Hands On Exercises

Unit test with JUnit and Mockito of a class that uses Counter

[Unit test of Counter Class 1](#_Toc31682)

[Unit test with Mockito of a class that uses Counter 2](#_Toc28990)

## Unit test of Counter Class

**Setup** : Import the maven project into Spring Tool Suite and run the Counter class.

**Requirement**: We have a Counter that always starts to count from 0 (0 is the minimum value) and that can increase and decrease a given number 

**Code**:

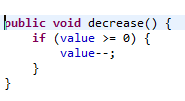
* src/main/java/Counter.java
* src/test/java/CounterTest.java 

We will see that:

▪ testIncrease() will succeed. The Counter increments correctly the number 10. The expected value (11) is equal to the actual value (11)

▪ testDecrease(), instead, will succeed initially, because we want a counter that will never have a value below 0.

Now change the testDecrease() method like so:



**Result** : The expected value (0) is NOT equal to the actual value (-1).

**Exercise 1:** Create some classes which extend Counter

▪ One class should count the number of even numbers less than or equal to the current value.

▪ One class should count the number of odd numbers less than or equal to the current value.

▪ One class should count the number of prime numbers less than or equal to the current value. 

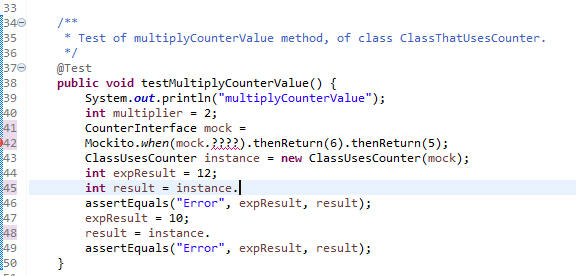
Which design pattern could we use? 

Test all these classes with JUnit

## Unit test with Mockito of a class that uses Counter

1. Enter the following code in the ClassUsesCounterTest.java test method. Then make the requested changes.

Exercise 2:



1. Create a mock object of the Counter class at line 41.
2. Replace the “????” with the correct method call at line 42.
3. Call the appropriate method on the “instance” reference variable at line 45 and 48.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*