**Week-2 HANDS-ON( TDD using JUnit5 and Mockito )**

**1.JUnit Testing Exercises**

**Exercise 1: Setting Up Junit**

**pom.xml :**

<?xml version="1.0" encoding="UTF-8"?>

<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>mocking-example</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- JUnit 5 -->

<dependency>

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

<artifactId>junit-jupiter</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.12.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Ensure JUnit 5 support -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

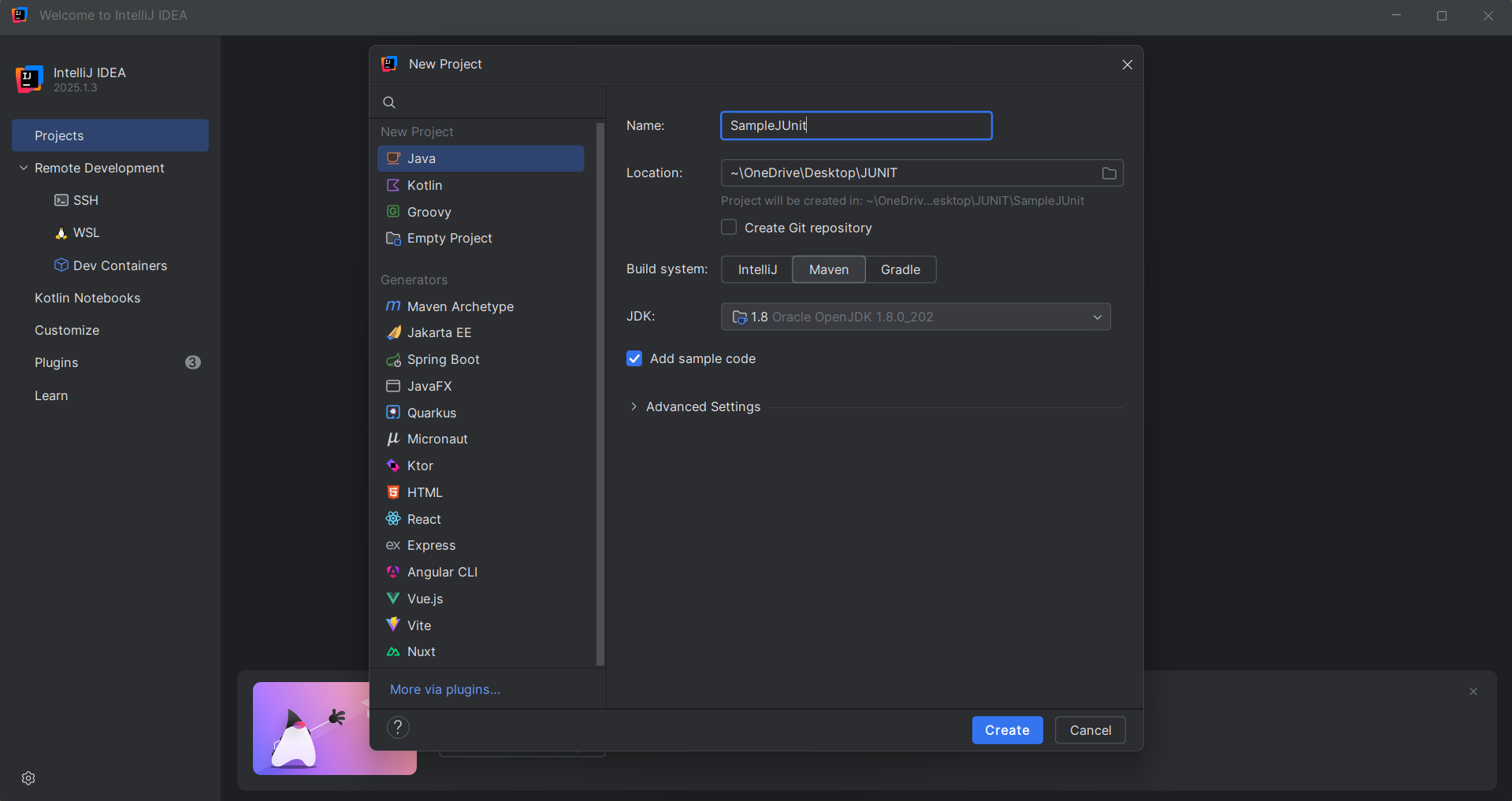
<version>3.2.5</version>

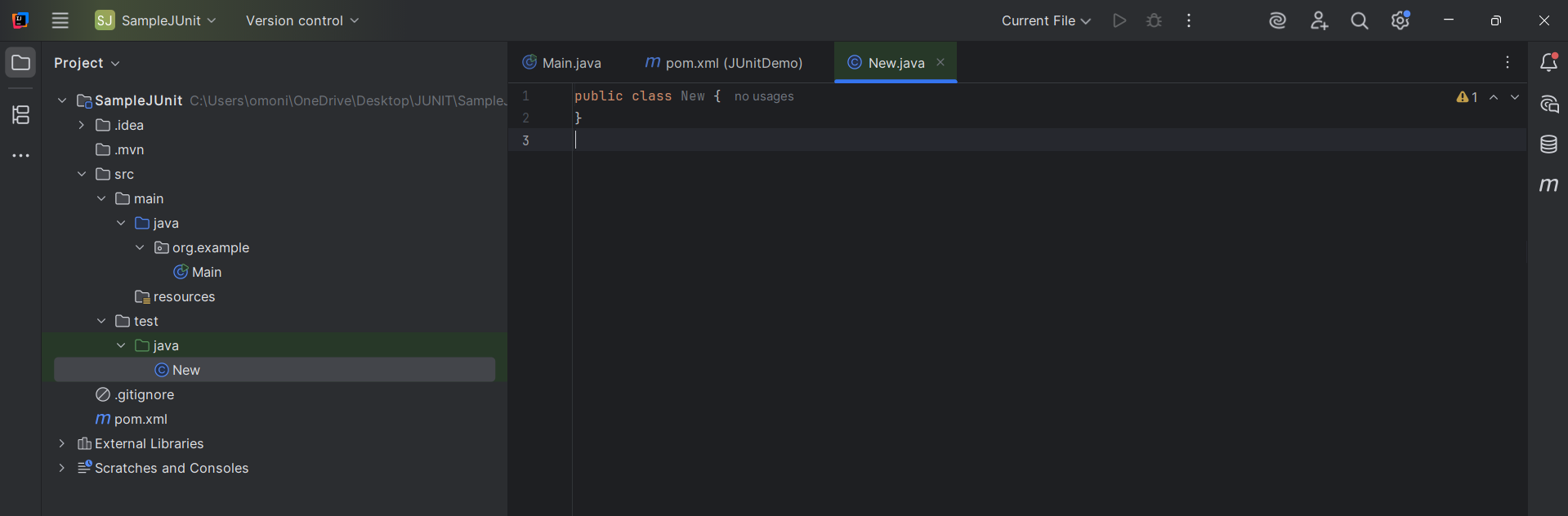
</plugin>

</plugins>

</build>

</project>





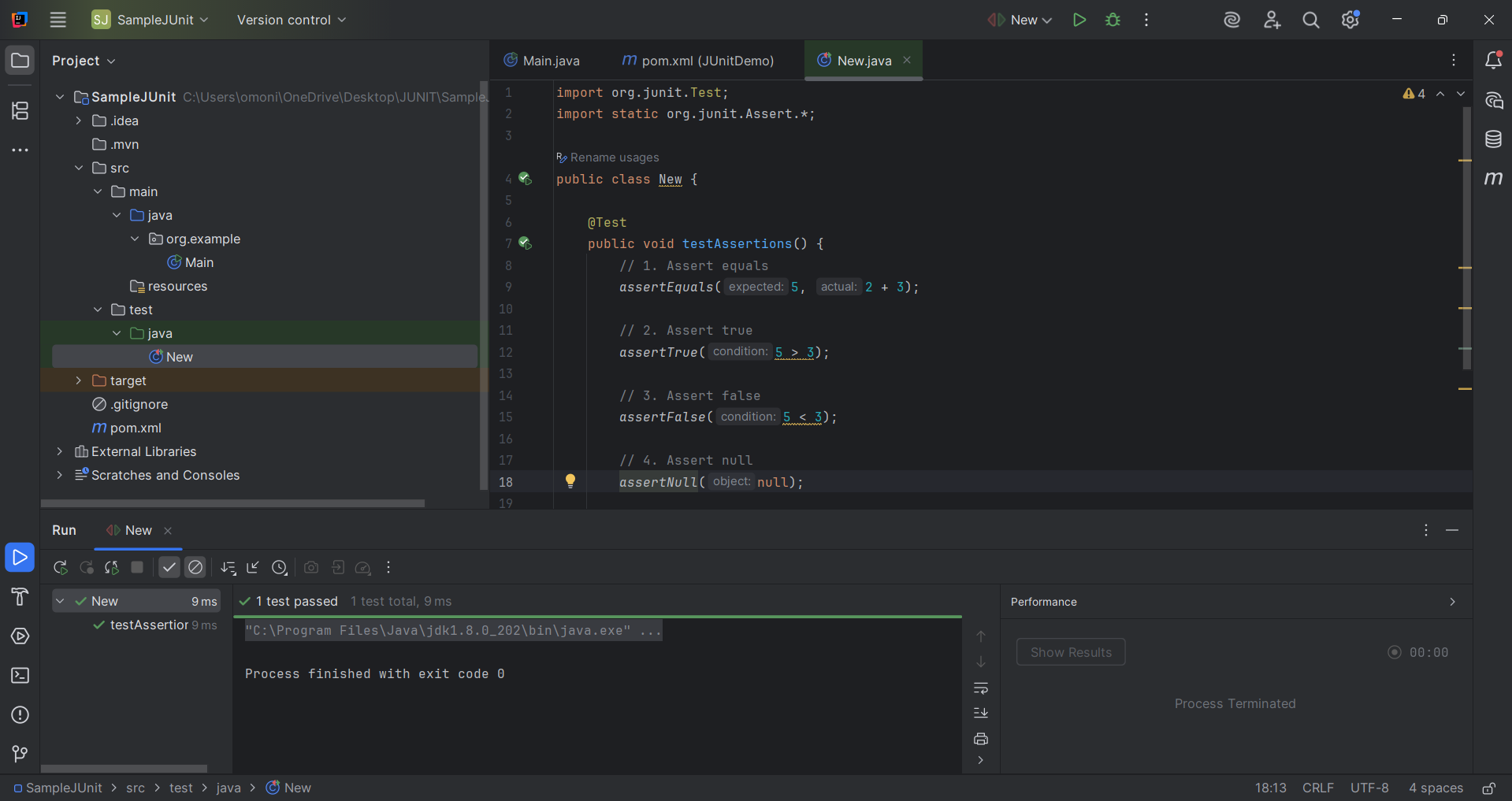
**Exercise 3: Assertions in Junit**

**New.java:**

**CODE :**

import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class New {  
  
 @Test  
 public void testAssertions() {  
 // 1. Assert equals  
 *assertEquals*(5, 2 + 3);  
  
 // 2. Assert true  
 *assertTrue*(5 > 3);  
  
 // 3. Assert false  
 *assertFalse*(5 < 3);  
  
 // 4. Assert null  
 *assertNull*(null);  
  
 // 5. Assert not null  
 *assertNotNull*(new Object());  
 }  
}

**OUTPUT :**



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

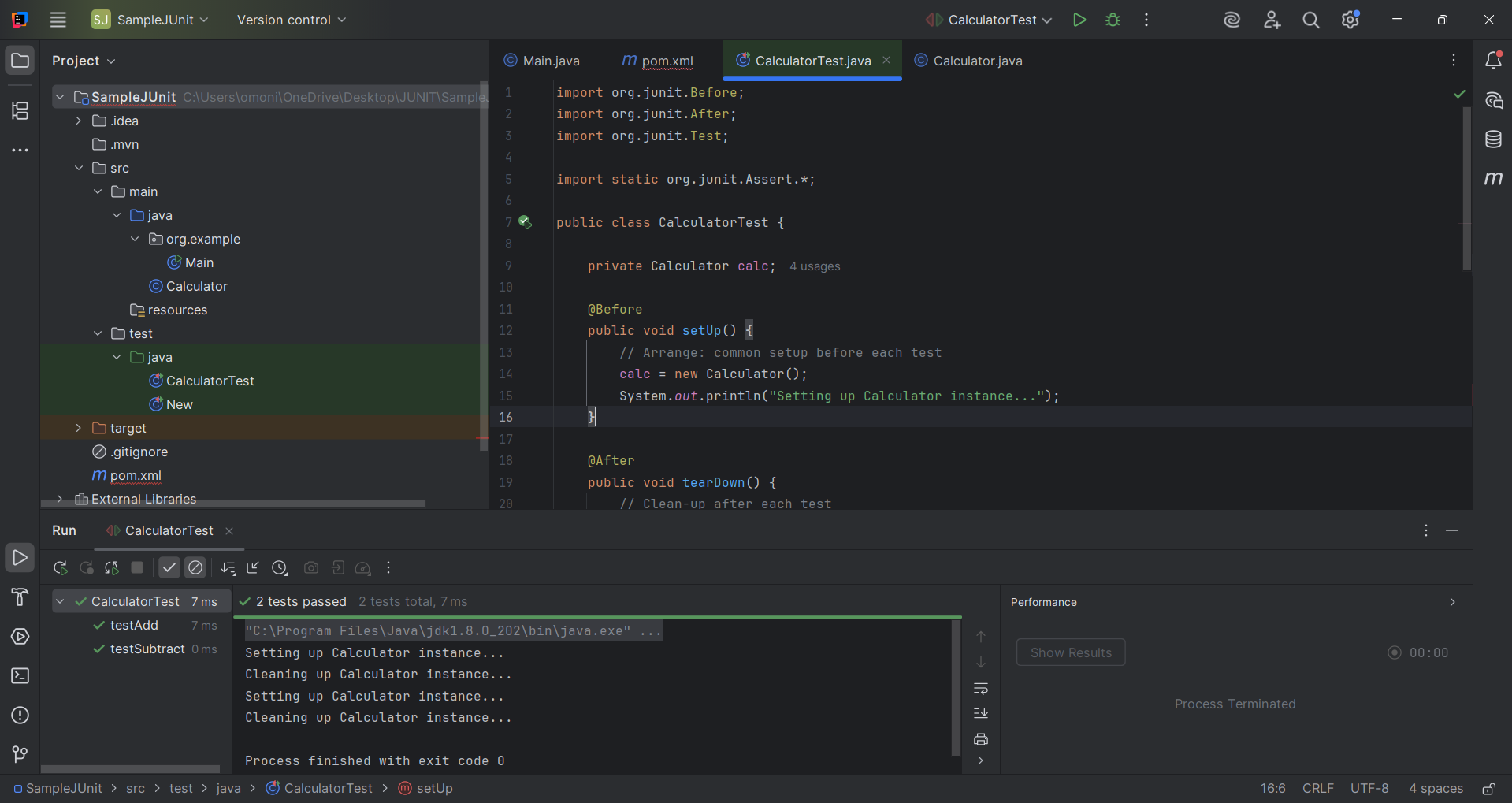
**Calculator.java :**

public class Calculator {  
 public int add(int a, int b) {  
 return a + b;  
 }  
 public int subtract(int a, int b) {  
 return a - b;  
 }  
}

**CaculatorTest.java :**

import org.junit.Before;  
import org.junit.After;  
import org.junit.Test;  
  
import static org.junit.Assert.\*;  
  
public class CalculatorTest {  
  
 private Calculator calc;  
  
 @Before  
 public void setUp() {  
 // Arrange: common setup before each test  
 calc = new Calculator();  
 System.*out*.println("Setting up Calculator instance...");  
 }  
  
 @After  
 public void tearDown() {  
 // Clean-up after each test  
 calc = null;  
 System.*out*.println("Cleaning up Calculator instance...");  
 }  
  
 @Test  
 public void testAdd() {  
 // Act  
 int result = calc.add(2, 3);  
  
 // Assert  
 *assertEquals*(5, result);  
 }  
  
 @Test  
 public void testSubtract() {  
 // Act  
 int result = calc.subtract(5, 3);  
  
 // Assert  
 *assertEquals*(2, result);  
 }  
}

**OUTPUT :**



**2.Mockito Hands-On Exercises**

**Exercise 1: Mocking and Stubbing**

**ExternalApi.java :**

**CODE :**

import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
  
import static org.junit.jupiter.api.Assertions.\*;  
import static org.mockito.Mockito.\*;  
  
public class MyServiceTest {  
  
 @Test  
 public void testExternalApi() {  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
 *when*(mockApi.getData()).thenReturn("Mock Data");  
  
 MyService service = new MyService(mockApi);  
 String result = service.fetchData();  
  
 *assertEquals*("Mock Data", result);  
 }  
}

**MyService.java :**

**CODE :**

public class MyService {  
 private ExternalApi api;  
  
 public MyService(ExternalApi api) {  
 this.api = api;  
 }  
  
 public String fetchData() {  
 return api.getData();  
 }  
}

**MyServiceTest.java :**

**CODE :**

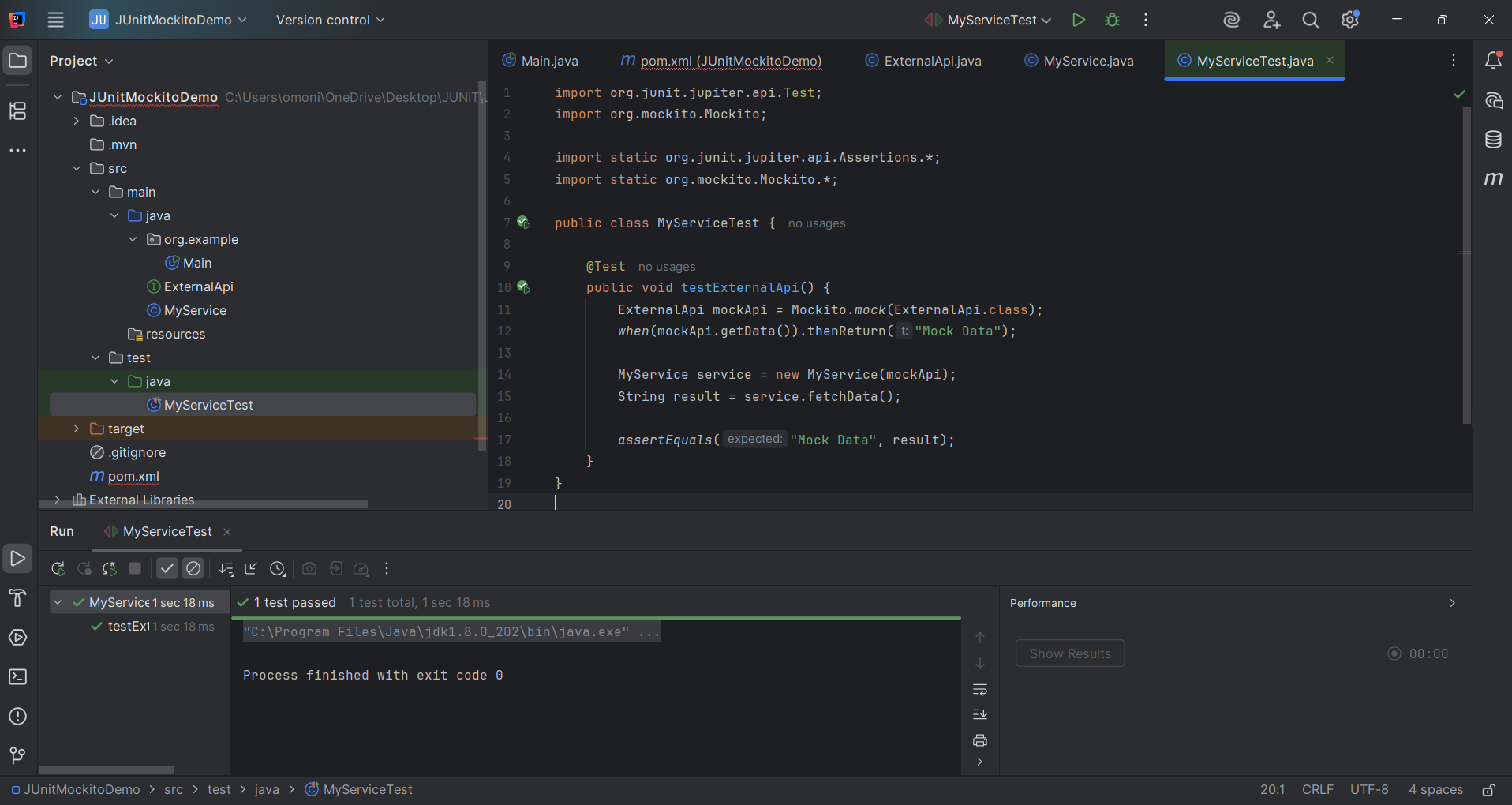
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
  
import static org.junit.jupiter.api.Assertions.\*;  
import static org.mockito.Mockito.\*;  
  
public class MyServiceTest {  
  
 @Test  
 public void testExternalApi() {  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
 *when*(mockApi.getData()).thenReturn("Mock Data");  
  
 MyService service = new MyService(mockApi);  
 String result = service.fetchData();  
  
 *assertEquals*("Mock Data", result);  
 }  
}

**pom.xml :**

**CODE :**

<?xml version="1.0" encoding="UTF-8"?>  
<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.example</groupId>  
 <artifactId>JUnitMockitoDemo</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>1.8</maven.compiler.source>  
 <maven.compiler.target>1.8</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>  
 <!-- ✅ JUnit 5 -->  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
  
 <!-- ✅ Mockito for Java 8 -->  
 <dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-core</artifactId>  
 <version>3.12.4</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <!-- ✅ Surefire plugin to run JUnit 5 tests -->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.0.0</version>  
 <configuration>  
 <useModulePath>false</useModulePath>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**OUTPUT :**



**Exercise 2: Verifying Interactions**

**ExternalApi.java :**

**CODE :**

public interface ExternalApi {  
 String getData();  
}

**MyService.java :**

**CODE :**

public class MyService {  
 private ExternalApi api; // Dependency  
  
 // Constructor injection  
 public MyService(ExternalApi api) {  
 this.api = api;  
 }  
  
 public String fetchData() {  
 return api.getData(); // Delegates to external API  
 }  
}

**MyServiceTest.java :**

**CODE :**

import static org.mockito.Mockito.\*;  
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
  
public class MyServiceTest {  
  
 @Test  
 public void testVerifyInteraction() {  
 // Step 1: Create mock object  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
  
 // Step 2: Pass it to service  
 MyService service = new MyService(mockApi);  
  
 // Step 3: Call service method  
 service.fetchData();  
  
 // Step 4: Verify that getData() was called once  
 *verify*(mockApi).getData();  
 }  
}

**pom.xml :**

**CODE :**

<?xml version="1.0" encoding="UTF-8"?>  
<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.example</groupId>  
 <artifactId>JUnitMockitoDemo</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>1.8</maven.compiler.source>  
 <maven.compiler.target>1.8</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>  
 <!-- ✅ JUnit 5 -->  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
  
 <!-- ✅ Mockito for Java 8 -->  
 <dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-core</artifactId>  
 <version>3.12.4</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <!-- ✅ Surefire plugin to run JUnit 5 tests -->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.0.0</version>  
 <configuration>  
 <useModulePath>false</useModulePath>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**OUTPUT :**

