# TDD using JUnit5 and Mockito

## Exercise 1: Setting Up JUnit

### CODE:

package javasample;

import org.junit.Test;

import static org.junit.Assert.\*;

public class Excerise1{

@Test

public void testAddition() {

int result = 2 + 3;

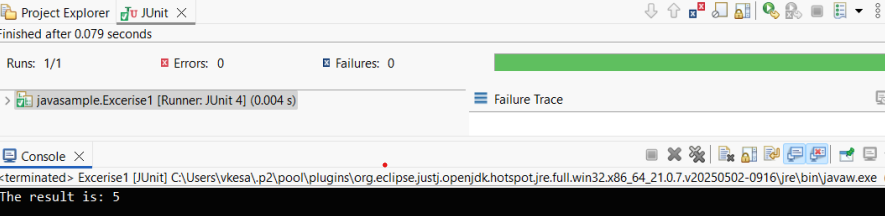
System.*out*.println("The result is: " + result);

*assertEquals*(5, result);

}

}

### OUTPUT:



## Exercise 3: Assertions in Junit

### CODE:

package javasample;

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

@Test

public void testAssertions() {

*assertEquals*(5, 2 + 3);

*assertTrue*(5 > 3);

*assertFalse*(5 < 3);

*assertNull*(null);

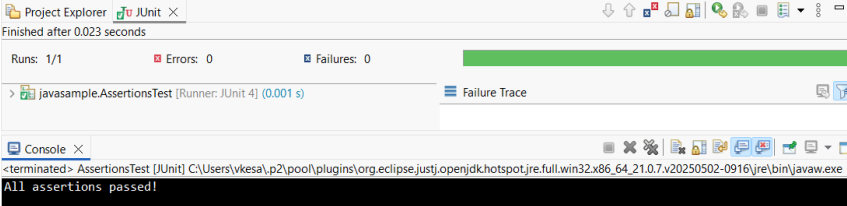
*assertNotNull*(new Object());

System.*out*.println("All assertions passed!");

}

}

### OUTPUT:



## Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in Junit

### CODE:

Calculator.java

package ArrangeActAssert;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

}

CalculatorTest.java

package ArrangeActAssert;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calculator;

@Before

public void setUp() {

System.*out*.println("Setting up test...");

calculator = new Calculator();

}

@After

public void tearDown() {

System.*out*.println("Cleaning up after test...");

calculator = null;

}

@Test

public void testAddition() {

int a = 5;

int b = 3;

int result = calculator.add(a, b);

*assertEquals*(8, result);

}

@Test

public void testSubtraction() {

int a = 10;

int b = 4;

int result = calculator.subtract(a, b);

*assertEquals*(6, result);

}

}

### OUTPUT:

