**Mockito Hands-On Exercises**

**Exercise 1: Mocking and Stubbing**

**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>mockito-demo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>mockito-demo</name>

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

<properties>

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

</properties>

<dependencies>

<!-- Mockito for mocking -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

<!-- JUnit 5 for testing -->

<dependency>

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

<artifactId>junit-jupiter-api</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>junit-jupiter-engine</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

**ExternalApi.java :**

package com.example.mockito\_demo;

public interface ExternalApi {

String getData();

}

**MyService.java :**

package com.example.mockito\_demo;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData();

}

}

**MyServiceTest.java :**

package com.example.mockito\_demo;

import static org.junit.jupiter.api.Assertions.*assertEquals*;

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

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testExternalApi() {

// 1. Create a mock of ExternalApi

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

// 2. Stub the method to return "Mock Data"

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

// 3. Inject mock into MyService

MyService service = new MyService(mockApi);

// 4. Call the method under test

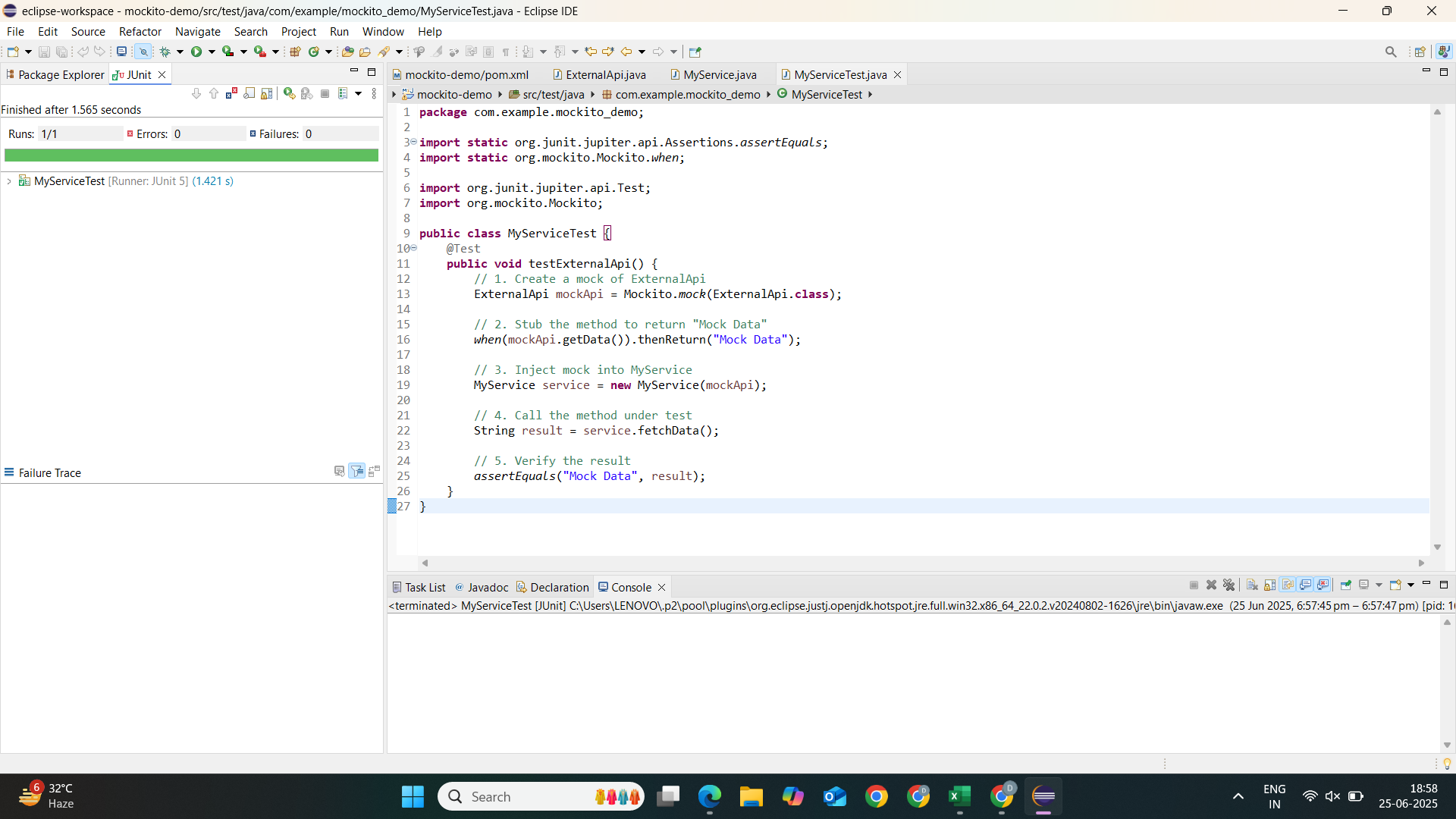
String result = service.fetchData();

// 5. Verify the result

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

}

}



**Exercise 2: Verifying Interactions**

**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>mockito-verification-demo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>mockito-verification-demo</name>

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

<properties>

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

</properties>

<dependencies>

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

<!-- JUnit 5 -->

<dependency>

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

<artifactId>junit-jupiter-api</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>junit-jupiter-engine</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

**ExternalApi.java :**

package com.example.mockito\_verification\_demo;

public interface ExternalApi {

String getData();

}

**MyService.java :**

package com.example.mockito\_verification\_demo;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData(); // Method to be verified

}

}

**MyServiceTest.java :**

package com.example.mockito\_verification\_demo;

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

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// 1. Create a mock of ExternalApi

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

// 2. Inject mock into MyService

MyService service = new MyService(mockApi);

// 3. Call the method under test

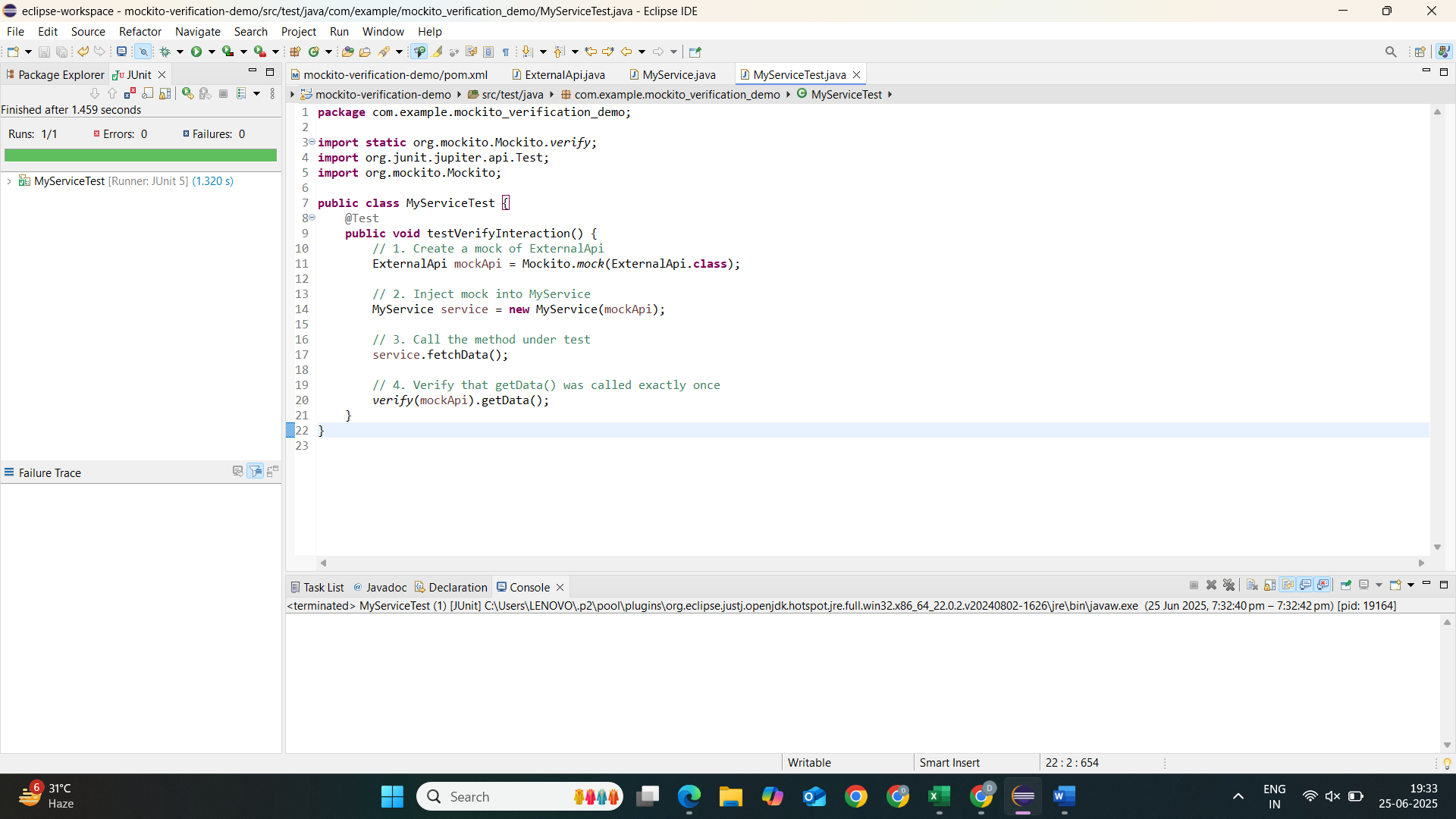
service.fetchData();

// 4. Verify that getData() was called exactly once

*verify*(mockApi).getData();

}

}

****