CSCI 2134 Lab 3: Testing and JUnit

Fall 2023, Due: 11:59pm, Monday, October 9, 2023, submitted via Git

Objective

In this lab, you will produce test cases for some Java code and will implement those test cases using JUnit.

Preparation

1. Ensure that you have your Integrated Development Editor (IDE) installed.
2. Ensure that you are able to write and run a JUnit test case in the IDE. You will not have the time to both debug your IDE environment and complete the lab task within the lab time. The lab time should be used to concentrate on the lab task.
3. Make sure that you know how to take a screenshot of your computer screen.
4. Clone the Lab 3 repo: https://git.cs.dal.ca/courses/2023-fall/csci-2134/lab3/????.git where ???? is your CSID. Note that the primary branch is called "main" and not "master".
5. Review the course material on creating test cases.
6. Review the methods available for assertions in JUnit at https://junit.org/junit5/docs/current/api/org.junit.jupiter.api/org/junit/jupiter/api/Assertions.html Concentrate on the methods that let you provide a string to explain the case as one parameter.
7. Review the provided code listed in the Resources section below by reading the code. You may also wish to read the provided primer on matrix operations, found in the `docs` directory. Note: You do not need to understand all of it, but you do need to have seen it before starting the lab.

Resources

- JUnit assertions https://junit.org/junit5/docs/current/api/org.junit.jupiter.api/org/junit/jupiter/api/Assertions.html
- Primer on matrices in the `docs` directory of the Lab 3 repository.
- Code base to be reviewed is in the `src` directory of the Lab 3 repository.

Procedure

Set-up

1. Open the Matrix project, using IntelliJ, in the Lab 3 repository that you have cloned.
2. Open the target class `Matrix.java` file in the `src` subdirectory
3. Open the test class `MatrixTest.java` in the `test` subdirectory

Lab steps

1. Create a set of unit tests for the methods:

   ```
   Matrix add(Matrix b, Matrix res)
   Matrix multiplyWithScalar(double s, Matrix res)
   Matrix multiplyWithMatrix(Matrix b, Matrix res)
   ```

   Your tests should detect **at least 2** or more defects. You can stop once you reach 8 meaningful (and non-redundant) tests.
2. In `MatrixTest.java` Implement the tests. For each test, be sure to
   a. Properly name each test
   b. Have a description of each test's purpose in the comment block above the test.
   c. Identify if the test is a white box or black box test
   d. Include any reasonable assumptions that the test is making.
3. Execute your JUnit tests.

Reporting and analysis

4. In the `docs/errors.txt` file, list which tests failed. **Note:** you do not need to fix the errors. For each failed test, please include:
   a. Name of method being tested
   b. Name of test method
   c. Message produced by the test method
   d. Suggest a way that the bug revealed by the test can be fixed.
   e. Any additional comments or analysis that the test revealed.
5. If you have time, create additional tests for other methods.
6. **Commit** and **push** the test class that you updated. Remember to push to "main".

Grading

The lab will be marked out of 4 points:

Task	2 Points	1 Point	0 Points
Tests Completed	All required tests completed (for the three methods). The tests are relatively thorough.	Only one or two method were tested. Tests are not very thorough.	Little or no evidence of any reasonable tests implemented.
Bugs Found	Two or more bugs found	One bug found	No bugs found