SMA:
Software Modeling and Analysis

Practical Session

Week 08

Assignment 08

Discussion

*You have to attend the lecture to reveal such slides.**

:-)

**Disclaimer:*

The content that has been shown on this slide is irrelevant for the exam.

A08 - Exercise 01 | Completion of Method

Task 1: Java *coding*.

Your task is to implement InternalInvocationAnalysis#process() that is applied on all methods defined within the analyzed application. ***Look at the class and its comments to get an idea on what is left to do.***

a) Submit the code of the method

b) Submit the output of the analysis

A08 - Exercise 01 | Completion of Method

a) Submit the code of the method

```
for (Unit unit :  body.getUnits()) {
  Stmt stmt = (Stmt) unit;

  if (!stmt.containsInvokeExpr()) {
    continue;
  }

  InvokeExpr invokeExpr = stmt.getInvokeExpr();
  SootMethod method = invokeExpr.getMethod();

  if (!this.isInternal(method.getDeclaringClass())) {
    continue;
  }

  int i = internalInvocations.getOrDefault(method, 0);
  internalInvocations.put(method, i + 1);
}
```

A08 - Exercise 01 | Completion of Method

b) Submit the output of the analysis

```
Analyzing /Users/maenu/.m2/repository/com/google/guava/guava/18.0/guava-18.0.jar

Soot started on Tue Nov 06 14:34:19 CET 2018

Soot finished on Tue Nov 06 14:34:21 CET 2018

Soot has run for 0 min.  2 sec.

872 <com.google.common.base.Preconditions:  java.lang.Object checkNotNull(java.lang.Object)>

162 <com.google.common.base.Preconditions:  void checkArgument(boolean,java.lang.String,java.lang.Object[])>

91 <com.google.common.base.Preconditions:  void checkArgument(boolean)>

75 <com.google.common.collect.Ordering:  com.google.common.collect.Ordering natural()>

57 <com.google.common.net.MediaType:  com.google.common.net.MediaType createConstant(java.lang.String,java.lang.String)>

54 <com.google.common.base.Objects:  boolean equal(java.lang.Object,java.lang.Object)>

49 <com.google.common.io.Closer:  void close()>

46 <com.google.common.cache.LocalCache$Segment:  void postWriteCleanup()>

45 <com.google.common.collect.Multiset$Entry:  java.lang.Object getElement()>

44 <com.google.common.util.concurrent.Monitor:  void leave()>
```

A08 - Exercise 01 | Inspection of results

Task 2: Java code *reading*.

Analyze the output of InternalInvocationAnalysis. Inspect the documentation and code of the 10 most used methods.

a) Submit a short summary of your inspection results.

A08 - Exercise 01 | Inspection of results

a) Submit a short summary of your inspection results.

- Guava most frequently performs precondition null and argument checks
- Consequently, Guava is mostly busy with itself ensuring that preconditions are met when methods or constructors get called
- Ordering is also heavily used throughout the library

Assignment 09

Preview

A09 - Exercise 01 | Roassal Visualization

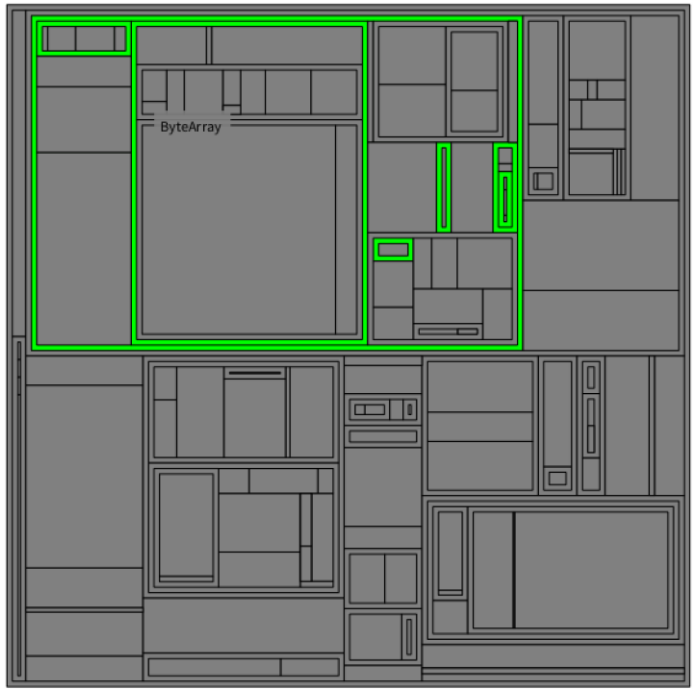
Smalltalk *coding*.

Build a *sunburst visualization* to analyze test coverage of the `Collection` class hierarchy.

A09 - Exercise 02 | Roassal Visualization

Smalltalk *coding*.

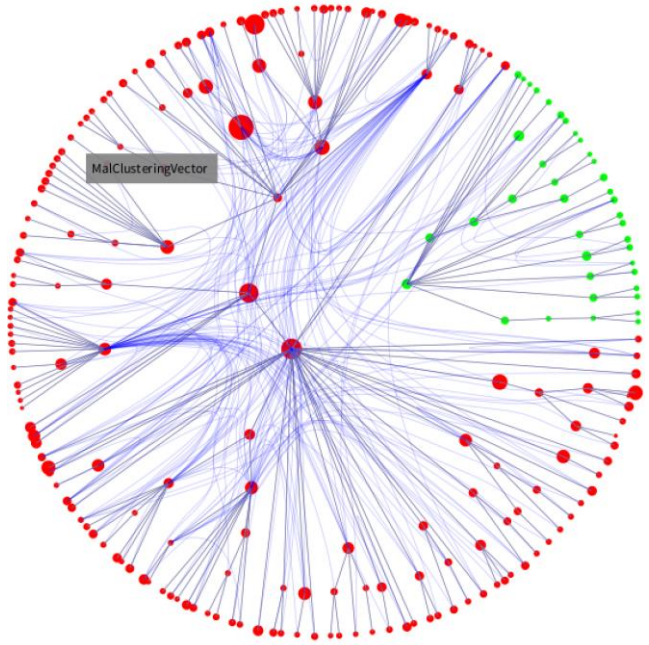
Build a *treemap visualization* to gather an overview of classes that have subclasses, and contain the string `Array` in their names.



A09 - Exercise 03 | Roassal Visualization

Smalltalk *coding*.

Create a visualization using the Mondrian builder to analyze the class dependencies between the `Collection` class hierarchy and the `RTLayout` class hierarchy.



A09 - Exercise 04 | Discussion

Visualization *reasoning*.

Comment on the strenghts and limitations of each visualization you just created.