**Table 1: Mutants evaluation results with using EvoSuite as a baseline.**

| PID | # Mutated target stmts | # Mutants | # inline tests | | Mutation score | | | | | T(s) | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | | R0 | R1 | R0 | R1 | Unit | Randoop | Evosuite | R0 | R1 | Unit | Randoop |
| P1 | 2 | 10 | 23 | 2 | 80.0 | 80.0 | 80.0 | 0.0 | 0.0 | 30.3 | 29.8 | 54.7 | 316.8 |
| P2 | 1 | 1 | 16 | 2 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 3.6 | 3.5 | 4.6 | 9.0 |
| P3 | 12 | 52 | 254 | 46 | 96.2 | 96.2 | 96.2 | 69.2 | 21.2 | 190.2 | 185.0 | 393.4 | 1237.2 |
| P4 | 10 | 24 | 77 | 15 | 95.8 | 95.8 | 95.8 | 33.3 | 45.8 | 88.2 | 87.3 | 2109.8 | 548.4 |
| P5 | 2 | 2 | 48 | 4 | 100.0 | 100.0 | 50.0 | 100.0 | 100.0 | 11.9 | 6.5 | 16.9 | 32.3 |
| P6 | 3 | 32 | 84 | 11 | 68.8 | 62.5 | 53.1 | 0.0 | 6.2 | 117.8 | 102.0 | 140.8 | 367.5 |
| P7 | 4 | 9 | 22 | 7 | 100.0 | 77.8 | 11.1 | 0.0 | - | 70.9 | 53.4 | 4675.3 | 467.0 |
| P8 | 2 | 3 | 15 | 3 | 100.0 | 100.0 | 100.0 | 33.3 | 100.0 | 6.6 | 6.6 | 14.0 | 11.5 |
| P9 | 14 | 56 | 94 | 26 | 82.1 | 82.1 | 100.0 | 3.6 | 5.4 | 501.8 | 500.0 | 31957.5 | 3902.1 |
| P10 | 16 | 285 | 442 | 34 | 80.7 | 71.2 | 52.3 | 0.0 | 48.8 | 889.7 | 938.9 | 1431.0 | 3313.2 |
| P11 | 9 | 63 | 146 | 13 | 93.7 | 92.1 | 39.7 | 49.2 | 69.8 | 178.8 | 179.8 | 367.2 | 601.5 |
| P12 | 243 | 492 | 993 | 248 | 100.0 | 100.0 | 12.2 | 11.2 | 100.0 | 1712.2 | 1756.4 | 2129.6 | 4061.1 |
| P13 | 2 | 25 | 21 | 2 | 92.0 | 44.0 | 28.0 | 0.0 | 36.0 | 60.8 | 59.4 | 84.6 | 272.2 |
| P14 | 2 | 2 | 26 | 3 | 100.0 | 100.0 | 100.0 | 50.0 | 100.0 | 5.7 | 5.4 | 30.6 | 18.3 |
| P15 | 3 | 5 | 214 | 6 | 100.0 | 100.0 | 60.0 | 80.0 | 100.0 | 30.0 | 17.2 | 156.5 | 37.0 |
| P17 | 20 | 125 | 938 | 41 | 84.0 | 70.4 | 51.2 | 4.8 | 4.0 | 493.6 | 518.5 | 6345.8 | 3664.2 |
| P18 | 1 | 5 | 1 | 1 | 40.0 | 40.0 | 40.0 | 0.0 | 0.0 | 13.7 | 13.7 | 21.5 | 51.1 |
| P19 | 11 | 49 | 224 | 28 | 85.7 | 77.6 | 83.7 | 10.2 | 6.1 | 218.3 | 214.6 | 440.3 | 496.8 |
| P20 | 30 | 474 | 1725 | 44 | 77.4 | 50.6 | 89.2 | 39.7 | 89.9 | 1207.3 | 1123.9 | 1753.3 | 4692.6 |
| P21 | 38 | 132 | 915 | 64 | 100.0 | 92.4 | 90.2 | 72.7 | 89.4 | 857.2 | 619.4 | 1047.8 | 1971.6 |
| P22 | 128 | 1420 | 2272 | 294 | 69.3 | 67.3 | 64.4 | 13.7 | 31.1 | 3751.2 | 3809.2 | 20329.8 | 6262.6 |
| P23 | 5 | 25 | 36 | 14 | 92.0 | 92.0 | 92.0 | 8.0 | 0.0 | 69.5 | 69.0 | 271.8 | 222.7 |
| P24 | 2 | 24 | 133 | 2 | 100.0 | 66.7 | 4.2 | 66.7 | 0.0 | 93.1 | 89.4 | 182.4 | 184.1 |
| P25 | 4 | 12 | 54 | 4 | 33.3 | 0.0 | 25.0 | 0.0 | 33.3 | 68.8 | 67.8 | 92.3 | 157.8 |
| P26 | 2 | 5 | 20 | 2 | 20.0 | 20.0 | 40.0 | 20.0 | 20.0 | 13.3 | 13.3 | 21.9 | 29.9 |
| P27 | 16 | 70 | 707 | 29 | 58.6 | 58.6 | 97.1 | 100.0 | 77.1 | 621.5 | 609.2 | 1213.7 | 1444.5 |
| P28 | 2 | 5 | 274 | 3 | 100.0 | 100.0 | 60.0 | 20.0 | 100.0 | 66.2 | 29.6 | 87.9 | 931.3 |
| P29 | 4 | 10 | 25 | 4 | 90.0 | 60.0 | 100.0 | 30.0 | 30.0 | 23.5 | 23.5 | 48.9 | 70.3 |
| P30 | 1 | 19 | 6 | 1 | 57.9 | 36.8 | 57.9 | 5.3 | 0.0 | 87.5 | 87.5 | 119.9 | 416.9 |
| P31 | 4 | 12 | 4 | 4 | 100.0 | 100.0 | 100.0 | 33.3 | 100.0 | 28.0 | 27.7 | 56.1 | 1618.3 |
| Total | 593 | 3448 | 9809 | 957 | 2497.5 | 2234.0 | 1906.6 | 1020.9 | 1414.1 | 11511.1 | 11247.6 | 75600.1 | 37409.7 |
| Avg | 19 | 111 | 316 | 31 | 80.6 | 72.1 | 61.5 | 32.9 | 47.1 | 371.3 | 362.8 | 2438.7 | 1206.8 |

## A   EXPERIMENT RESULTS WITH EVOSUITE

Table 1 shows the evaluation results after adding EvoSuite as a baseline. We use EvoSuite to generate tests for each class that has target statements, with a time limit of 120s per class, and with the same configuration as described in Sebastian et al. [1]. The results show that inline tests generated by ExLɪ achieve higher mutation score than unit tests from all three sources: developer-written, Randoop-generated, and EvoSuite-generated. Among the three sources of unit tests, EvoSuite-generated unit tests do achieve better mutation score than Randoop-generated ones, though still lower than developer-written ones. ExLɪ is agnostic to the source of unit tests, and the generated inline tests achieve better mutation scores regardless of the source.

Figure 1 shows the Venn diagram of the mutants killed by ExLɪ-generated inline tests, developer-written unit tests, Randoop-generated unit tests, and EvoSuite-generated unit tests. Inline tests can kill 279 mutants that are not killed by unit tests. There are 181 mutants killed by Randoop but not by EvoSuite and 1240 mutants killed by EvoSuite but not by Randoop, showing that the two unit test generation techniques are complementary. Figure 2 lists the Venn diagram per project.
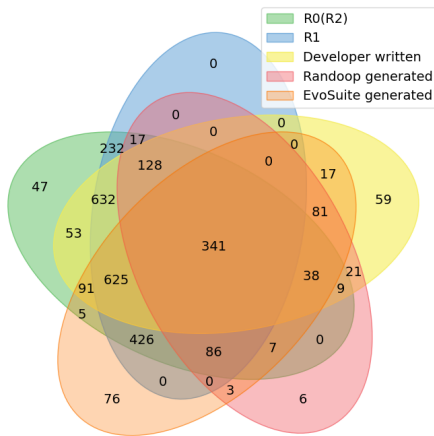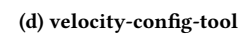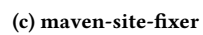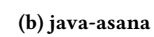
**Figure 1: Venn diagram of killed mutants.**

(a) aquarius-sdk-java



(b) java-asana



(c) maven-site-fixer
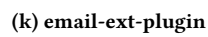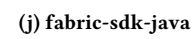


(d) velocity-config-tool
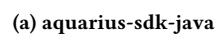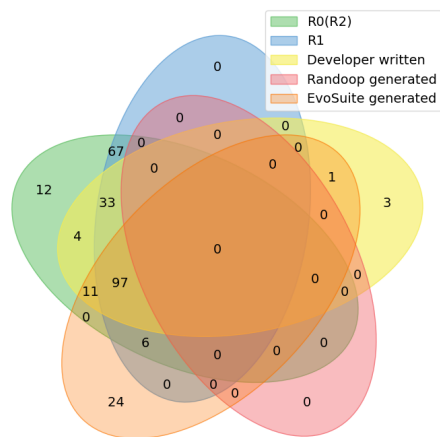
(e) cyclonedx-core-java
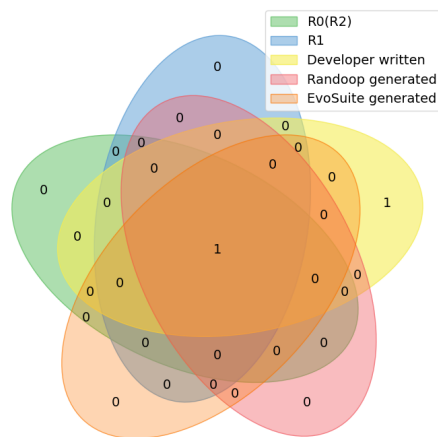


(f) property-loader



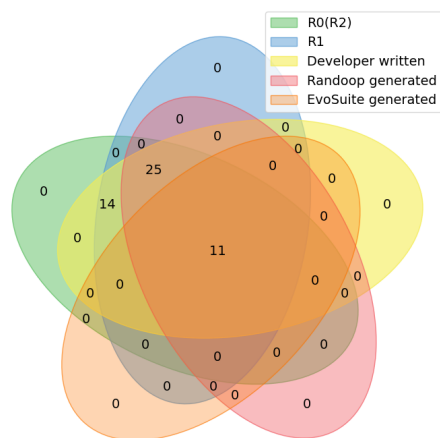(g) amazon-sqs-java-extended-client-lib



(h) core
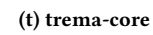
**(i) jproc**



**(j) fabric-sdk-java**



**(k) email-ext-plugin**



**(l) jscep**

**(m) sherdog-parser**



**(n) geoip-api-java**



**(o) elasticsearch-analysis-pinyin**



**(p) build-helper-maven-plugin**

**(q) properties-maven-plugin**

**(r) logstash-gelf**

**(s) mp3agic**

**(t) trema-core**
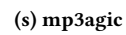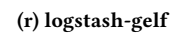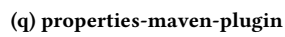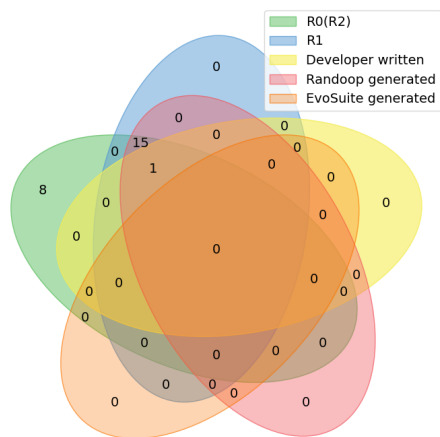
(u) ph-pdf-layout
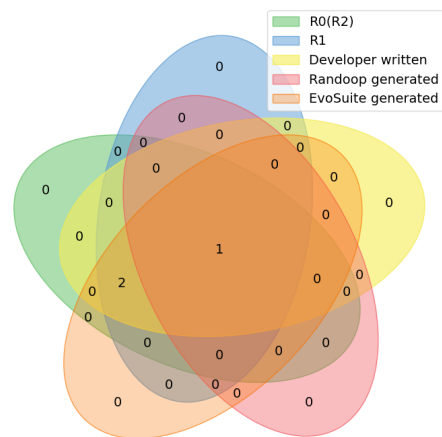


(v) extclassgenerator



(w) pdfcompare



(x) restfb
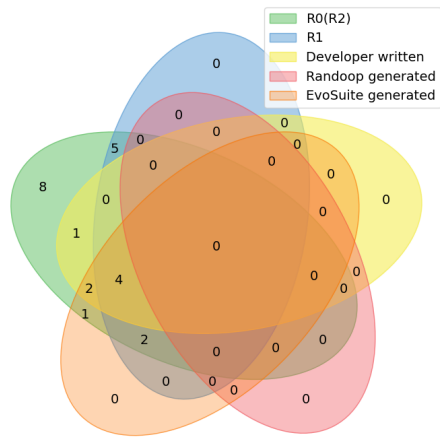
**(y) jopenfst**
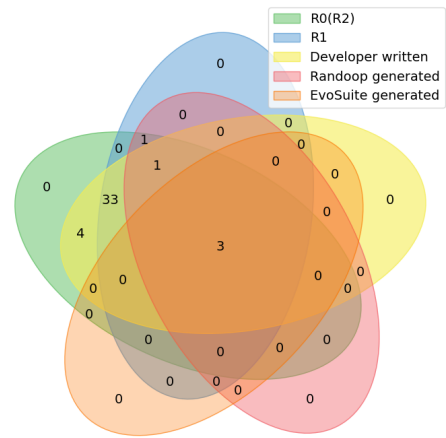
**(z) messageml-utils**

**(aa) gerrit-rest-java-client**

**(ab) visearch-sdk-java**

(ac) nfe



(ad) ognl

Figure 2: Sets of mutants killed by inline tests and unit tests.

# B  STUDY OF MUTANTS KILLED BY INLINE TESTS

This section reports our findings from manually inspecting the mutants killed by inline tests but not by unit tests (developer-written and Randoop-generated ones).

We randomly sampled 63 examples to inspect from 710 cases where inline tests kill mutants, but unit tests do not. The reasons for inline tests being able to kill more mutants than unit tests (where they are extracted from) can be grouped into three categories:
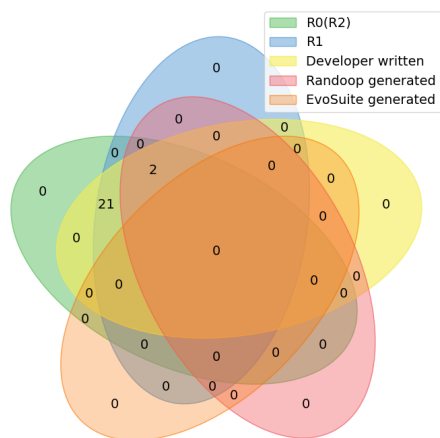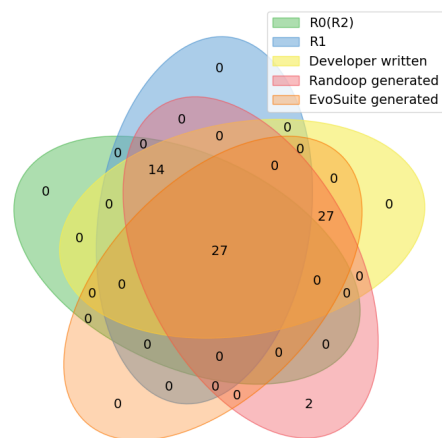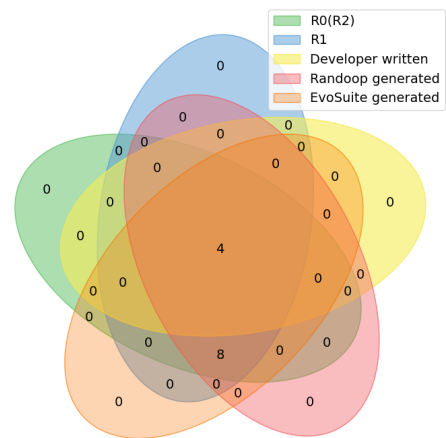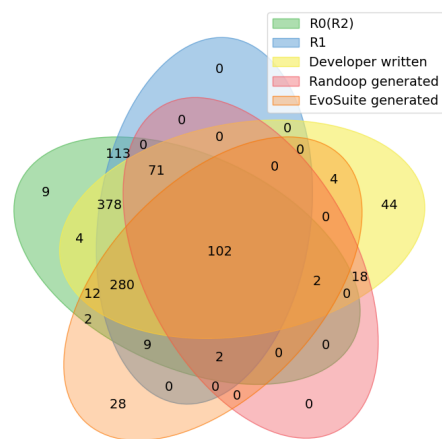
- Unit tests lack of good assertions (77.8%).

  Some unit tests may only assert some attribute of an object, and the attribute affected by the mutation may not be checked by unit tests. Randoop-generated tests typically have weak assertions, e.g., checking if an object is null or not null. Example:

  Method under test:

```
1  public ItemRequest<JsonElement> deleteProjectBrief(String projectBriefGid, List<String> optFields, Boolean optPretty) throws IOException {
2      String path = "/project_briefs/{project_brief_gid}".replace("{project_brief_gid}", projectBriefGid);
3      // mutated code: String path = "/project_briefs/{project_brief_gid}".replace(projectBriefGid, "{project_brief_gid}");
4      ItemRequest<JsonElement> req = new ItemRequest<JsonElement>(this, JsonElement.class, path, "DELETE").query("opt_pretty", optPretty).query("opt_fields", optFields);
5      return req;
6  }
```

  Randoop-generated unit test:

```
1  @Test
2  public void test1042() throws Throwable {
3      ...
4      com.asana.requests.ItemRequest<com.google.gson.JsonElement> jsonElementItemRequest95 = projectBriefsBase14.deleteProjectBrief("com.asana.errors.RetryableAsanaError: Not Found", (java.util.List<java.lang.String>) strList86, (java.lang.Boolean) true);
5      org.junit.Assert.assertNotNull(jsonElementItemRequest95);
6      ...
7  }
```

  Inline tests:

```
1  public ItemRequest<JsonElement> deleteProjectBrief(String projectBriefGid, List<String> optFields, Boolean optPretty) throws IOException {
2      String path = "/project_briefs/{project_brief_gid}".replace("{project_brief_gid}", projectBriefGid);
3      new Here("Randoop", 53).given(projectBriefGid, "com.asana.errors.RetryableAsanaError: Not Found").checkEq(path, "/project_briefs/com.asana.errors.RetryableAsanaError: Not Found");
4      new Here("Randoop", 53).given(projectBriefGid, "200").checkEq(path, "/project_briefs/200");
5      ItemRequest<JsonElement> req = new ItemRequest<JsonElement>(this, JsonElement.class, path, "DELETE").query("opt_pretty", optPretty).query("opt_fields", optFields);
6      return req;
7  }
```

- Inline tests perform "local" assertions (20.6%).

  Given the inputs provided by the unit tests, the program state that would have changed by the mutation are limited to local variables and thus not observable from outside the method. In other words, no assertion in the unit tests can kill such mutants. On the contrary, inline tests allow adding "local" assertions to check the changed program state.

  Using different test inputs in the unit tests may help. In the example below, the test inputs from all existing developer-written and Randoop-generated unit tests would set unresolved to "${p2}", "value", "${malformed", "$http : //${hostname} : ${port}/", "localhost : ${port}/", "${system.property}", "", none of which can make the return value change after mutation. A test input that sets unresolved to "value}" would kill the mutant, but it is not trivial to find such input in all cases.

  Example:

  Method under test:

```
1  public boolean hasMoreLegalPlaceholders()
2  {
3      int prefixPos = unresolved.indexOf( "${" );
4      // mutated code: int prefixPos = unresolved.indexOf( "" );
5      int suffixPos = unresolved.indexOf( "}", prefixPos + 2 );
6      return prefixPos >= 0 && suffixPos >= 0;
7  }
```

  Developer-written unit test:

```
1  @Test
2  public void multipleValuesAreResolved()
3  {
4      Properties properties = new Properties();
5      properties.setProperty( "hostname", "localhost" );
6      properties.setProperty( "port", "8080" );
7      properties.setProperty( "base.url", "http://${hostname}:${port}/" );
8      String value = resolver.getPropertyValue( "base.url", properties, new Properties() );
9      assertEquals( "http://localhost:8080/", value );
10 }
```

  Inline tests:

```
1  public boolean hasMoreLegalPlaceholders()
2  {
3      int prefixPos = unresolved.indexOf( "${" );
4      new Here("Unit", 35).given(unresolved, "localhost:${port}/").checkEq(prefixPos, 10);
5      int suffixPos = unresolved.indexOf( "}", prefixPos + 2 );
6      return prefixPos >= 0 && suffixPos >= 0;
7  }
```

- Performance bug not detectable by unit tests (1.6%).

  The mutant makes the program less efficient but does not change the semantics of the program (in the following example, mutated condition is always true as long as body is not an empty string, but the then-branch of the if-statement still only replaces ".." to "."). However, with inline tests, we can directly check if the condition is triggered when it should not, and find such performance bug.

  Example:

  Method under test:

```
1  protected String generateSetter(OgnlContext context, CtClass newClass, CtClass objClass, ClassPool pool,
2      CtMethod valueSetter, Node expression, Object root) throws Exception
3      {
4      ...
5      createLocalReferences(context, pool, newClass, objClass, valueSetter.getParameterTypes());
6      body = "{" + (castExpression != null ? castExpression : "") + pre + setterCode + ";}";
7      if (body.indexOf("..") >= 0) {
8      // mutated code: if (body.indexOf("") >= 0) {
9          body = body.replaceAll("\\.\\.", ".");
10      }
11      ...
12  }
```

  Developer-written unit test:

```
1  public SetterTest(String name, Object root, String expressionString, Object expectedResult, Object setValue)
2  {
3          super(name, root, expressionString, expectedResult, setValue);
4  }
```

  Inline tests:

```
1  protected String generateSetter(OgnlContext context, CtClass newClass, CtClass objClass, ClassPool pool,
2      CtMethod valueSetter, Node expression, Object root) throws Exception
3      {
4      ...
5      createLocalReferences(context, pool, newClass, objClass, valueSetter.getParameterTypes());
6      body = "{" + (castExpression != null ? castExpression : "") + pre + setterCode + ";}";
7      if (body.indexOf("..") >= 0) {
8          new Here("Unit", 639).given(body, "{new java.lang.Boolean(org.ognl.test.MethodTest#getA().isProperty());}").checkFalse(group());
9          new Here("Unit", 639).given(body, "{new java.lang.Integer(((java.lang.Object)$2)..hashCode());}").checkTrue(group());
10          body = body.replaceAll("\\.\\.", ".");
11      }
12      ...
13  }
```

Table 2 lists all examples we inspected and their reasons.

**Table 2: R0 tests killed mutants that are missed by developer-written unit tests.**

| Target stmt | Original code | Mutated code | Reason |
|---|---|---|---|
| java-asana CustomFields 100 | `String path = String.format("/custom_fields/%s/enum_options", customField);` | `String path = String.format("", customField);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana Projects 129 | `String path = String.format("/workspaces/%s/projects", workspace);` | `String path = String.format("", workspace);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNull/assertNotNull. |
| java-asana Projects 45 | `String path = String.format("/workspaces/%s/projects", workspace);` | `String path = String.format("", workspace);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNull/assertNotNull. |
| java-asana Projects 181 | `String path = String.format("/projects/%s/removeFollowers", project);` | `String path = String.format( project,"/projects/%s/removeFollowers");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana Projects 154 | `String path = String.format("/projects/%s/tasks", project);` | `String path = String.format( project,"/projects/%s/tasks");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana Sections 123 | `String path = String.format("/projects/%s/sections/insert", project);` | `String path = String.format("", project);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana Sections 104 | `String path = String.format("/sections/%s/addTask", section);` | `String path = String.format("", section);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |

| | | | |
|---|---|---|---|
| java-asana Stories 50 | `String path = String.format("/tasks/%s/ stories", task);` | `String path = String.format("", task);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull, fail. |
| java-asana Tags 46 | `String path = String.format("/workspaces/% s/tags", workspace);` | `String path = String.format("", workspace) ;` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNull/assertNotNull. |
| java-asana Tasks 212 | `String path = String.format("/tasks/%s/ dependencies", task);` | `String path = String.format("", task);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNull/assertNotNull. |
| java-asana Tasks 188 | `String path = String.format("/tags/%s/ tasks", tag);` | `String path = String.format("", tag);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNull/assertNotNull. |
| java-asana Tasks 413 | `String path = String.format("/tasks/%s/ subtasks", task);` | `String path = String.format("", task);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNull/assertNotNull. |
| java-asana Tasks 48 | `String path = String.format("/workspaces/% s/tasks", workspace);` | `String path = String.format( workspace,"/ workspaces/%s/tasks");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana Tasks 200 | `String path = String.format("/workspaces/% s/tasks/search", workspace);` | `String path = String.format( workspace,"/ workspaces/%s/tasks/search");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana Teams 61 | `String path = String.format("/teams/%s/ users", team);` | `String path = String.format("", team);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |

| | | | |
|---|---|---|---|
| java-asana<br>UserTaskLists<br>24 | `String path = String.format("/users/%s/`<br>`user_task_list", user);` | `String path = String.format("", user);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana<br>Users<br>50 | `String path = String.format("/users/%s/`<br>`favorites", user);` | `String path = String.format( user,"/users`<br>`/%s/favorites");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana<br>Workspaces<br>90 | `String path = String.format("/workspaces/%`<br>`s/typeahead", workspace);` | `String path = String.format("", workspace)`<br>`;` | Developer written tests not covered. Randoop tests covered. Randoop tests no assertions. |
| java-asana<br>AttachmentsBase<br>32 | `String path = "/attachments/{`<br>`attachment_gid}".replace("{`<br>`attachment_gid}", attachmentGid);` | `String path = "/attachments/{`<br>`attachment_gid}".replace(`<br>`attachmentGid,"{attachment_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana<br>AttachmentsBase<br>78 | `String path = "/tasks/{task_gid}/`<br>`attachments".replace("{task_gid}",`<br>`taskGid);` | `String path = "/tasks/{task_gid}/`<br>`attachments".replace( taskGid,"{`<br>`task_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana<br>CustomFieldSettingsBase<br>59 | `String path = "/projects/{project_gid}/`<br>`custom_field_settings".replace("{`<br>`project_gid}", projectGid);` | `String path = "/projects/{project_gid}/`<br>`custom_field_settings".replace("",`<br>`projectGid);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana<br>CustomFieldsBase<br>128 | `String path = "/workspaces/{workspace_gid`<br>`}/custom_fields".replace("{`<br>`workspace_gid}", workspaceGid);` | `String path = "/workspaces/{workspace_gid`<br>`}/custom_fields".replace(`<br>`workspaceGid,"{workspace_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |

| | | |
|---|---|---|
| java-asana CustomFieldsBase 104 | `String path = "/custom_fields/{`<br>`    custom_field_gid}".replace("{`<br>`    custom_field_gid}", customFieldGid);` | `String path = "/custom_fields/{`<br>`    custom_field_gid}".replace(`<br>`    customFieldGid,"{custom_field_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana PortfoliosBase 73 | `String path = "/portfolios/{portfolio_gid`<br>`    }/addMembers".replace("{portfolio_gid}`<br>`    ", portfolioGid);` | `String path = "/portfolios/{portfolio_gid`<br>`    }/addMembers".replace("",`<br>`    portfolioGid);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana ProjectBriefsBase 31 | `String path = "/projects/{project_gid}/`<br>`    project_briefs".replace("{project_gid}`<br>`    ", projectGid);` | `String path = "/projects/{project_gid}/`<br>`    project_briefs".replace( projectGid,"`<br>`    {project_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana ProjectBriefsBase 53 | `String path = "/project_briefs/{`<br>`    project_brief_gid}".replace("{`<br>`    project_brief_gid}", projectBriefGid)`<br>`    ;` | `String path = "/project_briefs/{`<br>`    project_brief_gid}".replace(`<br>`    projectBriefGid,"{project_brief_gid}"`<br>`    );` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana ProjectMembershipsBase 31 | `String path = "/project_memberships/{`<br>`    project_membership_gid}".replace("{`<br>`    project_membership_gid}",`<br>`    projectMembershipGid);` | `String path = "/project_memberships/{`<br>`    project_membership_gid}".replace("",`<br>`    projectMembershipGid);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana ProjectStatusesBase 99 | `String path = "/projects/{project_gid}/`<br>`    project_statuses".replace("{`<br>`    project_gid}", projectGid);` | `String path = "/projects/{project_gid}/`<br>`    project_statuses".replace( projectGid,`<br>`    "{project_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNull/assertNotNull. |

1708 1707 1706 1705 1704 1703 1702 1701 1700 1699 1698 1697 1696 1695 1694 1693 1692 1691 1690 1689 1688 1687 1686 1685 1684 1683 1682 1681 1680 1679 1678 1677 1676 1675 1674 1673 1672 1671 1670 1669 1668 1667 1666 1665 1664 1663 1662 1661

Extracting Inline Tests from Unit Tests

| | | | |
|---|---|---|---|
| java-asana ProjectsBase 116 | `String path = "/teams/{team_gid}/projects". replace("{team_gid}", teamGid);` | `String path = "/teams/{team_gid}/projects". replace( teamGid,"{team_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNull/assertNotNull. |
| java-asana ProjectsBase 30 | `String path = "/projects/{project_gid}/ addCustomFieldSetting".replace("{ project_gid}", projectGid);` | `String path = "/projects/{project_gid}/ addCustomFieldSetting".replace( projectGid,"{project_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana StatusUpdatesBase 78 | `String path = "/status_updates/{status_gid} ".replace("{status_gid}", statusGid);` | `String path = "/status_updates/{status_gid} ".replace("", statusGid);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana StoriesBase 53 | `String path = "/stories/{story_gid}". replace("{story_gid}", storyGid);` | `String path = "/stories/{story_gid}". replace("", storyGid);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana TagsBase 102 | `String path = "/tags/{tag_gid}".replace("{ tag_gid}", tagGid);` | `String path = "/tags/{tag_gid}".replace( tagGid,"{tag_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana TasksBase 75 | `String path = "/tasks/{task_gid}/ addFollowers".replace("{task_gid}", taskGid);` | `String path = "/tasks/{task_gid}/ addFollowers".replace( taskGid,"{ task_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana TasksBase 633 | `String path = "/workspaces/{workspace_gid }/tasks/search".replace("{ workspace_gid}", workspaceGid);` | `String path = "/workspaces/{workspace_gid }/tasks/search".replace( workspaceGid, "{workspace_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |

17

ISSTA 2023, 17-21 July, 2023, Seattle, USA

1756 1755 1754 1753 1752 1751 1750 1749 1748 1747 1746 1745 1744 1743 1742 1741 1740 1739 1738 1737 1736 1735 1734 1733 1732 1731 1730 1729 1728 1727 1726 1725 1724 1723 1722 1721 1720 1719 1718 1717 1716 1715 1714 1713 1712 1711 1710 1709

| | | | |
|---|---|---|---|
| java-asana<br>TasksBase<br>230 | `String path = "/tasks/{task_gid}/`<br>`    dependencies".replace("{task_gid}",`<br>`    taskGid);` | `String path = "/tasks/{task_gid}/`<br>`    dependencies".replace( taskGid,"{`<br>`    task_gid}");` | Developer written tests not covered.<br>Randoop tests covered. Randoop tests<br>assertNotNull. |
| java-asana<br>TasksBase<br>306 | `String path = "/tasks/{task_gid}".replace(`<br>`    "{task_gid}", taskGid);` | `String path = "/tasks/{task_gid}".replace(`<br>`    taskGid,"{task_gid}");` | Developer written tests not covered.<br>Randoop tests covered. Randoop tests<br>assertNotNull. |
| java-asana<br>TasksBase<br>206 | `String path = "/tasks/{task_gid}/duplicate`<br>`    ".replace("{task_gid}", taskGid);` | `String path = "/tasks/{task_gid}/duplicate`<br>`    ".replace( taskGid,"{task_gid}");` | Developer written tests not covered.<br>Randoop tests covered. Randoop tests<br>assertNotNull. |
| java-asana<br>TasksBase<br>729 | `String path = "/tasks/{task_gid}".replace(`<br>`    "{task_gid}", taskGid);` | `String path = "/tasks/{task_gid}".replace(`<br>`    taskGid,"{task_gid}");` | Developer written tests not covered.<br>Randoop tests covered. Randoop tests<br>assertNotNull. |
| java-asana<br>TasksBase<br>256 | `String path = "/tasks/{task_gid}/`<br>`    dependents".replace("{task_gid}",`<br>`    taskGid);` | `String path = "/tasks/{task_gid}/`<br>`    dependents".replace( taskGid,"{`<br>`    task_gid}");` | |
| java-asana<br>TeamMembershipsBase<br>31 | `String path = "/team_memberships/{`<br>`    team_membership_gid}".replace("{`<br>`    team_membership_gid}",`<br>`    teamMembershipGid);` | `String path = "/team_memberships/{`<br>`    team_membership_gid}".replace(`<br>`    teamMembershipGid,"{`<br>`    team_membership_gid}");` | Developer written tests not covered.<br>Randoop tests covered. Randoop tests<br>assertNull/assertNotNull. |
| java-asana<br>TeamsBase<br>106 | `String path = "/organizations/{`<br>`    workspace_gid}/teams".replace("{`<br>`    workspace_gid}", workspaceGid);` | `String path = "/organizations/{`<br>`    workspace_gid}/teams".replace("",`<br>`    workspaceGid);` | Developer written tests not covered.<br>Randoop tests covered. Randoop tests<br>assertNotNull. |

| | | | |
|---|---|---|---|
| java-asana<br>TeamsBase<br>31 | `String path = "/teams/{team_gid}/addUser".`<br>`replace("{team_gid}", teamGid);` | `String path = "/teams/{team_gid}/addUser".`<br>`replace("", teamGid);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana<br>TeamsBase<br>133 | `String path = "/users/{user_gid}/teams".`<br>`replace("{user_gid}", userGid);` | `String path = "/users/{user_gid}/teams".`<br>`replace("", userGid);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana<br>TeamsBase<br>80 | `String path = "/teams/{team_gid}".replace(`<br>`"{team_gid}", teamGid);` | `String path = "/teams/{team_gid}".replace(`<br>`"", teamGid);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana<br>UserTaskListsBase<br>54 | `String path = "/users/{user_gid}/`<br>`user_task_list".replace("{user_gid}",`<br>`userGid);` | `String path = "/users/{user_gid}/`<br>`user_task_list".replace( userGid,"{`<br>`user_gid}");` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| java-asana<br>UsersBase<br>33 | `String path = "/users/{user_gid}/favorites`<br>`".replace("{user_gid}", userGid);` | `String path = "/users/{user_gid}/favorites`<br>`".replace("", userGid);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull/org.junit.Assert.fail(...) |
| java-asana<br>WebhooksBase<br>125 | `String path = "/webhooks/{webhook_gid}".`<br>`replace("{webhook_gid}", webhookGid);` | `String path = "/webhooks/{webhook_gid}".`<br>`replace("", webhookGid);` | Developer written tests not covered. Randoop tests covered. Randoop tests assertNotNull. |
| fabric-sdk-java<br>IdemixRoles<br>47 | `mask = mask | getIdemixRoleFromMSPRole(`<br>`role);` | `mask = mask & getIdemixRoleFromMSPRole(`<br>`role);` | Developer written tests covered. Randoop tests not covered. org.hyperledger.fabric.sdk.identity.IdemixIdentitiesTest. variable value does not change. no good inputs. |

| | | | |
|---|---|---|---|
| fabric-sdk-java<br>TLSCertificateBuilder<br>113 | `KeyUsage usage = new KeyUsage(KeyUsage.`<br>`    keyEncipherment | KeyUsage.`<br>`    digitalSignature);` | `KeyUsage usage = new KeyUsage(KeyUsage.`<br>`    keyEncipherment & KeyUsage.`<br>`    digitalSignature);` | Developer written tests covered. Randoop tests not covered. org.hyperledger.fabric.sdk.NetworkConfigTest, no assertions to diff, before mutant: critical(false) 2.5.29.15 value = *****, after mutnat: critical(false) KeyUsage: 0xa0 |
| ognl<br>IntHashMap<br>201 | `int index = ( key & 0x7FFFFFFF ) % table.`<br>`    length;` | `int index = ( key | 0x7FFFFFFF ) % table.`<br>`    length;` | Developer written tests not covered. Randoop tests covered. org.junit.Assert.fail |
| ognl<br>ListPropertyAccessor<br>232 | `if (indexStr.indexOf('"') >= 0)` | `if (!(indexStr.indexOf('"') >= 0))` | Developer written tests covered. Randoop tests covered. org.ognl.test.CollectionDirectPropertyTest. method's return value does not change with the mutant. no good inputs. |
| ognl<br>OgnlParser<br>3209 | `if ((jj_la1_1[i] & (1<<j)) != 0) {` | `if ((jj_la1_1[i] & ((1+1)<<j)) != 0) {` | Developer written tests covered. Randoop tests covered. org.ognl.test.enhance.TestExpressionCompiler, org.ognl.test.ArrayCreationTest, the changed variable value (the variable is about error message) is not asserted. No tests for the changed field. no assertions. |
| ognl<br>OgnlParserTokenManager<br>398 | `if ((active0 & 0x100L) != 0L)` | `if ((active0 & 0x100L) >= 0L)` | Developer written tests covered. Randoop tests not covered. org.ognl.test.ArithmeticAndLogicalOperatorsTest, method's return value does not change with the mutant. no good inputs. |
| ognl<br>OgnlParserTokenManager<br>1132 | `long l2 = 1L << (curChar & 077);` | `long l2 = 1L << (curChar & 1);` | Developer written tests covered. Randoop tests not covered. org.ognl.test.ConstantTest, method's return value does not change with the mutant. no good inputs. |

| | | | |
|---|---|---|---|
| ognl<br>OgnlParserTokenManager<br>655 | `if ((0x3ff000000000000L & l) != 0L)` | `if ((0x1ff000000000000L & l) != 0L)` | Developer written tests covered. Randoop tests covered. org.ognl.test.PrimitiveArrayTest mutant does not change variable's value, jjroundss. no good inputs. |
| ognl<br>OgnlParserTokenManager<br>730 | `if ((0x3ff000000000000L & l) != 0L)` | `if ((0x0ff000000000000L & l) != 0L)` | Developer written tests covered. Randoop tests covered. org.ognl.test.PrimitiveArrayTest mutant does not change variable's value, jjroundss. no good inputs. |
| ognl<br>ExpressionCompiler<br>639 | `if (body.indexOf("..") >= 0)` | `if (body.indexOf("") >= 0)` | Developer written tests covered. Randoop tests not covered. Mutant does not change the semantics of the program. |
| geoip-api-java<br>LookupService<br>304 | `if ((dboptions & GEOIP_CHECK_CACHE) != 0) {` | `if ((dboptions & GEOIP_CHECK_CACHE) == 0) {` | Developer written tests covered. Randoop tests not covered. com.maxmind.geoip.CityLookupTest. In if statement, the value of variable mtime changed from 0 to 1674327714003, but it is never used. no assertions. |
| properties-maven-plugin<br>ExpansionBuffer<br>35 | `int prefixPos = unresolved.indexOf( "${" );` | `int prefixPos = unresolved.indexOf( "" );` | Developer written tests covered. Randoop tests not covered. mvn test -Dsurefire.redirectTestOutputToFile=False, no good inputs. |
| mp3agic<br>ID3v2ChapterTOCFrameData<br>45 | `if ((flags & 0x02) == 0x02) {` | `if ((flags & 0x02) <= 0x02) {` | Developer written tests covered. Randoop tests covered. com.mpatric.mp3agic.ID3v2TagTest, com.mpatric.mp3agic.ID3v2ChapterTOCFrameDataTest, no good inputs. |
| jopenfst<br>IndexWeight<br>66 | `result = 31 * result + (int) (temp ^ (temp >>> 32));` | `result = 0 * result + (int) (temp ^ (temp >>> 32));` | Developer written tests covered. Randoop tests not covered. com.github.steveash.jopenfst.operations.NShortestPathsTest, com.github.steveash.jopenfst.IndexWeightTest, hashCode() does not change. no good inputs. |

| nfe DFXMLValidador 112 | ```final URL xsdPath = DFXMLValidador.class. getClassLoader().getResource(String. format("schemas/PL_CTeDistDFe_100/%s", xsd));``` | ```final URL xsdPath = DFXMLValidador.class. getClassLoader().getResource(String. format("", xsd));``` | Developer written tests not covered. Randoop tests covered. Randoop tests org.junit.Assert.fail(...) |
|---|---|---|---|
| nfe DFXMLValidador 15 | ```final URL xsdPath = DFXMLValidador.class. getClassLoader().getResource(String. format("schemas/PL_008i2/%s", xsd));``` | ```final URL xsdPath = DFXMLValidador.class. getClassLoader().getResource(String. format("", xsd));``` | Developer written tests not covered. Randoop tests covered. Randoop tests org.junit.Assert.fail(...) org.junit.Assert.fail("Expected exception of type org.xml.sax.SAXParseException; message: Content is not allowed in prolog."); |

# REFERENCES

[1] Sebastian Schweikl, Gordon Fraser, and Andrea Arcuri. 2022. EvoSuite at the SBST 2022 Tool Competition. In *SBST*. 33–34.