Assignment 2

COSC 4313 3.0 Software Engineering Testing, Winter 2013

Due: Monday, February 18, 11am.

Format: Individual.

Testing with JUnit

The purpose of this assignment is to give you experience in creating automated test code with JUnit. Your task will be to create a test suite in JUnit for the non-GUI portion of BORG Calendar, produce bug reports, and submit a written report describing your testing.

To get started

- Create a directory called `winter2013`. We will refer to it as the *working directory*
- Download the BORG Calendar source code (see course webpage for link) to your working directory
- Download jgoodies-common and jgoodies-looks from `http://www.jgoodies.com` to your working directory
- Install the `lombok` package for Eclipse if you are using your personal computer (link on course website), or use `eclipse-lombok` on York University's computers.
- Create a workspace in the working directory
- Create a new Eclipse project using the "Importing existing project into workspace" wizard and configure the missing dependencies.
- Select Properties from the Project menu. Select `Run/Debug settings`. Click on `Edit...`. Under the Classpath tab, click on `Advanced...` and `Add Folders`. Select the `res` and click OK.
- Select Properties from the Project menu. Select `Java Build Path`. Select `Libraries` tab and click on `Add External JARs...` and select jgoodies-common and jgoodies-looks jar files (downloaded earlier).
- Launch BORG Calendar from Eclipse

What to do

You should select at least four non-trivial classes for your testing such as you can demonstrate at least one example of using boundary testing, equivalence class testing and decision table testing. (Note: You should not select classes with already developed test cases.) You should define a detailed specification for every selected class based on your understanding of the purpose of the class and then develop a test suite for the selected classes. You may use any testing strategy you believe is appropriate including the ones discussed in class. You are free to create as many test classes as you like.

For each implemented test case, develop a clear description of the test case. You should also provide a justification for the developed test case. Both test case description and test case justification will be a part of your report. If you believe that the implementation fails to meet the specification in some way, create bug reports. The bug reports will be part of your written report, so you can create them in whatever format you prefer.

Following is an example of specification for a Java class:

Specification

The `GraphGenerator` class contains the following two methods:

1. `public GraphGenerator (int numberOfNode, int maxDegree, boolean isConnected, String outputFileName)`

   The first argument is the number of nodes that the generated graph should have. The second argument is the maximum degree for any node.

   If the boolean argument is true, then the produced graph must be connected in the non-directed sense. If the boolean argument is false, then the graph may or may not be connected.

   Finally, the String argument is the name of the file that will contain the produced output in the XML format.

2. `public void createGraph() throws Exception`

   This method generates the random graph and outputs it in the designated file.

   If the number of nodes or the maximum degree is less than 1, then a `NegativeInputException` is thrown.

What to Submit

Before the deadline, submit electronically the test code package you created. To submit, navigate to the directory that contains the working directory (i.e. `winter2013`), and give a command like the following:

```
submit 4313 a2 winter2013
```

You also need to submit a written report. This report must include:

- Specification of the selected Java classes and justification of your selection
- A description of the testing strategies you applied, as well as the test cases you created. The marker will not read your code in order to see what you tested. You have to describe it.
- Show one detailed example of each testing technique that was discussed in class
- The bug reports you created.

Submit a hard copy of the written report to the 4313 drop box by the deadline. Attach the last page of this handout as the first page of the hard copy submission. Fill in your name and student number. Submit the same report electronically before the deadline as well (PDF format is preferable, but other formats will also be accepted). To submit, give:

```
submit 4313 a2 a2.pdf
```

where `a2.pdf` is your report. The marker will mark the hard copy submission. Electronic submissions will be used for remarking purposes.

Grading

This assignment will be marked out of 100. The marks will be divided as follows:

- Bug finding power of your testing: 20%
- Coverage of your testing: 20%
- Description of your testing approach: 40%
- Quality of bug reports: 10%
- Presentation/English: 10%

Presenting your thought processes in English is an important skill for a software professional. If you have trouble writing English, have somebody proof-read and correct your prose. You might find the services offered by The English as a Second Language Open Learning Centre useful: `http://www.yorku.ca/eslolc`

Assignment 2 Grade Sheet

COSC 4313 3.0 Software Engineering Testing, Section M, Winter 2013

Student Name:

Student Number:

Bug finding power . . . . ⌊______⌋

Coverage ⌊______⌋ . . . . ⌊______⌋

Description of testing approach . . . . ⌊______⌋

Quality of bug reports . . . . ⌊______⌋

Presentation / English . . . . ⌊______⌋

Total . . . . ⌊______⌋

Letter grade . . . . ⌊______⌋