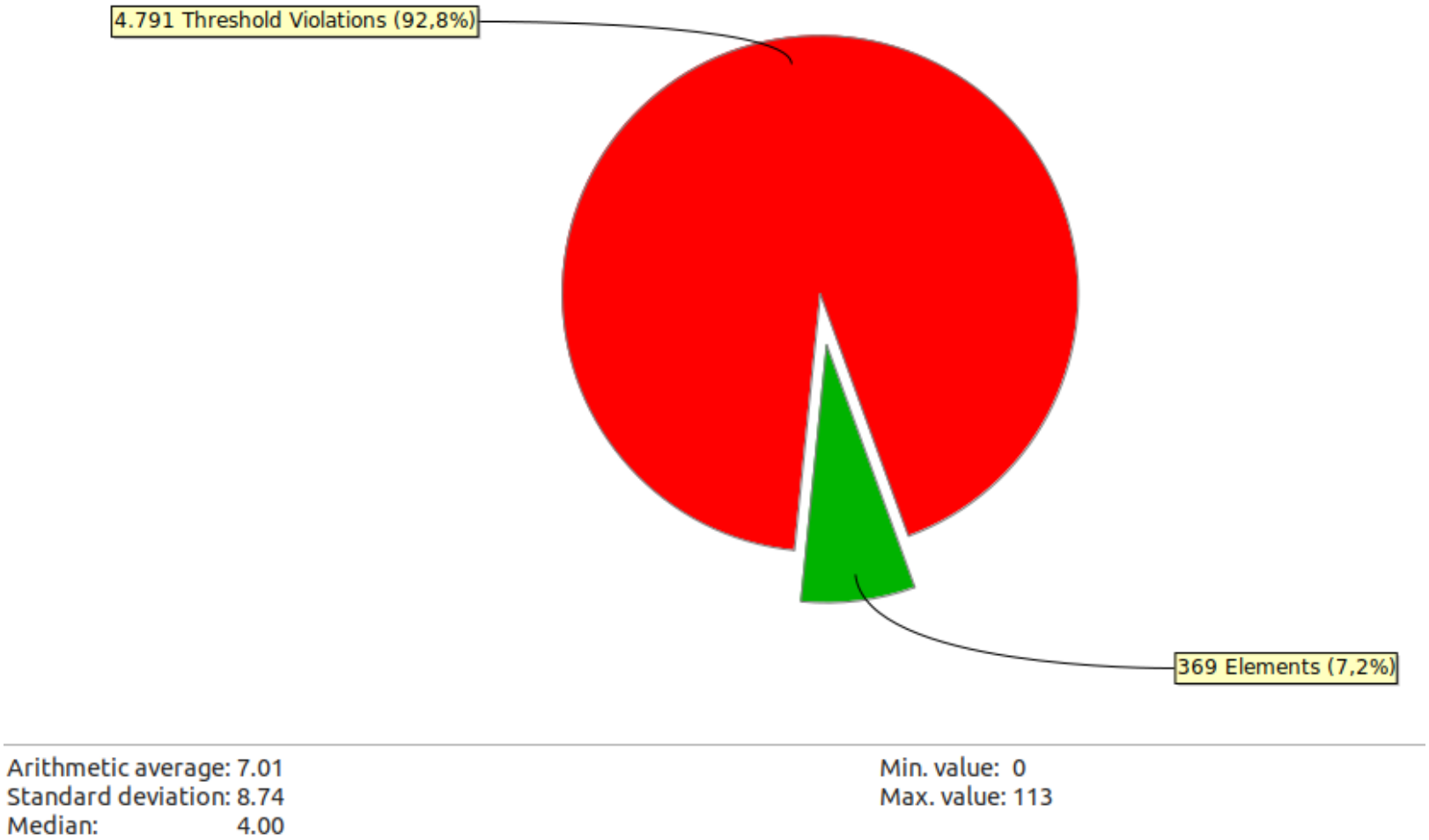
	Tipos de Dependência
JobMaster	Aggregated, Catch, Field, Implements, Interface Method Call, Local Variable, Parameter, Read Field, Static Method Call, Type Argument, Uses, Virtual Method Call
TaskExecutor	Aggregated, Catch, Has Annotation, Implements, Interface Method Call, Local Variable, Parameter, Static Method Call, Type Argument, Uses, Virtual Method Call
WebMonitorEndpoint	Aggregated, Has Annotation, Implements, Interface Bound, Parameter, Read Field, Returns, Static Method Call, Type Argument, Uses
TaskManager	Aggregated, Catch, Extends, Field, Has Annotation, Implements, Instance Of, Interface Method Call, New Array, Parameter, Read Field, Returns, Static Method Call, Uses, Virtual Method Call
RestClusterClient	Aggregated, Catch, Implements, Interface Method Call, New Array, Returns, Static Method Call, Throws, Type Argument, Uses, Virtual Method Call

4. Quais daquelas dependências são do tipo Associação?

- Quando uma classe possui um atributo do tipo da outra
  - Field ou New
  - Pode ser contido pelo Aggregated



5. Averiguar se há classes cujas “outgoing dependencies” são zero.



6. Utilize a métrica Abstractness para identificar padrões de projetos.

Element [640]	Abstractness (System)
org.apache.flink.api.common.io.statistics	1.00
org.apache.flink.streaming.api.functions.co	1.00
org.apache.flink.streaming.api.scala.function	1.00
org.apache.flink.api.java.hadoop.common	1.00
org.apache.flink.table.plan.nodes.exec	1.00
org.apache.flink.table.plan.nodes.physical	1.00
org.apache.flink.runtime.state.internal	1.00
org.apache.flink.runtime.highavailability.nonha	1.00
org.apache.flink.runtime.util.event	1.00
org.apache.flink.streaming.api.checkpoint	1.00
org.apache.flink.graph.library.clustering	1.00
org.apache.flink.cep.scala.conditions	1.00
org.apache.flink.util.concurrent	1.00
org.apache.flink.runtime.event	1.00
org.apache.flink.cep.functions	1.00
org.apache.flink.cep.time	1.00
org.apache.flink.metrics.reporter	1.00
org.apache.flink.annotation	1.00
org.apache.flink.hcatalog	1.00

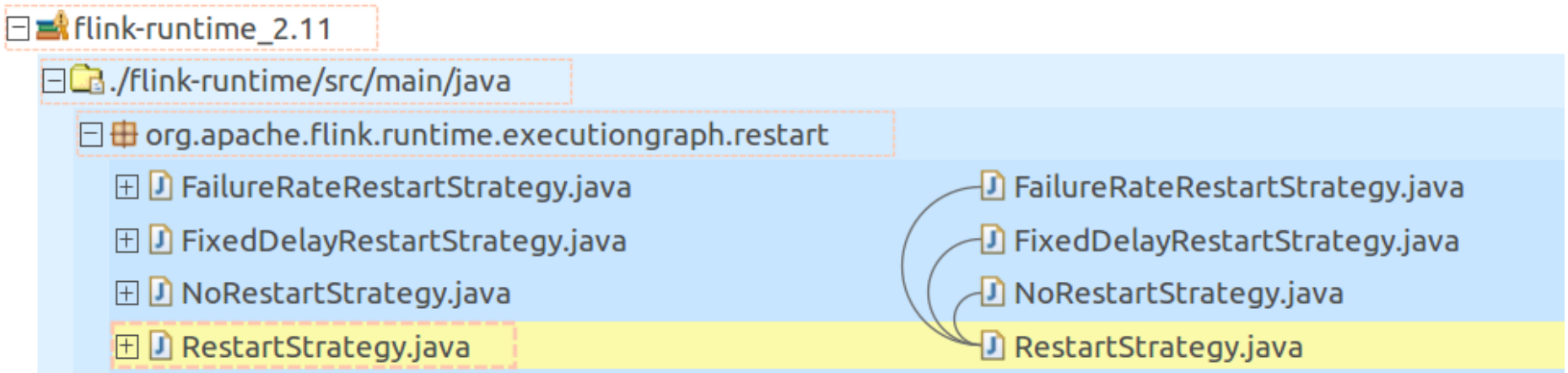
7. Design Patterns

- **Strategy**

```java
package org.apache.flink.runtime.executiongraph.restart;

import org.apache.flink.runtime.concurrent.ScheduledExecutor;
import org.apache.flink.runtime.executiongraph.ExecutionGraph;

/**
 * Strategy for {@link ExecutionGraph} restarts.
 */
public interface RestartStrategy {

    /**
     * True if the restart strategy can be applied to restart the {@link ExecutionGraph}.
     *
     * @return true if restart is possible, otherwise false
     */
    boolean canRestart();

    /**
     * Called by the ExecutionGraph to eventually trigger a full recovery.
     * The recovery must be triggered on the given callback object, and may be delayed
     * with the help of the given scheduled executor.
     *
     * <p>The thread that calls this method is not supposed to block/sleep.
     *
     * @param restarter The hook to restart the ExecutionGraph
     * @param executor An scheduled executor to delay the restart
     */
    void restart(RestartCallback restarter, ScheduledExecutor executor);
}
```

7. Design Patterns

- **Strategy**



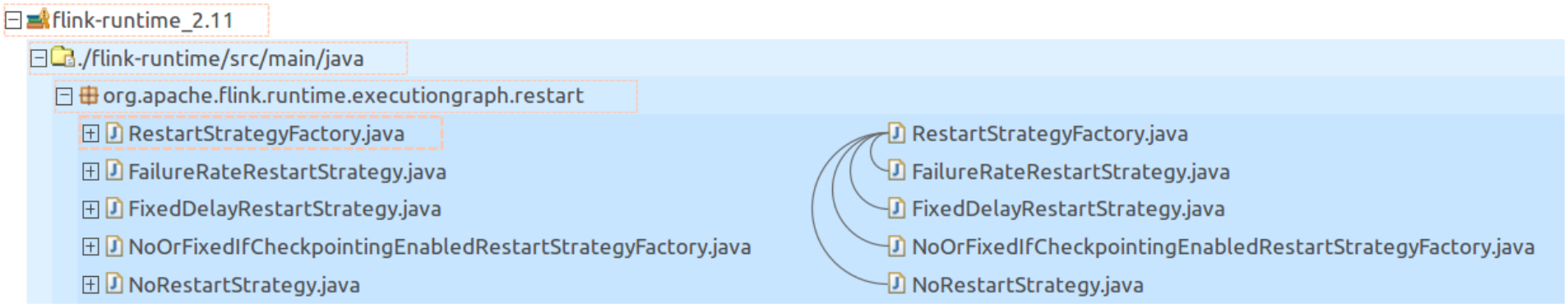
7. Design Patterns

- **Factory**
- public abstract class RestartStrategyFactory

```
public static RestartStrategy createRestartStrategy(RestartStrategies.RestartStrategyConfiguration restartStrategyConfiguration) {
    if (restartStrategyConfiguration instanceof RestartStrategies.NoRestartStrategyConfiguration) {
        return new NoRestartStrategy();
    } else if (restartStrategyConfiguration instanceof RestartStrategies.FixedDelayRestartStrategyConfiguration) {
        RestartStrategies.FixedDelayRestartStrategyConfiguration fixedDelayConfig =
            (RestartStrategies.FixedDelayRestartStrategyConfiguration) restartStrategyConfiguration;

        return new FixedDelayRestartStrategy(
            fixedDelayConfig.getRestartAttempts(),
            fixedDelayConfig.getDelayBetweenAttemptsInterval().toMilliseconds());
    } else if (restartStrategyConfiguration instanceof RestartStrategies.FailureRateRestartStrategyConfiguration) {
        RestartStrategies.FailureRateRestartStrategyConfiguration config =
                (RestartStrategies.FailureRateRestartStrategyConfiguration) restartStrategyConfiguration;
        return new FailureRateRestartStrategy(
                config.getMaxFailureRate(),
                config.getFailureInterval(),
                config.getDelayBetweenAttemptsInterval()
        );
    } else if (restartStrategyConfiguration instanceof RestartStrategies.FallbackRestartStrategyConfiguration) {
        return null;
    } else {
        throw new IllegalArgumentException("Unknown restart strategy configuration " +
            restartStrategyConfiguration + ".");
    }
}
```

7. Design Patterns

- **Factory**
- Relação NEW



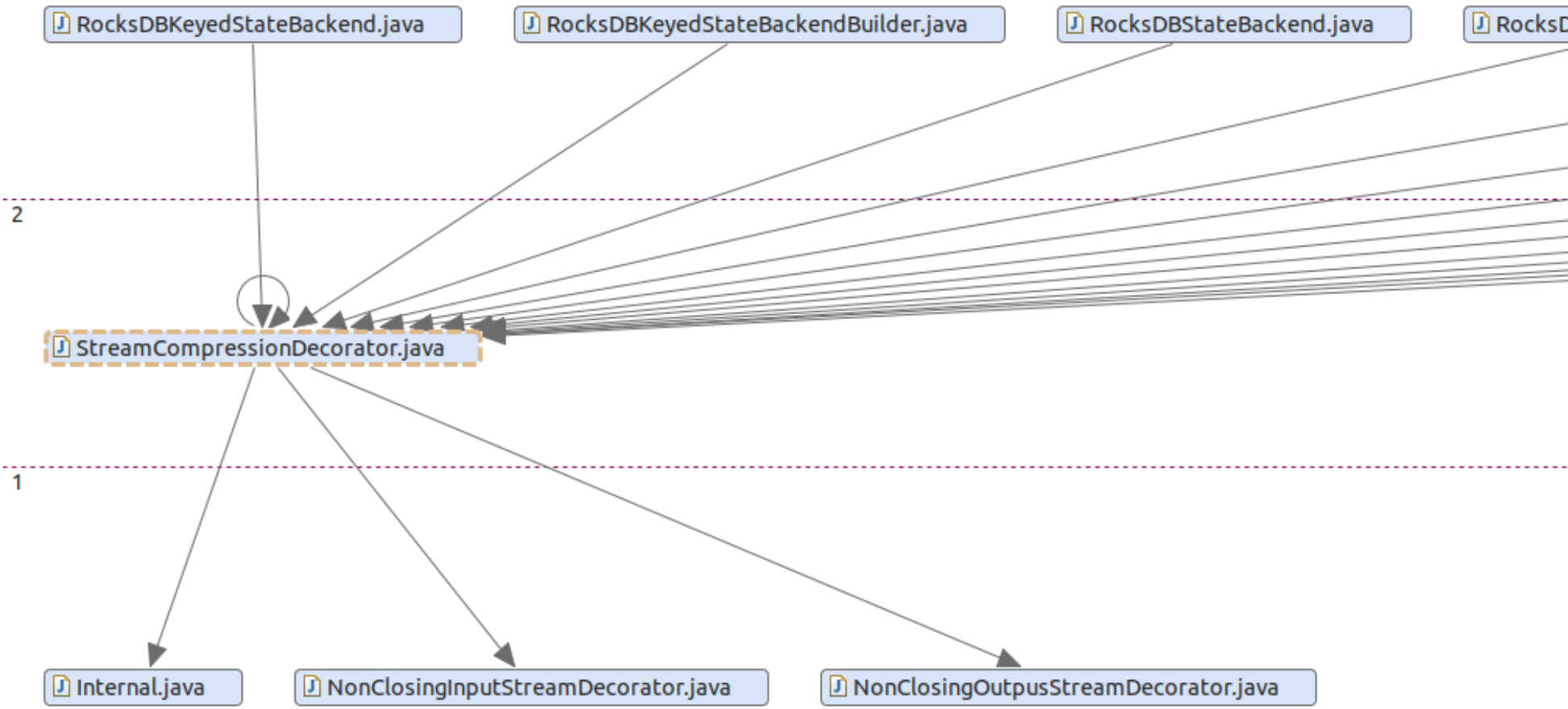
7. Design Patterns

- **Decorator**

```
/**
 * Implementations of this interface decorate streams with a compression scheme. Subclasses should be stateless
 */
@Internal
public abstract class StreamCompressionDecorator implements Serializable {
```

7. Design Patterns

- **Decorator**

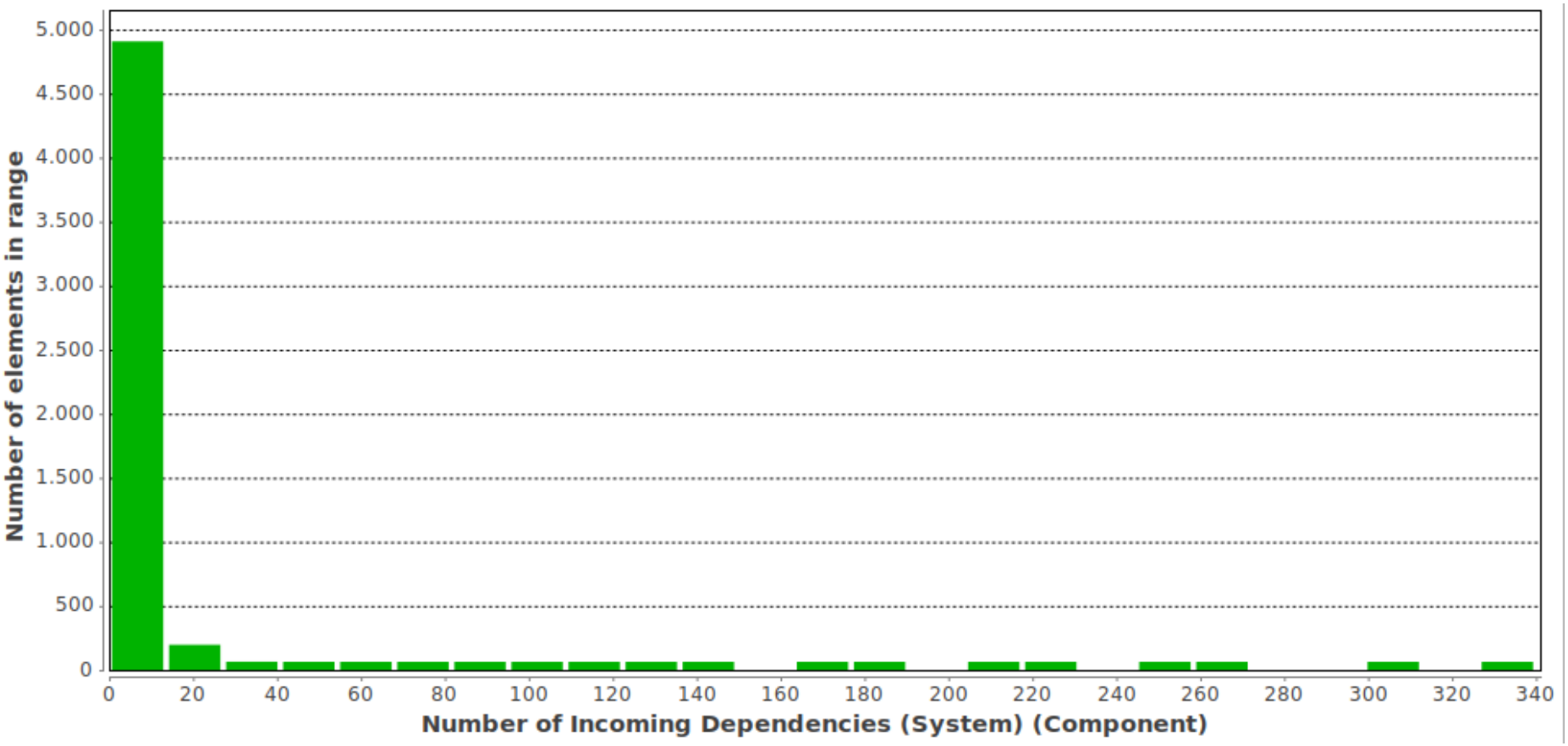


Eclipse Che

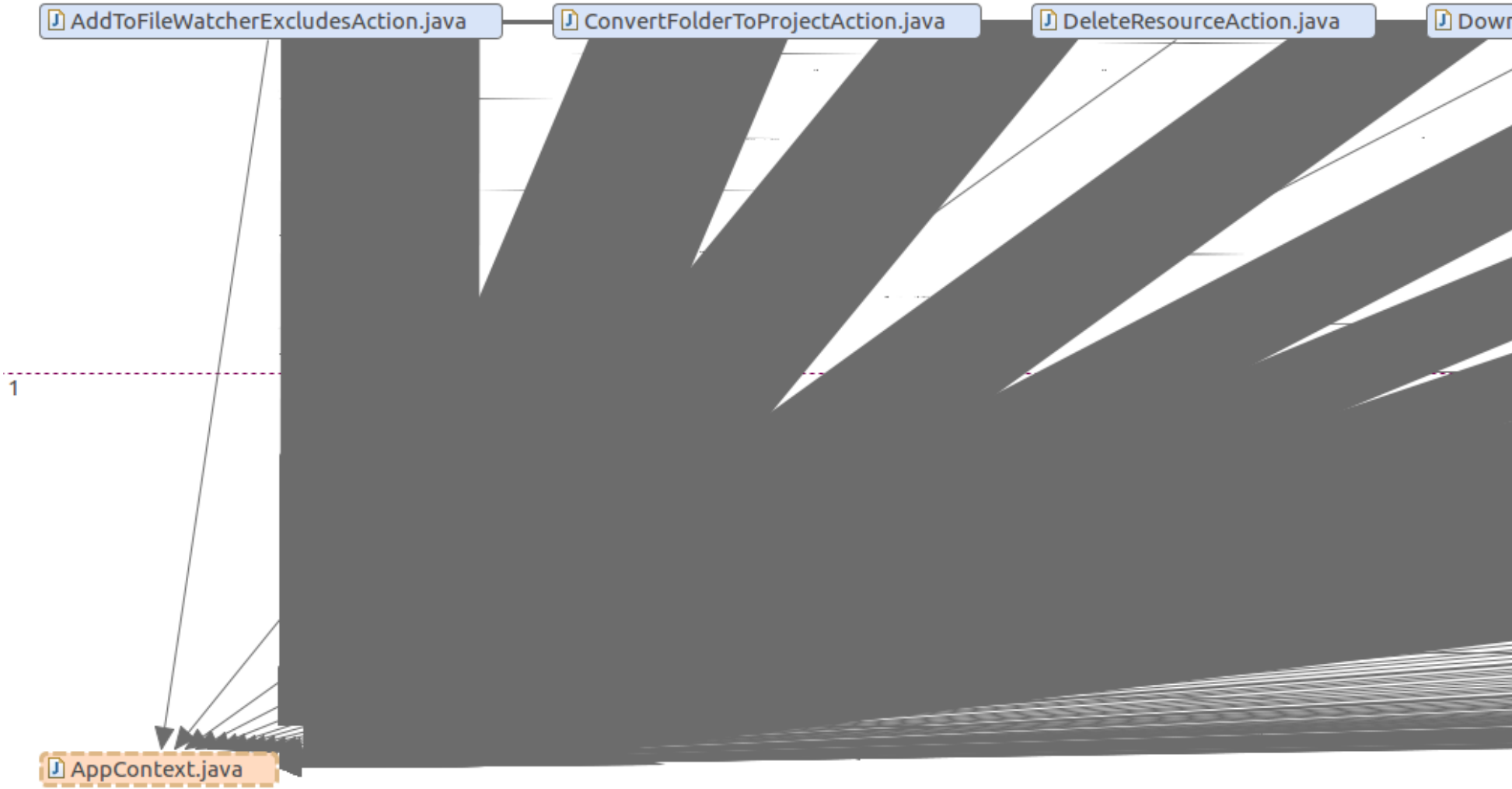
1. Quais as cinco classes em que a métrica de “Number of Incoming Dependencies” é maior?

Classe	**N.I.D**
/eclipse/che/api/promises/client/Promise.java	340
/eclipse/che/commons/annotation/Nullable.java	303
/eclipse/che/dto/shared/DTO.java	267
/eclipse/che/ide/api/app/AppContext.java	258
/eclipse/che/api/core/ServerException.java	255

Eclipse Che


Number of elements in range
5.000
4.500
4.000
3.500
3.000
2.500
2.000
1.500
1.000
500
0
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
Number of Incoming Dependencies (System) (Component)

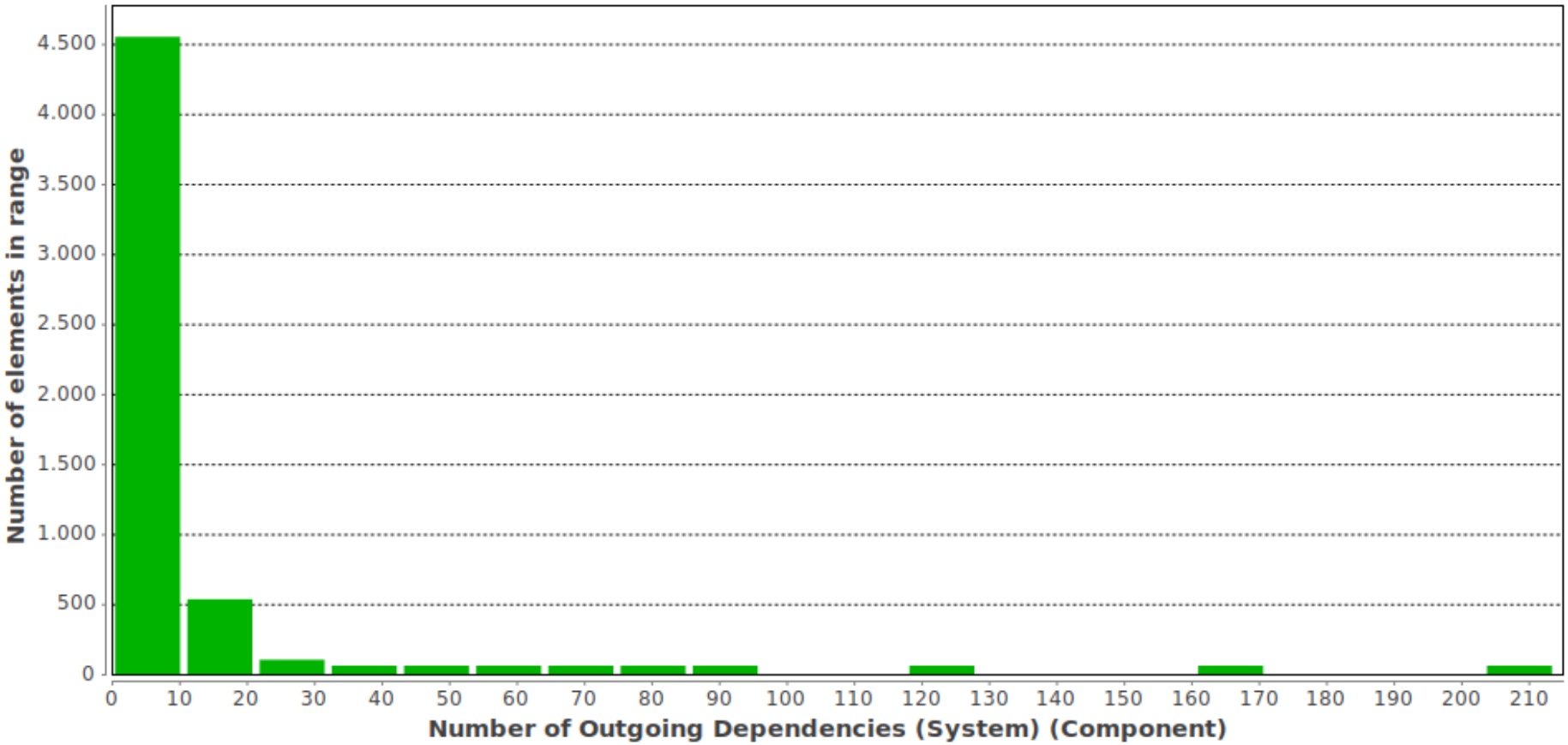
Eclipse Che


AddToFileWatcherExcludesAction.java
ConvertFolderToProjectAction.java
DeleteResourceAction.java
Down
1
AppContext.java

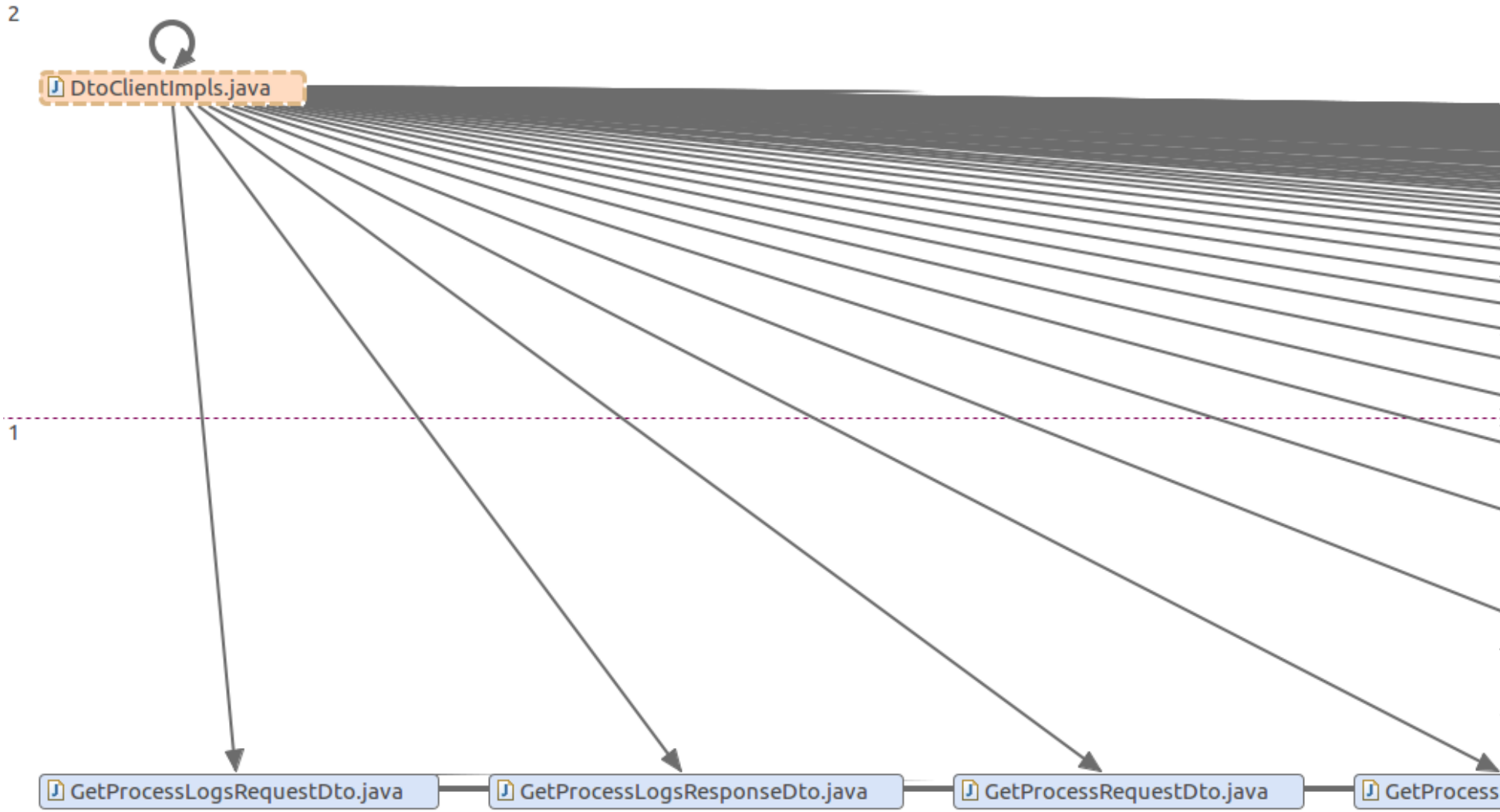
2. Quais as cinco classes em que a métrica de “Number of Outgoing Dependencies” é maior?

Classe	**N.O.D**
/eclipse/che/ide/api/dto/DtoClientImpls.java	214
/eclipse/che/api/deploy/WsMasterModule.java	164
/org/eclipse/che/ide/editor/orion/client/OrionEditorPresenter.java	127
/eclipse/che/ide/core/StandardComponentInitializer.java	120
/eclipse/che/ide/editor/orion/client/OrionEditorWidget.java	93

Eclipse Che


Number of elements in range
4.500
4.000
3.500
3.000
2.500
2.000
1.500
1.000
500
0
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
Number of Outgoing Dependencies (System) (Component)

Eclipse Che


2
1
DtoClientImpls.java
GetProcessLogsRequestDto.java
GetProcessLogsResponseDto.java
GetProcessRequestDto.java
GetProcess

3. Mostrar quais são os tipos de dependências existentes as classes anteriores.

Incoming

	Tipos de Dependência
Promise	Aggregated, Interface Method Call, Local Variable, Returns
Nullable	Aggregated, Has Annotation
DTO	Aggregated, Has Annotation
AppContext	Aggregated, Implements, Parameter
ServerException	Aggregated, Catch, Instance Of, Throws

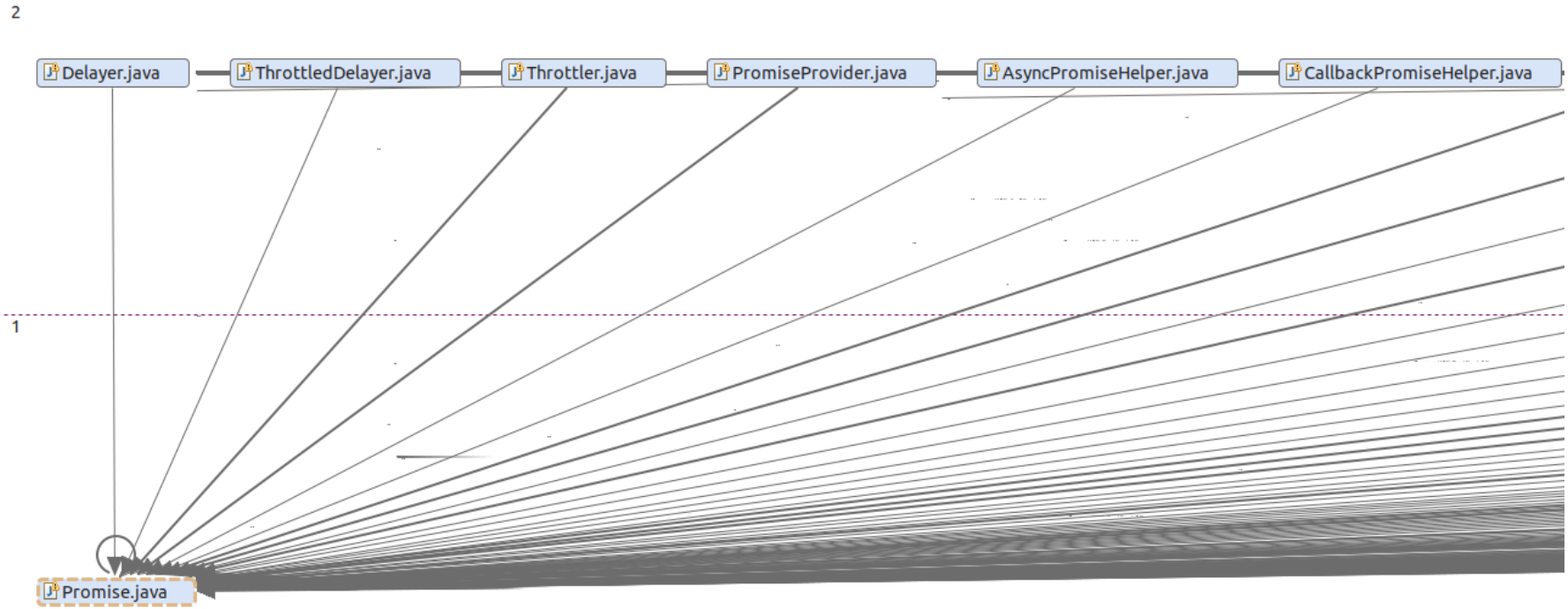
3. Mostrar quais são os tipos de dependências existentes as classes anteriores.

Outgoing

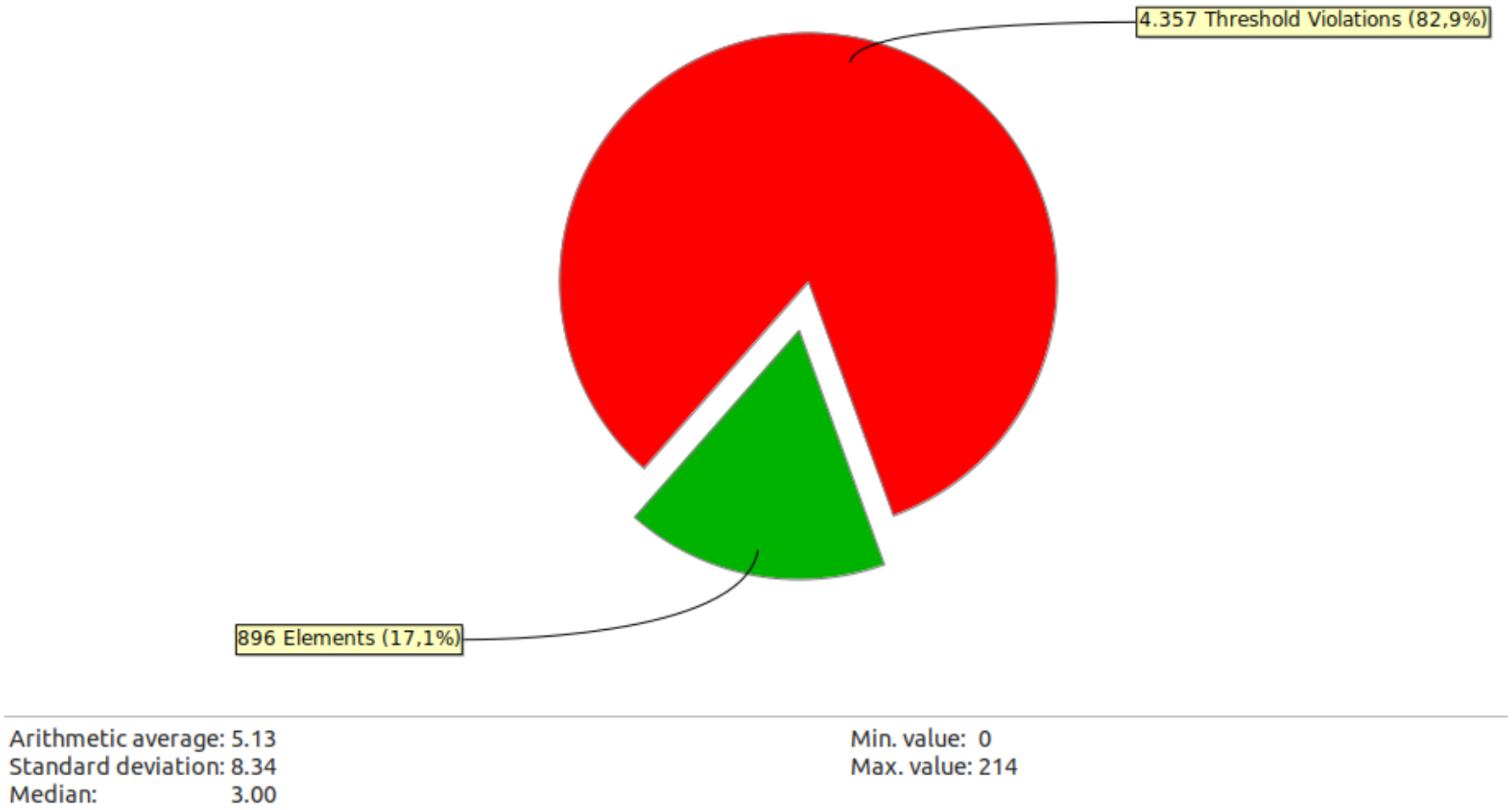
	Tipos de Dependência
DtoClientImpls	Aggregated, Has Annotation, Implements
WsMasterModule	Aggregated, Has Annotation, Interface Method Call, New Array, Read Field Inline, Static Method Call, Type Argument
OrionEditorPresenter	Aggregated, Has Annotation, Implements, Interface Method Call, Local Variable, Parameter, Read Field, Read Field Inline, Returns, Static Method Call, Throws, Type Argument, Uses, Virtual Method Call
StandardComponentInitializer	Aggregated, Extends, Field, Has Annotation, Interface Method Call, Parameter, Read Field Inline, Returns
OrionEditorWidget	Aggregated, Catch, Extends, Has Annotation, Implements, Interface Method Call, Parameter, Read Field, Read Field Inline, Returns, Static Method Call, Throws, Uses, Virtual Method Call

4. Quais daquelas dependências são do tipo Associação?

- Quando uma classe possui um atributo do tipo da outra
  - Field ou New
  - Pode ser contido pelo Aggregated



5. Averiguar se há classes cujas “outgoing dependencies” são zero.

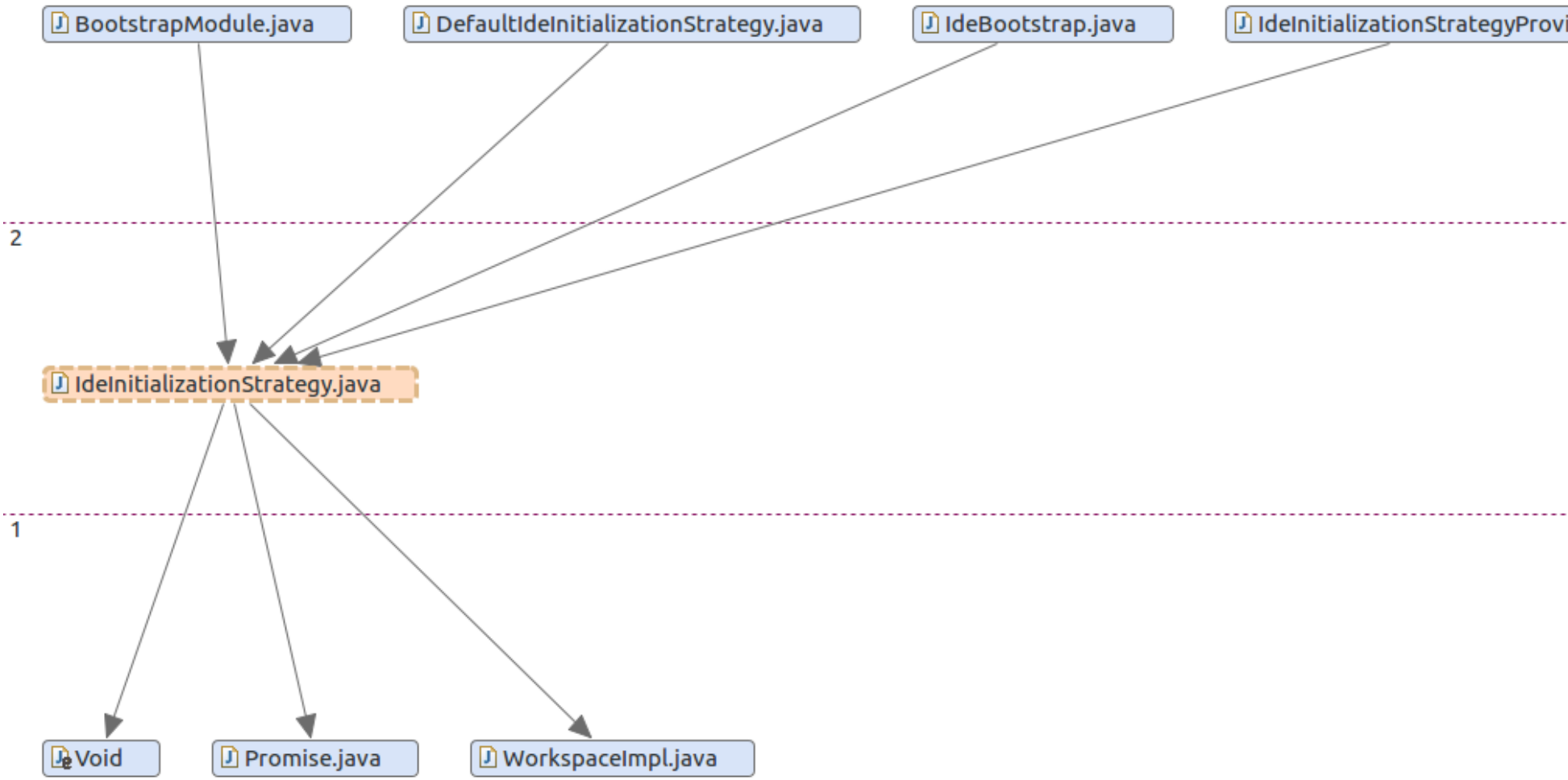


6. Utilize a métrica Abstractness para identificar padrões de projetos.

Element [1.135]	Abstractness (System)
org.eclipse.che.ide.ext.java.client.project.classpath.valueproviders.pages	1.00
org.eclipse.che.workspace.infrastructure.kubernetes.server.secure.jwtproxy.factory	1.00
org.eclipse.che.multiuser.api.permission.server.jpa.listener	1.00
org.eclipse.che.multiuser.machine.authentication.server.signature.spi	1.00
org.eclipse.che.ide.ext.java.shared.dto.refactoring	1.00
org.eclipse.che.ide.ext.java.client.inject.factories	1.00
org.eclipse.che.plugin.testing.ide.view.navigation.factory	1.00
org.eclipse.che.api.core.model.workspace.config	1.00
org.eclipse.che.api.core.model.project.type	1.00
org.eclipse.che.api.devfile.server.convert.tool	1.00
org.eclipse.che.api.project.templates.shared.dto	1.00
org.eclipse.che.api.workspace.shared.dto.stack	1.00
org.eclipse.che.api.project.shared.dto.service	1.00
org.eclipse.che.api.debug.shared.dto.event	1.00
org.eclipse.che.api.debug.shared.dto.action	1.00
org.eclipse.che.multiuser.api.permission.shared.model	1.00
org.eclipse.che.multiuser.api.permission.shared.dto	1.00
org.eclipse.che.multiuser.api.permission.server.spi	1.00
org.eclipse.che.multiuser.permission.workspace.server.model	1.00

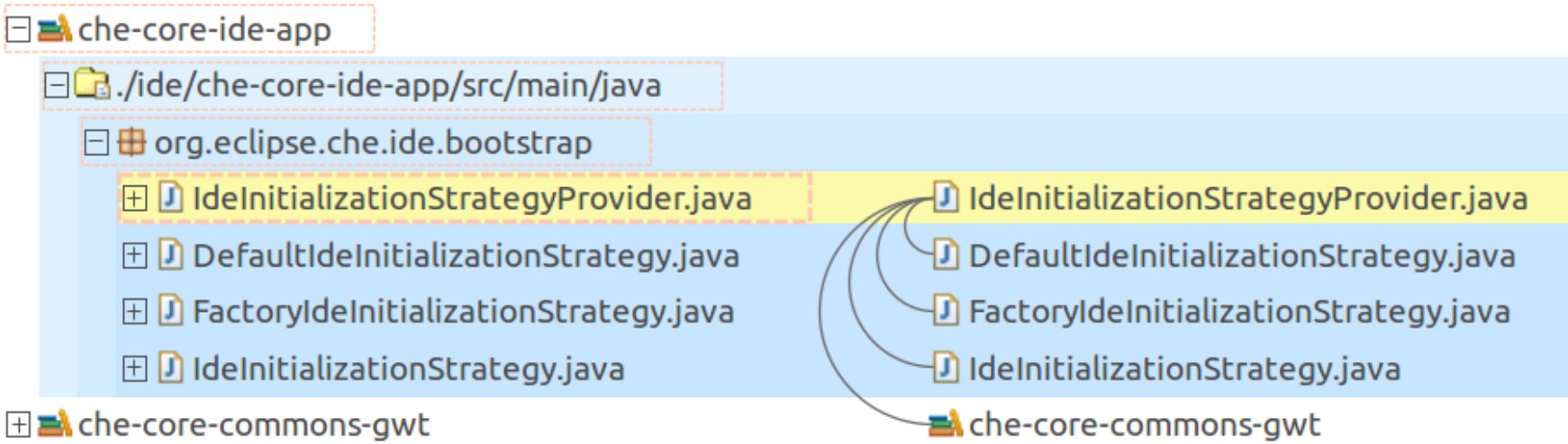
7. Design Patterns

- **Strategy'**



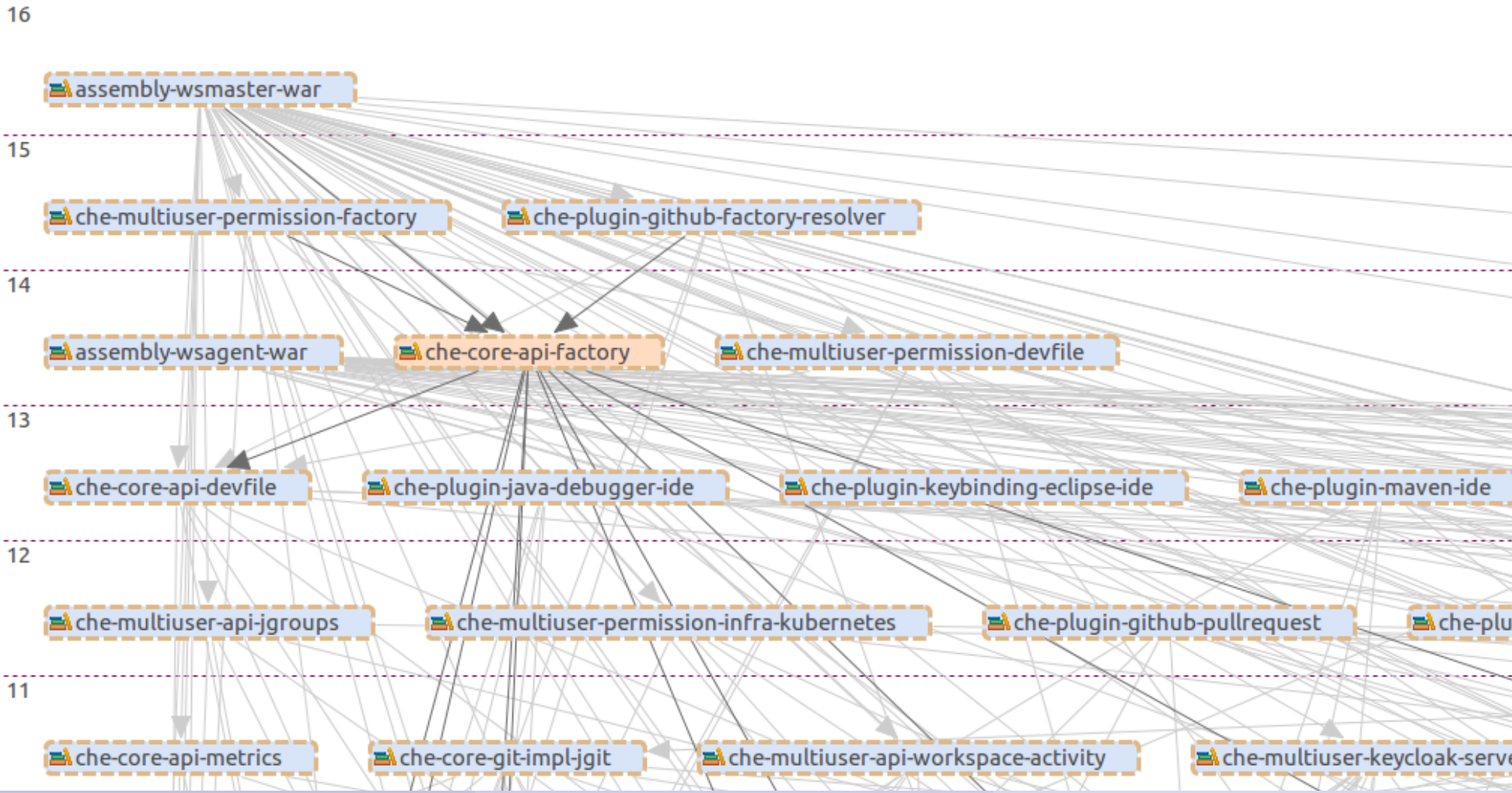
7. Design Patterns

- **Factory das** *Strategies*
- “Provides link IdeInitializationStrategy depending on the loading mode (default or factory).”



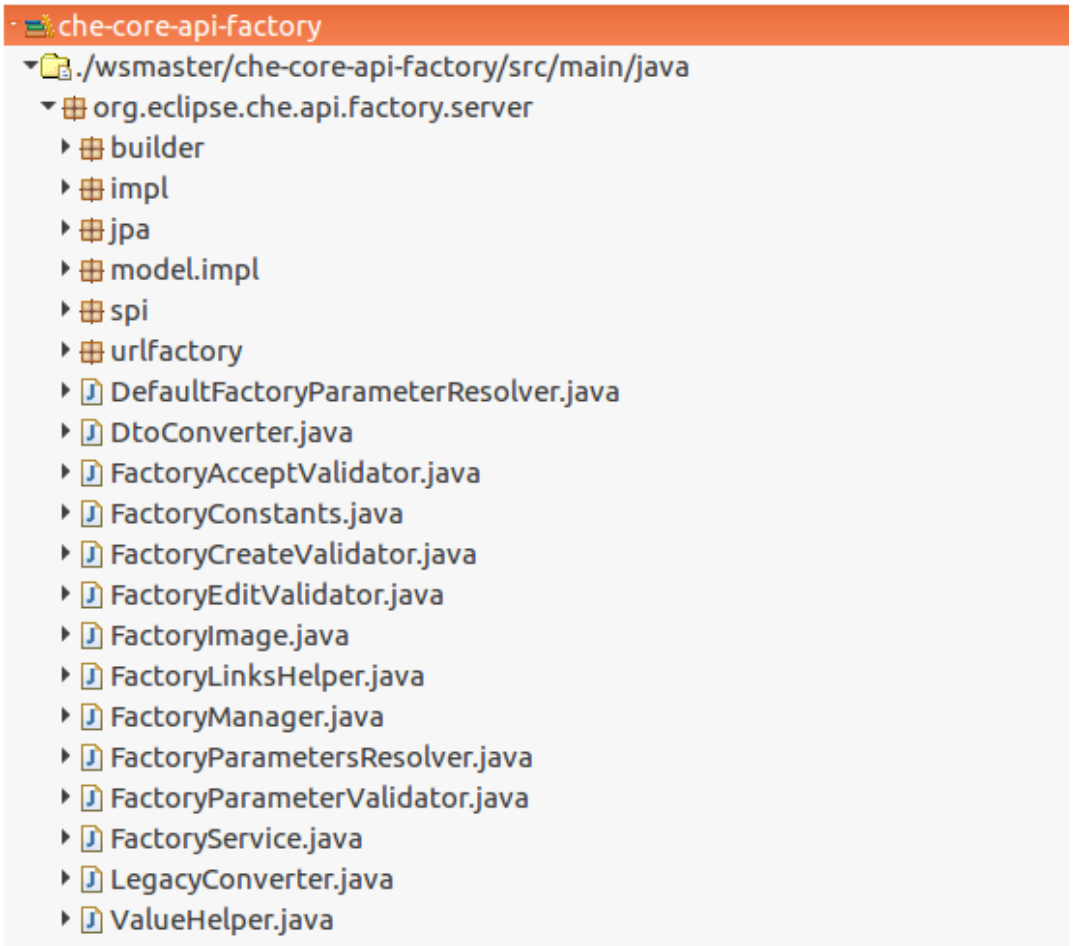
7. Design Patterns

- **Factory**



7. Design Patterns

- **Factory**
- “Provides version of third parties artifacts to use in the platform projects”



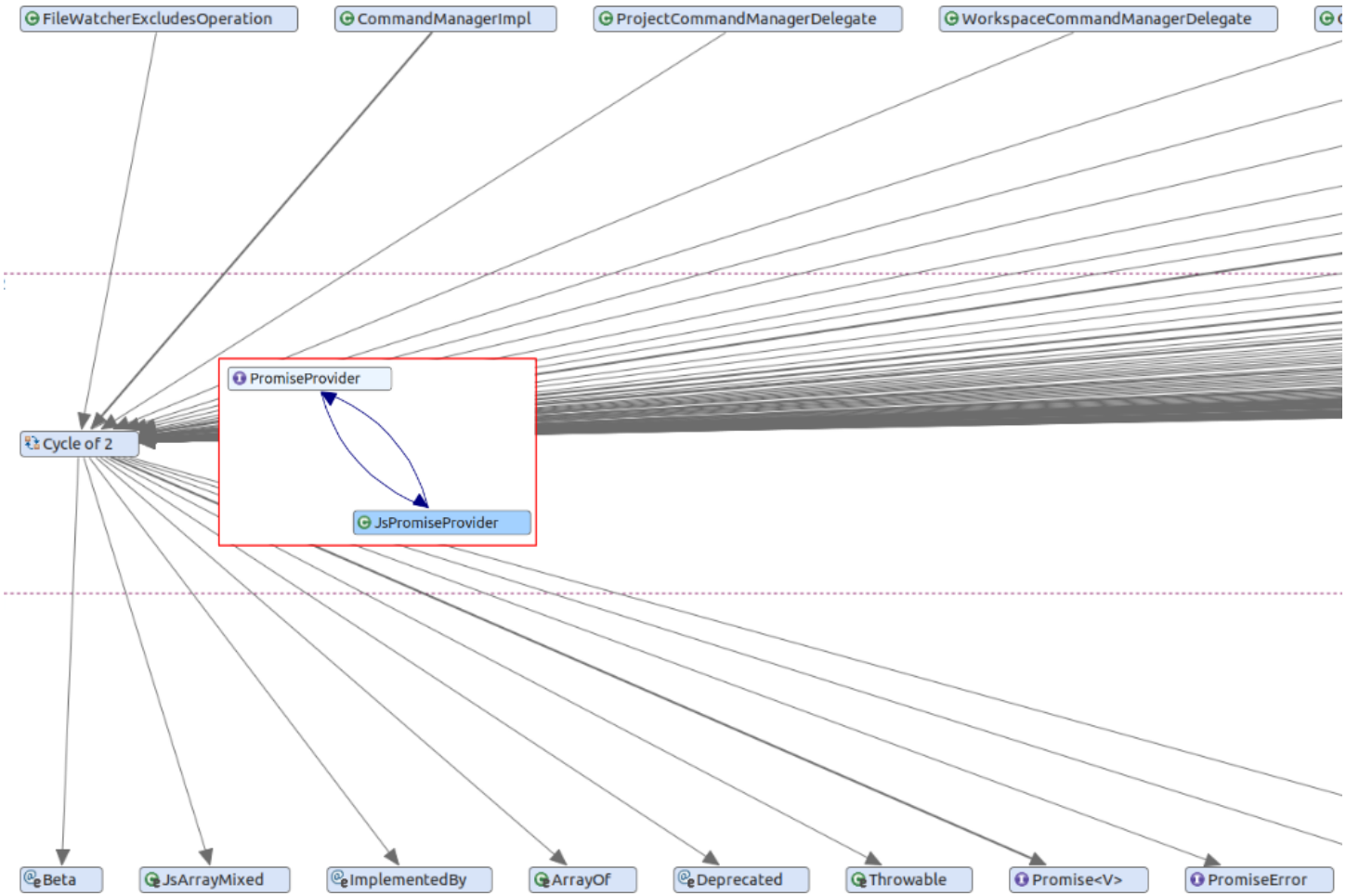
7. Design Patterns

- **Decorator**

```java
package org.eclipse.che.ide.ext.java.client.tree;

import static org.eclipse.che.ide.api.resources.Resource.FOLDER;

import com.google.common.annotations.Beta;
import com.google.common.base.Optional;
import com.google.inject.Inject;
import org.eclipse.che.api.promises.client.PromiseProvider;
import org.eclipse.che.ide.api.resources.Resource;
import org.eclipse.che.ide.api.resources.marker.Marker;
import org.eclipse.che.ide.ext.java.client.JavaResources;
import org.eclipse.che.ide.ext.java.client.resource.SourceFolderMarker;
import org.eclipse.che.ide.ext.java.shared.ContentRoot;
import org.eclipse.che.ide.project.node.icon.NodeIconProvider;
import org.eclipse.che.ide.ui.smartTree.data.settings.SettingsProvider;
import org.vectomatic.dom.svg.ui.SVGResource;

/** @author Vlad Zhukovskiy */
@Beta
public class SourceFolderDecorator implements NodeIconProvider {

  protected final PromiseProvider promises;
  protected final JavaNodeFactory nodeFactory;
  protected final SettingsProvider settingsProvider;
  private final JavaResources javaResources;

  @Inject
  public SourceFolderDecorator(
      PromiseProvider promises,
      JavaNodeFactory nodeFactory,
      SettingsProvider settingsProvider,
```

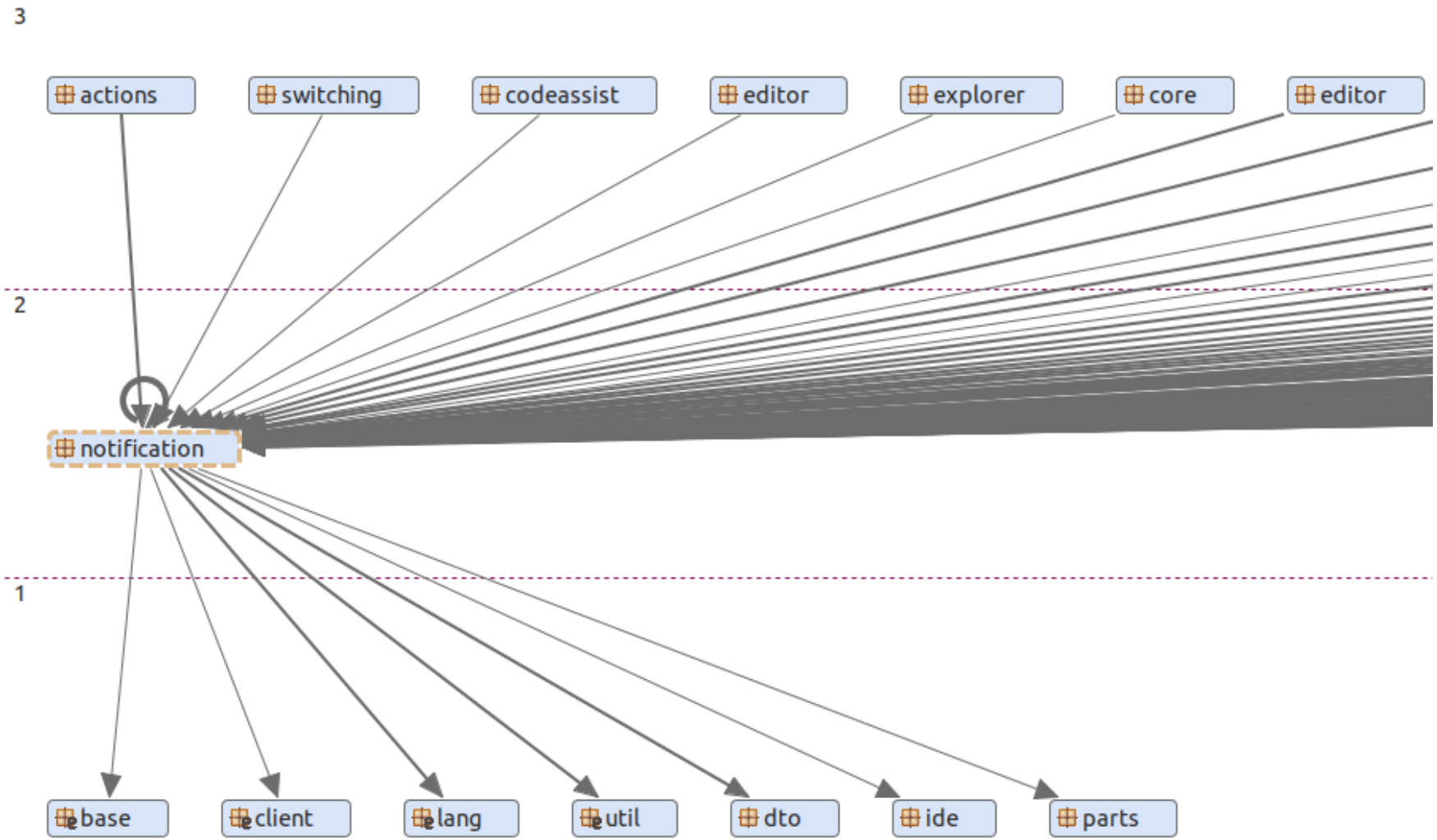
7. Design Patterns

- **Decorator**



7. Design Patterns

- **Observer**



Relembrando...Comparação

	Man. Level	Sys. ACD	Hig. Mod. ACD	Vul.
Jenkins	21.52	533.79	531.36	472
Apache Flink	27.27	623.18	219.23	486
Che	60.17	83.20	27.53	253

- Man.Level: System Maintainability Level
- Sys. ACD: System ACD
- Hig. Mod. ACD: Highest Module ACD
- Vul.: Vulnerabilities

Comparação Final

	Singleton	Strategy	State	Factory	Decorator	Composite
Jenkins	X	X	X	X		X
Apache Flink	X	X		X	X	
Che	X	X		X	X	X

- **DÚVIDAS?**