JUnit5 Hands On Exercises

# Exercise 1: Setting Up JUnit

CalculatorTest.java

package com.cognizant.junitdemo;

import static org.junit.Assert.*assertEquals*;

import org.junit.jupiter.api.Test;

public class CalculatorTest {

*@Test*

public void testAdd() {

Calculator calc = new Calculator();

int result = calc.add(4, 5);

*assertEquals*(9, result);

}

}

Calculator.java

package com.cognizant.junitdemo;

public class Calculator {

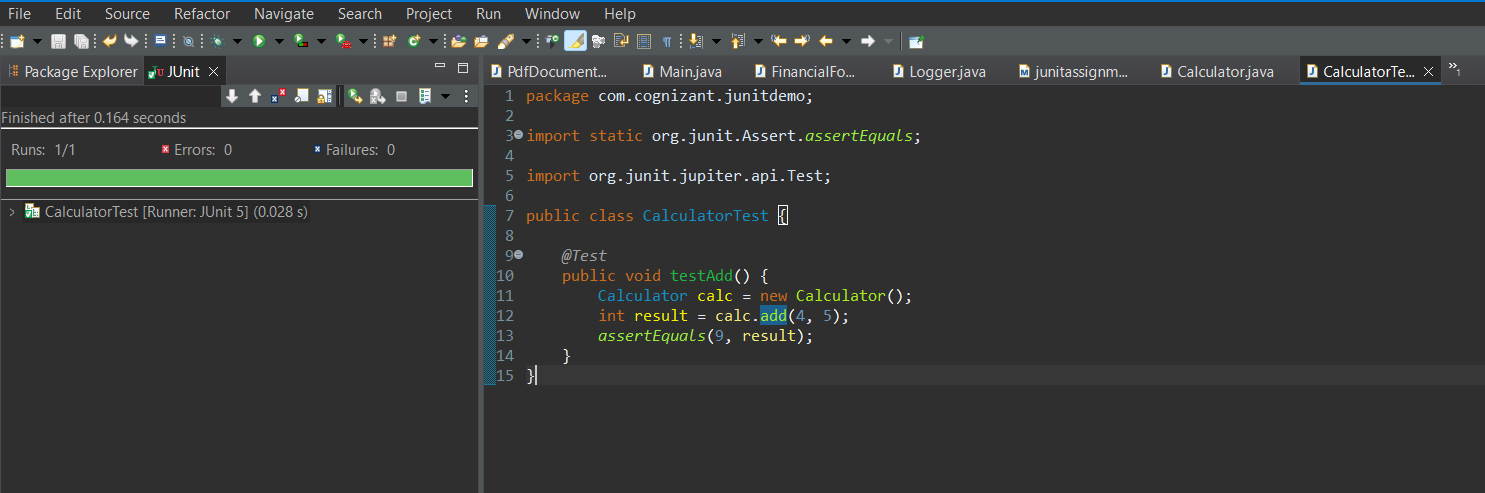
public int add(int a, int b) {

return a + b;

}

}

Output:



# Exercise 3: Assertions in JUnit

Assertions.java

package com.cognizant.junitdemo;

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);

String str = null;

*assertNull*(str);

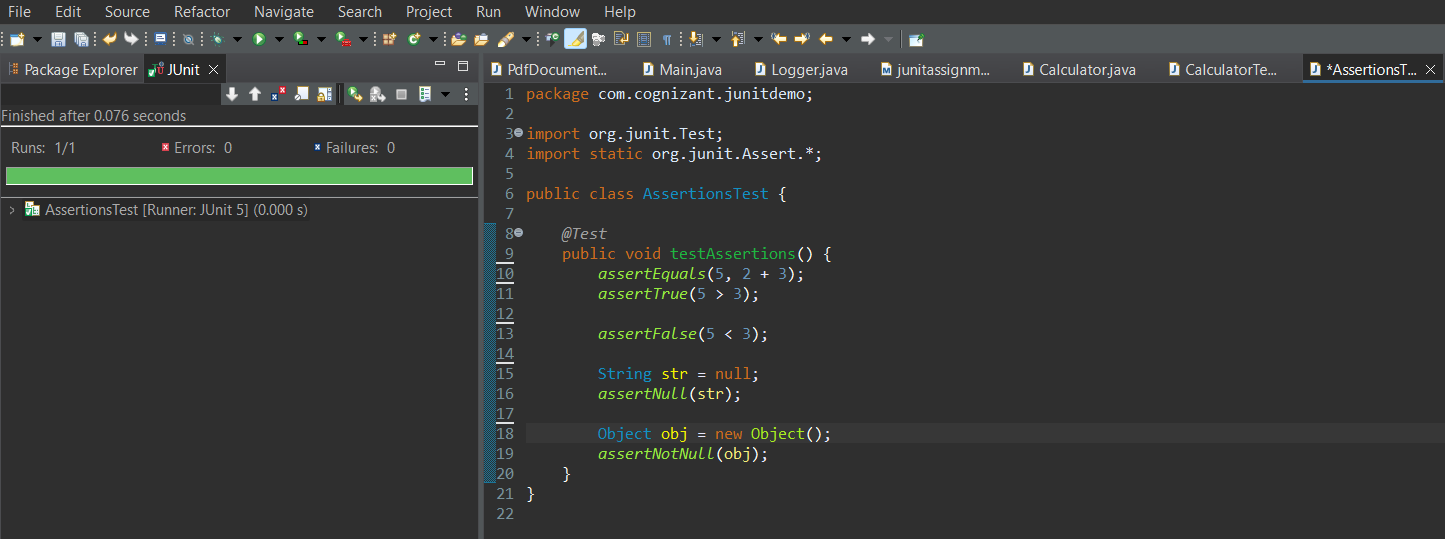
Object obj = new Object();

*assertNotNull*(obj);

}

}

Output:



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

package com.cognizant.junitdemo;

import static org.junit.Assert.*assertEquals*;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

public class CalculatorAAATest {

private Calculator calculator;

*@Before*

public void setUp() {

System.***out***.println("Setting up Calculator instance");

calculator = new Calculator(); // Arrange

}

*@After*

public void tearDown() {

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

calculator = null;

}

*@Test*

public void testAdd() {

int result = calculator.add(10, 5);

*assertEquals*(15, result);

}

*@Test*

public void testAddWithNegative() {

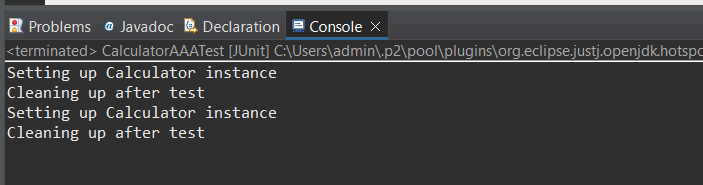
int result = calculator.add(-5, -3);

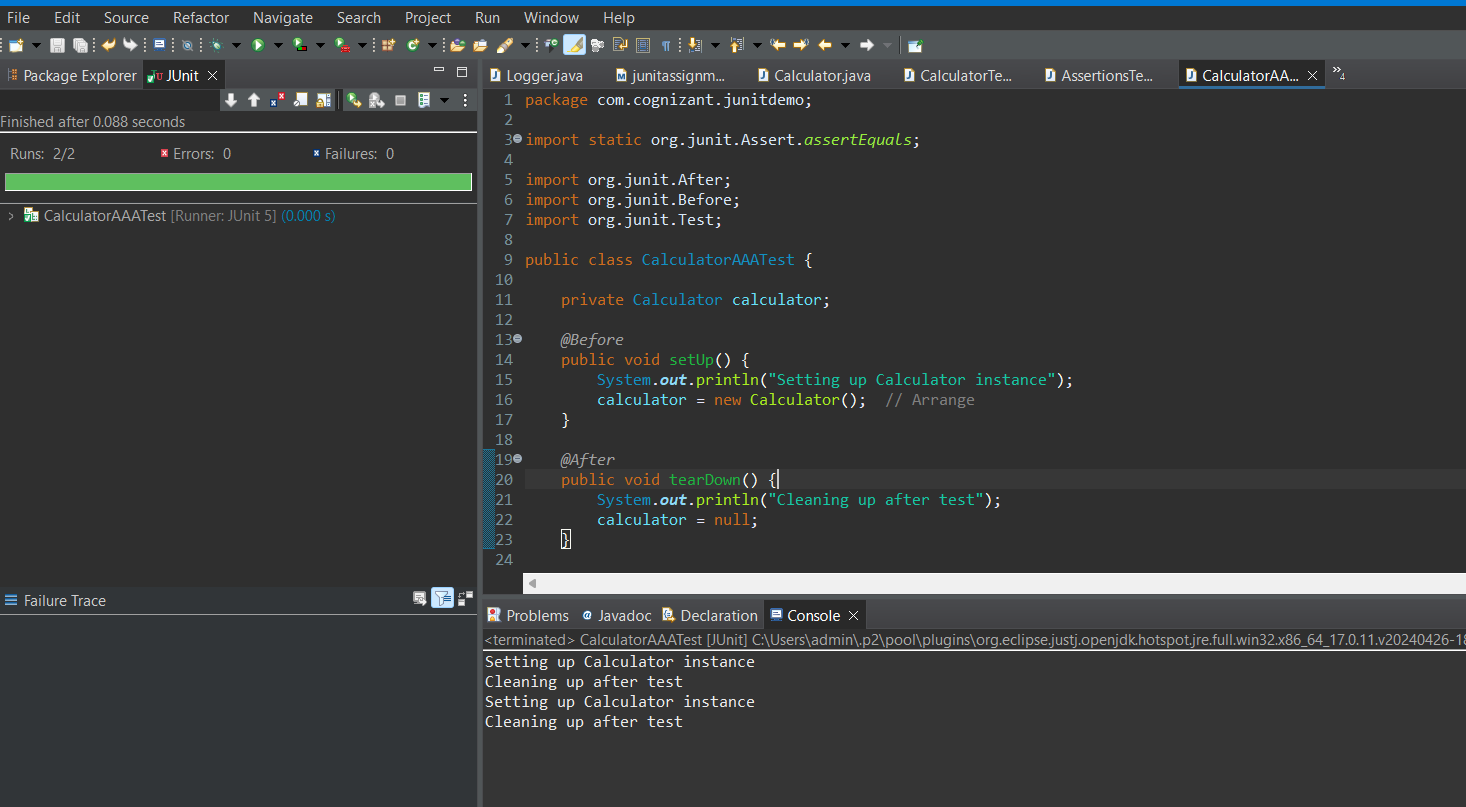
*assertEquals*(-8, result);

}

}

Output:





Mockito Hands On Exercises

# Exercise 1: Mocking and Stubbing

Adding Mockito Dependency

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>4.11.0</version>

<scope>test</scope>

</dependency>

ExternalApi.java

package com.cognizant.junitdemo;

public interface ExternalApi {

String getData();

}

MyService.java

package com.cognizant.junitdemo;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

MyServiceTest.java

package com.cognizant.junitdemo;

import org.junit.Test;

import static org.junit.Assert.*assertEquals*;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

*@Test*

public void testExternalApi() {

ExternalApi mockApi = *mock*(ExternalApi.class);

*when*(mockApi.getData()).thenReturn("Mock Data");

MyService service = new MyService(mockApi);

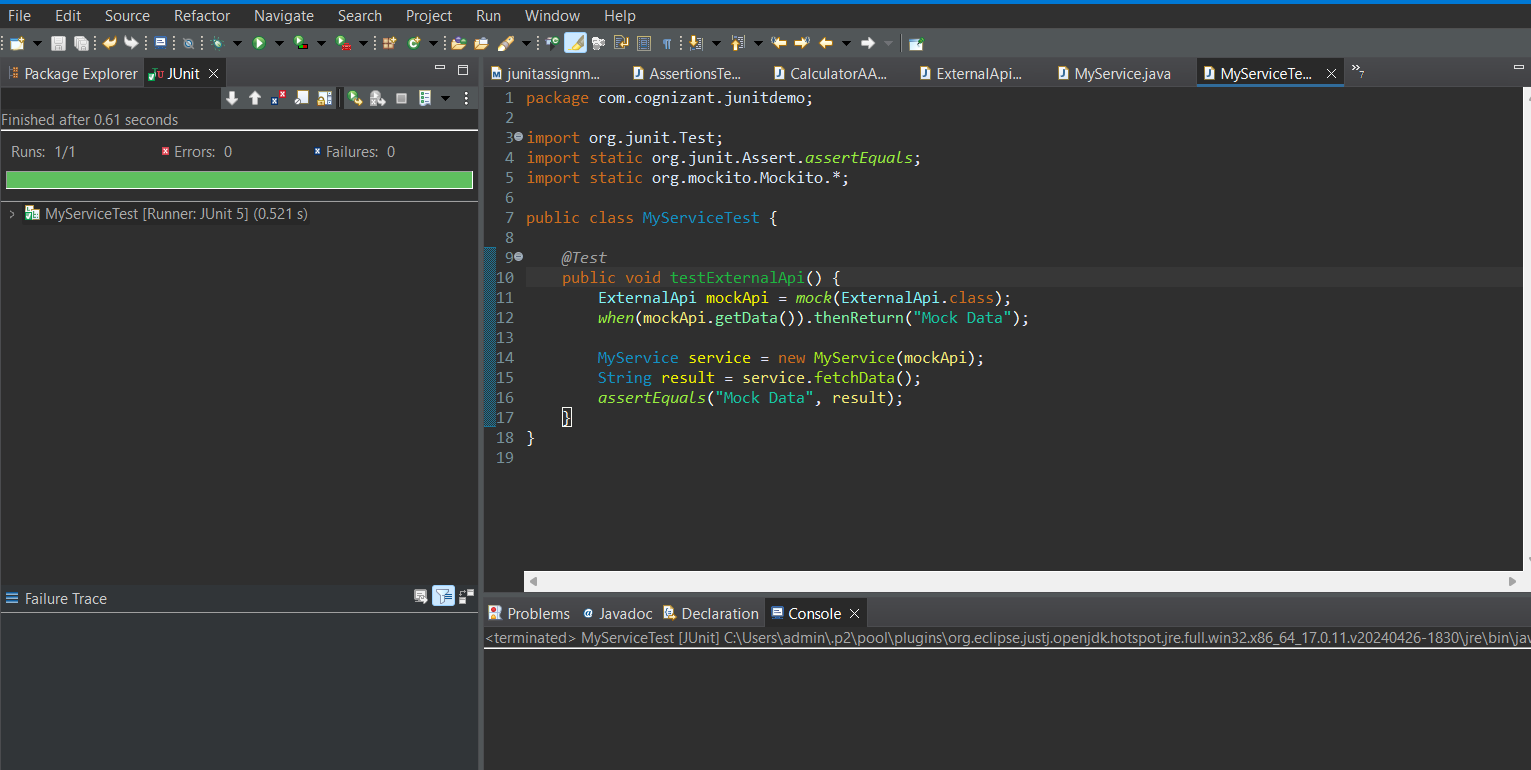
String result = service.fetchData();

*assertEquals*("Mock Data", result);

}

}

Output:



# Exercise 2: Verifying Interactions

ExternalApi.java

package com.cognizant.junitdemo;

public interface ExternalApi {

String getData();

}

MyService.java

package com.cognizant.junitdemo;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

MyServiceTest.java

package com.cognizant.junitdemo;

import org.junit.Test;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

*@Test*

public void testVerifyInteraction() {

ExternalApi mockApi = *mock*(ExternalApi.class);

MyService service = new MyService(mockApi);

service.fetchData();

*verify*(mockApi).getData();

}

}

Output:

