CSIT314 Software Development Methodologies

Lab 5

This exercise is to introduce test-driven development with JUnit.

1. Download and install JUnit and Java environment (if you haven't got it). See this link for https://www.tutorialspoint.com/junit/index.htm for details. You can choose to use either JUnit 4 or JUnit 5.

2. Replicate the Calculator example as in the lecture slides. You can copy the code for Calculator.java, CalculatorTest.java and TestRunner.java as in the lecture slides. Compile and run the code to ensure that you see similar results.

3. Add a few more test cases: adding two negative numbers, adding a negative number and a positive number, etc. Note: write each testcase in a separate test function, e.g. testAddNegativeNumbers, testAddNegativePositiveNumbers, etc. Make sure you place the @Test annotation.

4. Continue test-driven development for the Calculator example by following the steps below:
   a. Step 1: Identify a new functionality for the Calculator: **multiplication**. Define a function multiply which takes as input two integer numbers in Calculator.java. Leave the body of the function empty (e.g. make it return 0 and do not write an implementation yet).

   b. Step 2: Write a few test cases for this functionality: e.g. multiply two positive numbers, two negative numbers, with 0, with 1, etc. Note: write each test case in a separate test function.

   c. Step 3: Compile and run the test cases and see the results.

   d. Step 4: Write an implementation of multiplication in its method body.

   e. Step 5: Compile and run the test cases and see the results.

5. Import the Calculator example into an IDE of your choice (e.g. IntelliJ, Eclipse, NetBeans, Visual Studio Code, etc.). Install JUnit into your IDE and run the example again.

   IntelliJ IDEA https://blog.jetbrains.com/idea/2016/08/using-junit-5-in-intellij-idea/
   Eclipse https://www.tutorialspoint.com/junit/junit_plug_with_eclipse.htm
   NetBeans https://netbeans.org/kb/docs/java/junit-intro.html
   Visual Studio Code https://code.visualstudio.com/docs/java/java-testing

6. Develop a similar Calculator example in another language that you are familiar with (e.g. .NET/C++, C++, PHP, Python, etc.

   NUnit is widely used unit-testing framework use for .NET languages https://nunit.org/
   PHPUnit: unit testing tool for PHP https://phpunit.de/

cppUnit: unit testing for C++

unittest: Python unit test

See a comprehensive list here:
https://en.wikipedia.org/wiki/List_of_unit_testing_frameworks