**Exercise 2: Verifying Interactions**

1.Main class:

public class Main {

private static Service *service*=null;

Main(Service service)

{

this.*service*=service;

}

public int performOperation(int a,int b)

{

//using mocked method

return service.add(4,5)+1;

//without using mocked method

//return *a+b+1;*

}

}

2.Helper class which acts as External API

public class Service{

public int add(int i, int j) {

return 0;

}

}

3.pom.xml file

<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>mockito</groupId>

<artifactId>MockitoOperations</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- Mockito Core (use with JUnit 5) -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.11.0</version> <!-- or latest version -->

<scope>test</scope>

</dependency>

<!-- JUnit Jupiter Engine to run tests -->

<dependency>

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

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

<version>5.12.0</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

4.unit test file for mocking and stubbing

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

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

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

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

import org.junit.jupiter.api.BeforeAll;

import org.junit.jupiter.api.Test;

class TestMain {

private static Main *m*=null;

private static Service *service*=null;

@BeforeAll

public static void beforeAll()

{

//creating a mock for service

*service*=*mock*(Service.class);

//injecting mock of service into Main

*m*=new Main(*service*);

}

@Test

void test() {

//Arrange

//stubbing: we are defining the behaviour of the mock

*when*(*service*.add(4, 5)).thenReturn(9);

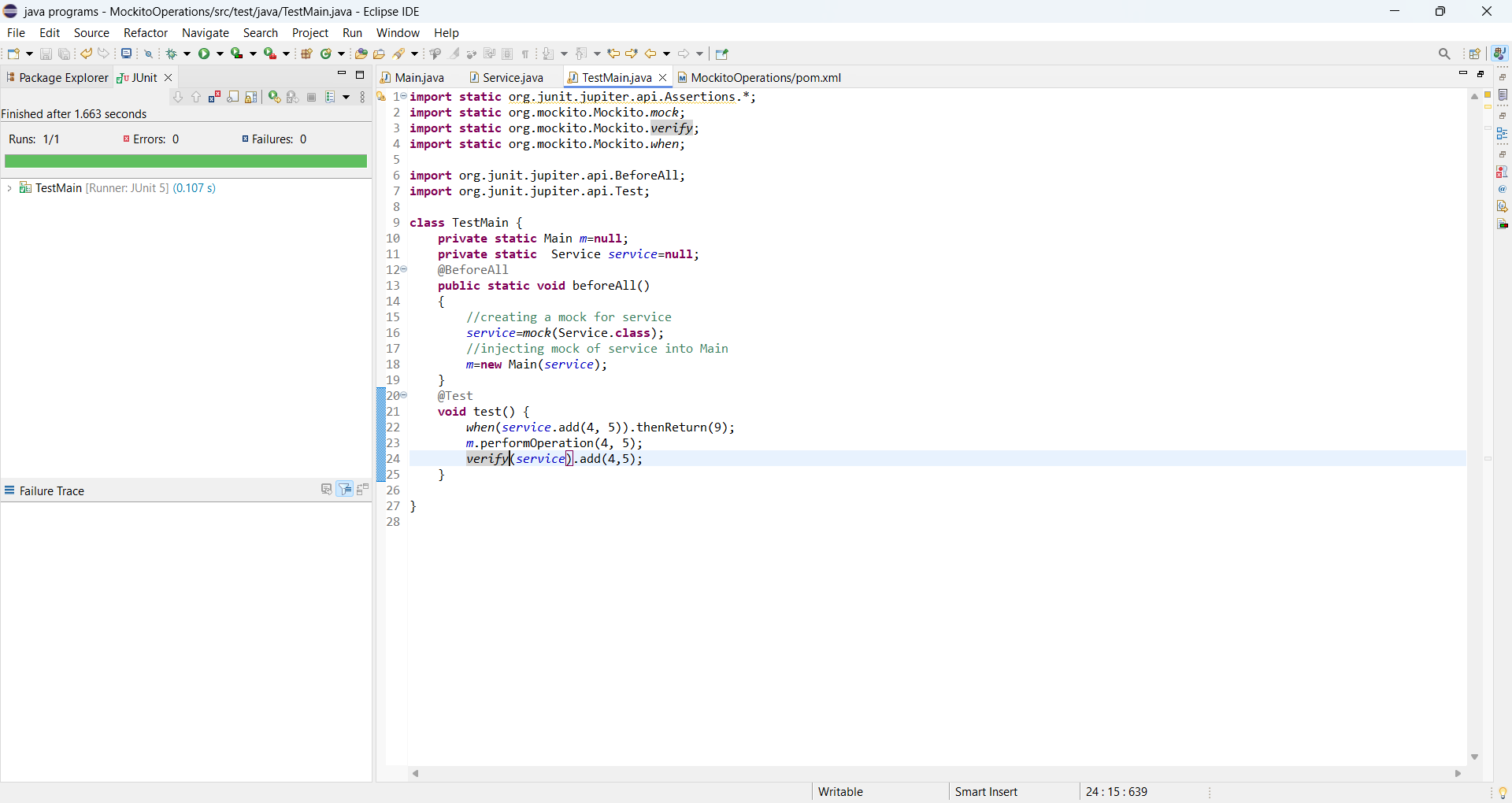
*//Act and Assert*

*assertEquals*(10,*m*.performOperation(4, 5));

}

}

Output:

1. Since the mocked behaviour is used it passes 

2.

Since the mocked behaviour is not used it fails