Prof. Dr. Oscar Nierstrasz
Pascal Gadient, Pooja Rani

Assignment 08 — 04.11.2020 – v1.0
Code and Test Smells

Please submit this exercise by email to pascal.gadient@inf.unibe.ch before 11 November 2020, 10:15am.

You must submit your code as editable text, i.e., use plain text file(s).

For the second part of this exercise, we use the latest GT release from here. If you already imported some Moose (MSE) models from the previous assignments into your GT environment, please start with a fresh copy of GT to reduce the strains on your computer's random access memory.

First, we have to download and extract the *Weka* dataset, and second, we need to import it into GT. We can perform both tasks using GT's Playground. Be warned: this process will take several minutes depending on your device's CPU and internet connection. We strongly advise you to save the image when the process succeeded to avoid redoing these steps.

The datasets can be downloaded and extracted with the following script:

```
targetFolder := (FileLocator imageDirectory asFileReference / 'models')
ensureCreateDirectory.
archiveFileName := 'weka-3-8.zip'.
archiveUrl := 'https://dl.feenk.com/moose-tutorial/weka/'.
ZnClient new
  url: archiveUrl, archiveFileName;
  signalProgress: true;
  downloadTo: targetFolder.
(ZipArchive new
  readFrom: targetFolder / archiveFileName)
  extractAllTo: targetFolder.
```

The sample dataset can be imported with the following script:

```
modelFile := (FileLocator imageDirectory asFileReference / 'models')
  / 'weka-3-8'
  / 'weka-3-8.mse'.
modelWeka := MooseModel new
  importMSEFromFile: modelFile.
```

Exercise 1: Code smells (4 pts)

a) Choose two different code smells and answer the following questions for each of the smells: Which code smell did you chose? What are its characteristics? What is the resulting problem? (2 pts)

b) What is the fundamental problem in developers' code smell perception? (1 pt)

c) What is *association rule mining* in the context of the HIST code smell paper you can find here? (1 pt)

Exercise 2: Test code smells (3 pts)

a) Choose one test code smell (except "Eager Test") and answer the following questions: Which test code smell did you chose? What are its characteristics? What is the resulting problem? (1 pt)

b) Considering the code below, report in which line you can find the "Eager Test" code smell and explain why it represents a problem. (2 pts)

```
01:  public void testDataIsVariable() throws Throwable {
02:    JSTerm term = new JSTerm();
03:    term.makeVariable();
04:    term.add((Object) "");
05:    jSTerm0.matches(jSTerm0);
06:    assertEquals(false, term.isGround());
07:    assertEquals(true, term.isVariable());
08:  }
```

Exercise 3: Detection of eager tests (3 pts)

Your task is to extract all JUnit 3 tests from `modelWeka` that suffer from the "Eager Test" code smell. That is, you have to find every method with `#isJUnit3Test` set to `true` that contains an assertion statement at least two times.