Name: Chun-Chi Huang
Course: Software Testing
Professor: Michael McKee
Date: 04/26/2018

How will You Be Able to Do Unit Testing

After studying several websites that discussed Unit Testing, I will summarize some principles for performing Unit Testing. The first principle is related to the procedures. It is called "Arrange-Act-Assert". Before executing Unit Testing, the pre-condition of the running environment should be set up, and then the main function that developers want to test is executed. Finally, developers check the result and compare it with the expectation. These steps should be included in each single Unit Testing. As for how to achieve the implementation, JUnit is a unit testing framework for the Java which uses annotations to identify methods that indicate a test. Also, Unit Testing should run short and fast. Developers should not implement a large test function with complex logics. Unit Testing should explicitly include "Arrange-Act-Assert" in a clear manner and make the test brief and direct.

Moreover, removing external dependencies is one of the important principles. Because Unit Testing focuses on small units and not the whole class or the entire system, developers should take away dependencies from other classes within the target test case. In real-world projects, the classes of one project usually have complicated interactions. This means that one class could be involved with several classes, and it also depends on the states of other classes to perform its task. Consequently, these situations result in a major challenge in Unit Testing for developers. A unit test should be performed in isolation, and side effects from other classes should be eliminated. Thus, in order to accomplish this task, a popular mock framework, Mockito, is created. It can work with JUnit and allows developers to create and configure mock objects. Utilizing Mockito can remove the external dependencies and simplify the tests for classes.

Furthermore, Unit Testing should be repeatable and timely. This means that every time the test cases are executed, the test results must be the same. This characteristic is very useful for developers to maintain the code base. This concept can also explain why the test implementation can be generated before the function. Before developers implement the function, they should already know the inputs, outputs, and logic of the function. According to this information, developers are eligible to create Unit Testing.

In conclusion, Unit Testing should be implemented succinctly and executed quickly, and it also should be repeatable and generated in a timely manner. The most important thing is that Unit Testing can be performed by using JUnit and Mockito to implement "Arrange-Act-Assert" and then achieve the requirements of Unit Testing.

Name: Chun-Chi Huang
Course: Software Testing
Professor: Michael McKee
Date: 04/26/2018

There are some websites relevant to Unit Testing.

https://developer.android.com/training/testing/unit-testing/local-unit-tests.html

http://www.vogella.com/tutorials/AndroidTesting/article.html#androidtesting_unittests

https://www.toptal.com/android/testing-like-a-true-green-droid

http://www.java2novice.com/junit-examples/

https://www.tutorialspoint.com/junit/index.htm

https://www.martinfowler.com/articles/mocksArentStubs.html

http://tutorials.jenkov.com/java-unit-testing/simple-test.html