**WEEK 2 HANDS ON EXERCISE – TDD using JUnit5 and Mockito – Mockito Exercises**

**Exercise 1 : Mocking and Stubbing**

**PROBLEM STATEMENT:**

You need to test a service that depends on an external API. Use Mockito to mock the external API and stub its methods.

**CODE :**

**MyService.java**

package com.example.JUnitDemo;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**ExternalApi.java**

package com.example.JUnitDemo;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**Calculator.java**

package com.example.JUnitDemo;

import static org.junit.Assert.\*;

import static org.mockito.Mockito.*mock*;

import static org.mockito.Mockito.*when*;

import org.junit.Test;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

public class CalculatorTest

{

private Calculator calculator;

*@Test*

public void testExternalApi() {

// Step 1: Create a mock for the ExternalApi

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

// Step 2: Stub the method getData() to return "Mock Data"

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

// Step 3: Inject mock into service and call fetchData()

MyService service = new MyService(mockApi);

String result = service.fetchData();

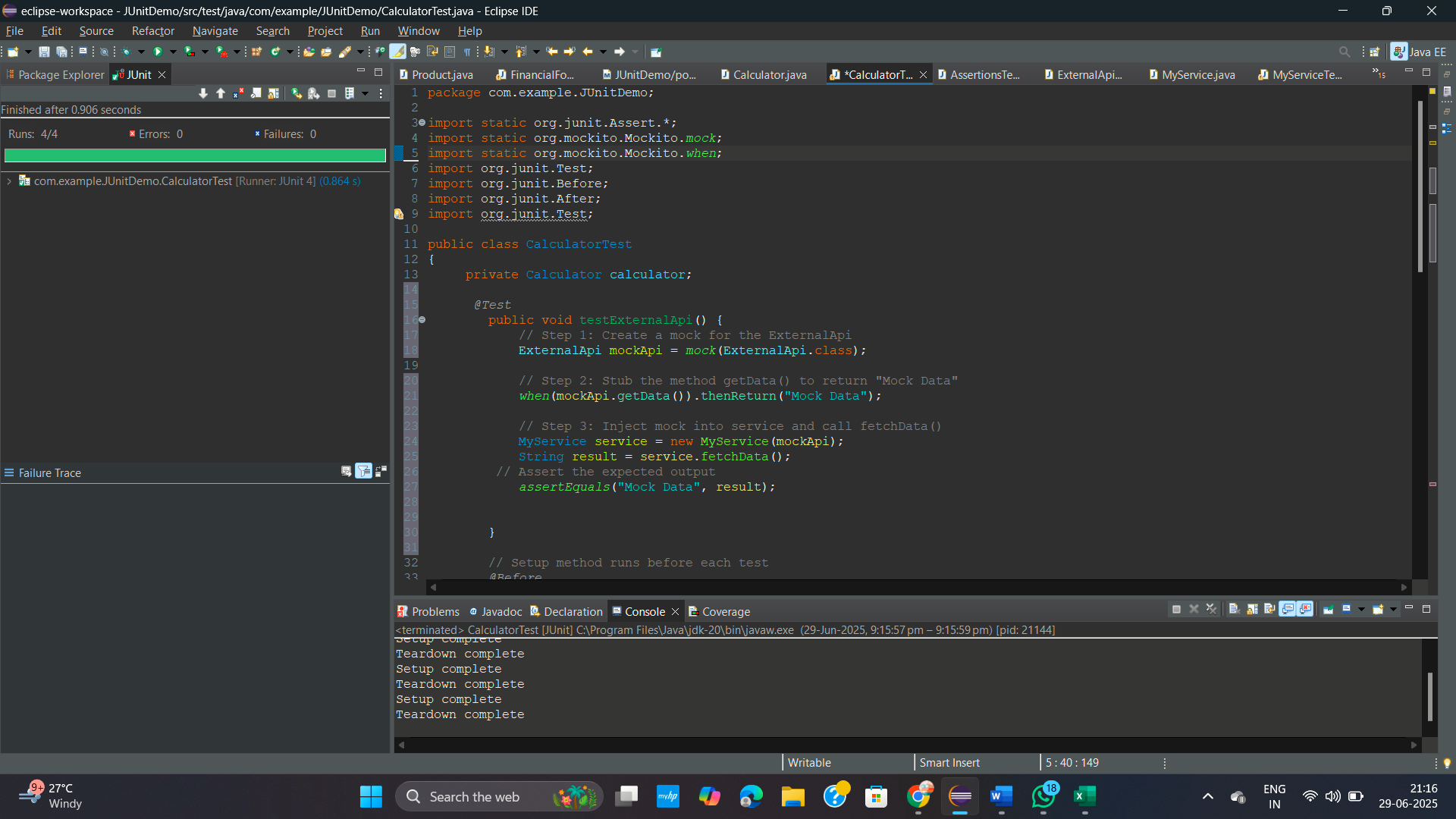
// Assert the expected output

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

}

}

**OUTPUT :**



**Exercise 2 : Verifying Interactions**

**PROBLEM STATEMENT:**

You need to ensure that a method is called with specific arguments.

**CODE :**

**MyService.java**

package com.example.JUnitDemo;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**ExternalApi.java**

package com.example.JUnitDemo;

public interface ExternalApi {

String getData();

}

**CalculatorTest.java**

package com.example.JUnitDemo;

import static org.junit.Assert.\*;

import static org.mockito.Mockito.*mock*;

import static org.mockito.Mockito.*when*;

import org.junit.Test;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.mockito.Mockito.*verify*;

public class CalculatorTest

{

private Calculator calculator;

*@Test*

public void testVerifyInteraction() {

// Step 1: Create mock

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

// Step 2: Inject mock into service

MyService service = new MyService(mockApi);

service.fetchData(); // This should call mockApi.getData()

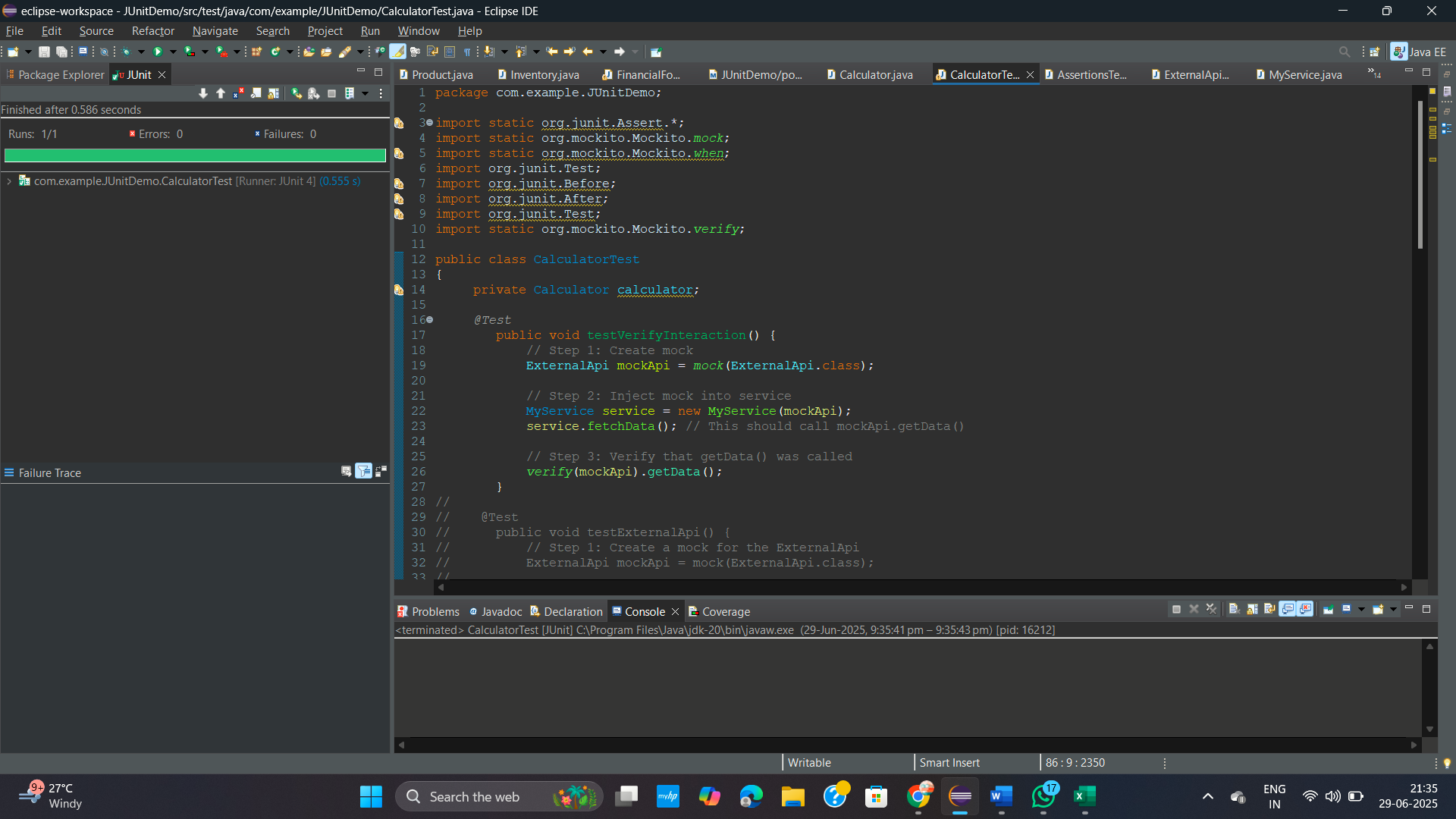
// Step 3: Verify that getData() was called

*verify*(mockApi).getData();

}

}

**OUTPUT :**



**DEPENDENCIES :**

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>JUnitDemo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>JUnitDemo</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<!-- JUnit 5 -->

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter</artifactId>

<version>5.9.3</version>

<scope>test</scope>

</dependency>

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>4.11.0</version>

<scope>test</scope>

</dependency>

<!-- SLF4J API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<!-- Logback implementation for SLF4J -->

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

</dependencies>

</project>