Replicate the study "What Programmers Do with Inheritance in Java" (1) to verify its results.

Research Questions

How are the results of our replication study with respect to the research questions of the original study? How often do we see:

- Late-bound self-reference
- Subtyping
- Down-calling
- External reuse
- Other reuse

Project Overview

- The original study proposes a model for usage of defined inheritance relationships in Java.
- Using this model, the authors analyzed the byte code of 93 open source Java projects from Qualitas Corpus (2).
- They found that defined inheritance relationships are also used intensively.
- We replicated this study on Java source code.
- We analyzed 90 projects from Qualitas.class Corpus (3).
- Our results are similar to the original results, but they are not the same.

Inheritance Usage - Examples

```java
public class P {
    void p() {
        q(); // down-call
    }
    void q() {}
    void a() {}
}
public class N {
    void n() {
        P ap = new C(); // subtype
        C ac = new C();
        ac.a(); // external reuse
    }
}
```

Differences in Study Set-up

<table>
<thead>
<tr>
<th>Original</th>
<th>Replication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java byte code</td>
<td>Java source code</td>
</tr>
<tr>
<td>Qualitas Corpus (2)</td>
<td>Qualitas.class Corpus (3)</td>
</tr>
</tbody>
</table>

Replication - Implementation

- Build Java source AST's with Rascal.
- Visit the relevant AST nodes and use the Rascal M3 model
- Bring the results together for different metrics and put them in Rascal relations.

Replication - Major Limitations

- 25 out of 90 projects have different versions in Qualitas.class Corpus. For the projects with same versions, the content of source and byte code distributions are quite different.
- Limitation about analysis of methods defined outside of the project: fewer cases for subtype and external reuse.
- We suspect some false positives for down-call and external reuse in the original study, but there are not many of them. We suspect that byte code analysis can be misleading in these cases.

Conclusion

- Our results verify the results of the original study for all research questions to a great extent.
- Most differences are the result of differences between the analyzed projects and our analysis limitation about external methods.
- We suspect some false positives for down-call and external reuse in the original study, but there are not many of them. We suspect that byte code analysis can be misleading in these cases.

Works Cited