**Mockito Exercises**

* **Exercise 1: Mocking and Stubbing**
* ***Scenario:***

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

* **To make this scenario work, the following needs to be done:**

1. **ExternalApi interface:**

Name- ExternalApi.java

Path- src/main/java/org/example/ExternalApi.java

Role- This will act as the external dependency to mock.

**Code:**

package org.example;

public interface ExternalApi {

String getData();

}

1. **MyService class:**

Name- MyService.java

Path- src/main/java/org/example/MyService.java

Role- This is the class to be tested. It uses the ExternalApi.

**Code:**

package org.example;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

1. **MyServiceTest class:**

Name- MyServiceTest.java

Path- src/test/java/org/example/MyServiceTest.java

Role- This is the final test case that uses the mock object. Its code is provided in the later steps.

1. **Add Mockito to pom.xml:**

Add this dependency-

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.11.0</version>

<scope>test</scope>

</dependency>

* **Step 1-**

Create a mock object for the external API.

**Code:**

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

* *This creates a mock object of ExternalApi, which simulates the real API.*
* **Step 2-**

Stub the methods to return predefined values.

**Code:**

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

* *This tells the mock object to return "Mock Data" when getData() is called.*
* **Step 3-**

Write a test case that uses the mock object.

**Code:**

MyService service = new MyService(mockApi); //Inject the mock

String result = service.fetchData(); //Call the method under test

assertEquals("Mock Data", result); //Verify the result

* *This is where we used the mock object, called the actual method, and checked the expected behavior using assertEquals.*
* **Full Code:**

package org.example;

import org.junit.Test;

import static org.junit.Assert.\*;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testFetchDataReturnsMockedValue() {

ExternalApi mockApi = mock(ExternalApi.class);

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

MyService service = new MyService(mockApi);

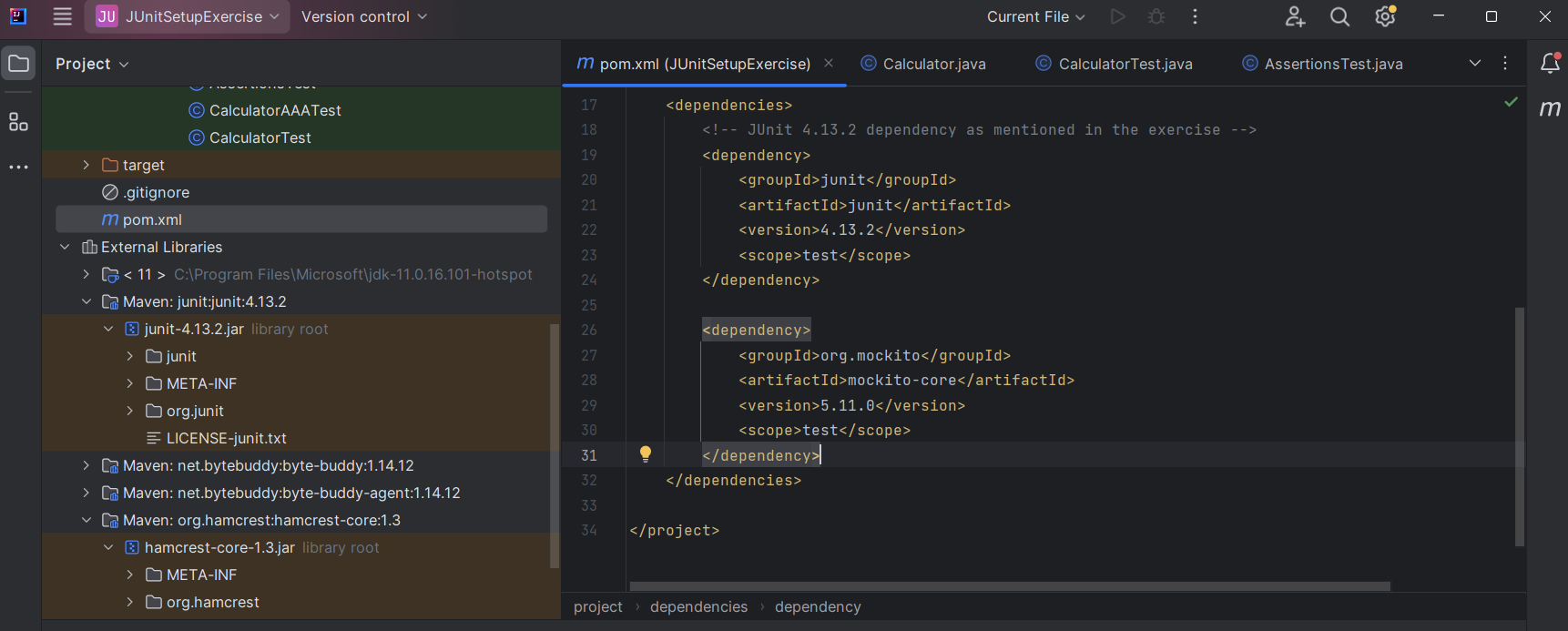
String result = service.fetchData();

assertEquals("Mock Data", result);

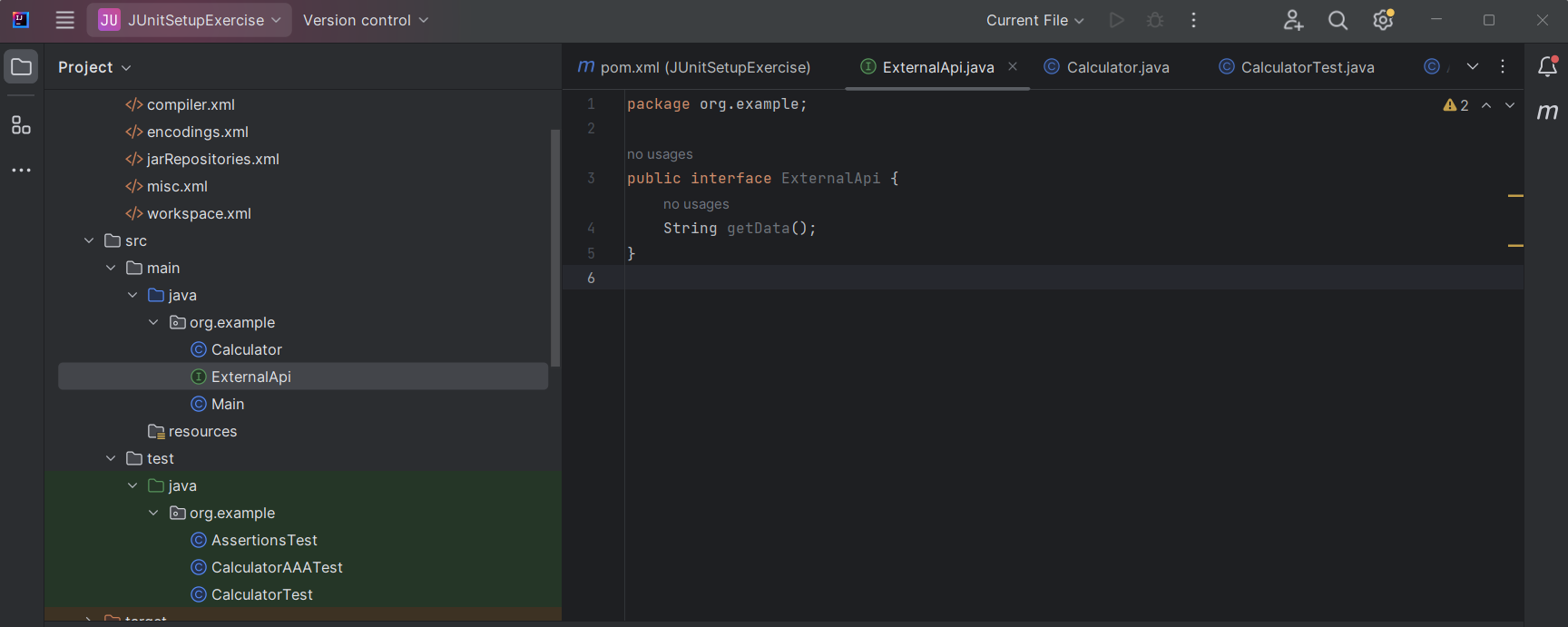
}

}

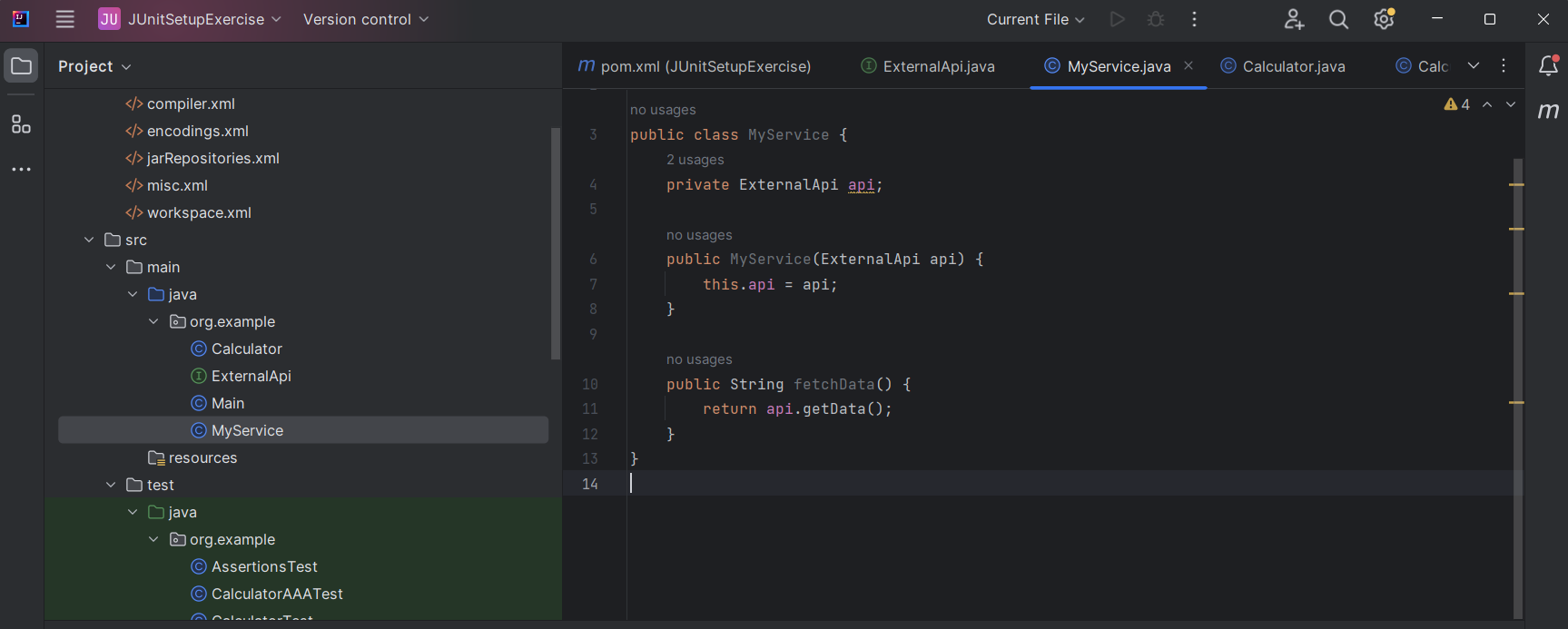
* ***Outputs:***
* **Adding the dependency**



* **ExternalApi.java file (interface)**



* **MyService.java file**



* **MyServiceTest.java file**

