Ex-1 Setting up junit

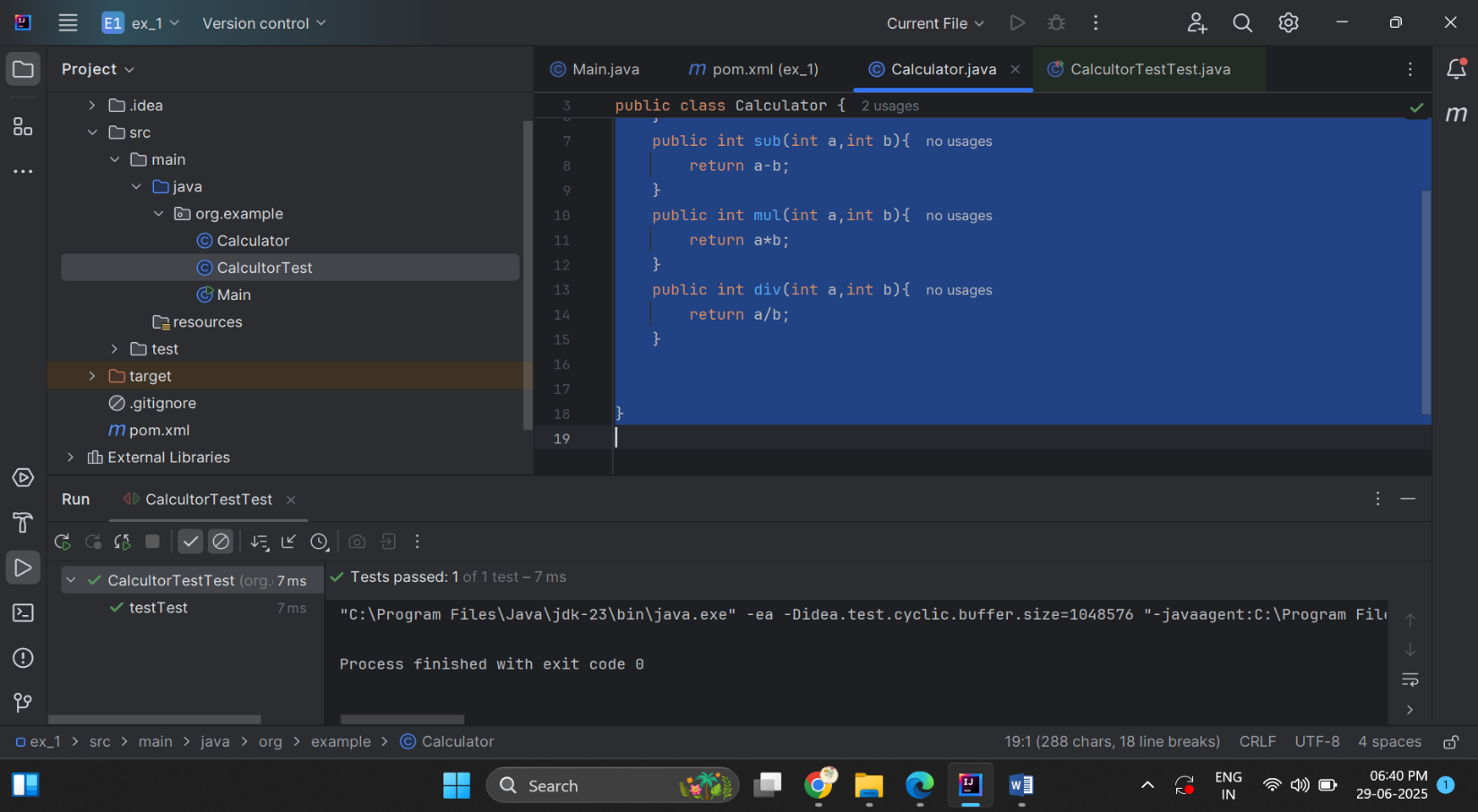
<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example</groupId>  
 <artifactId>ex\_1</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>23</maven.compiler.source>  
 <maven.compiler.target>23</maven.compiler.target>  
 <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>  
 </dependencies>  
  
  
</project>

Ex-2 Basic junit test.

package org.example;  
  
public class Calculator {  
 public int add(int a,int b){  
 return a+b;  
 }  
 public int sub(int a,int b){  
 return a-b;  
 }  
 public int mul(int a,int b){  
 return a\*b;  
 }  
 public int div(int a,int b){  
 return a/b;  
 }  
  
  
}

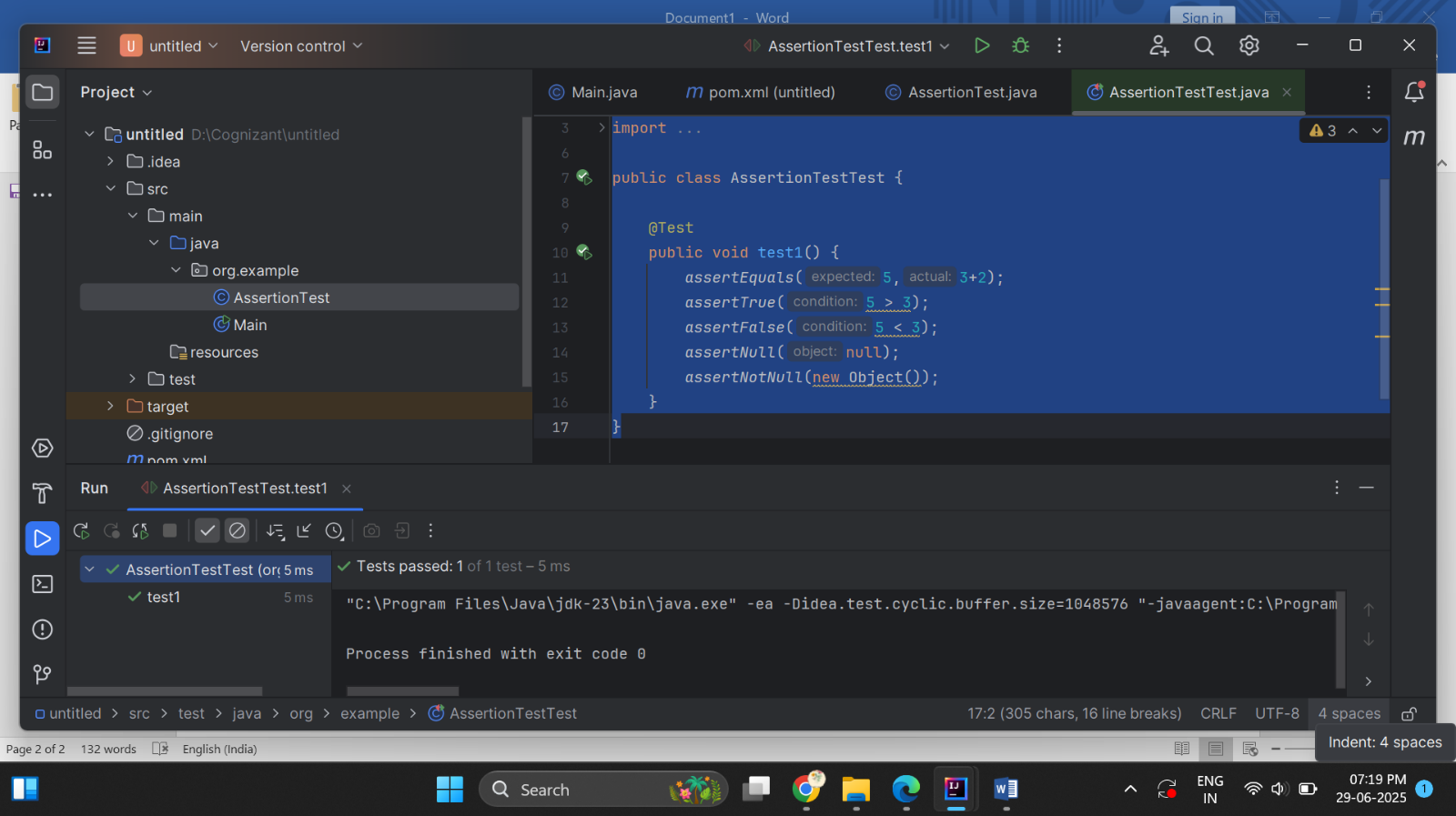
package org.example;  
  
import junit.framework.TestCase;  
import org.junit.Test;  
  
public class CalcultorTestTest extends TestCase {  
  
 @Test  
 public void testTest() {  
 Calculator calculator=new Calculator();  
 *assertEquals*(6,calculator.add(3,3));  
 *assertEquals*(0,calculator.sub(3,3));  
 *assertEquals*(100,calculator.mul(10,10));  
 *assertEquals*(3,calculator.div(9,3));  
 *assertTrue*(5>3);  
 *assertFalse*(100 < 20);  
  
  
 }  
}

output:



Ex – 3 ASSERTION TEST

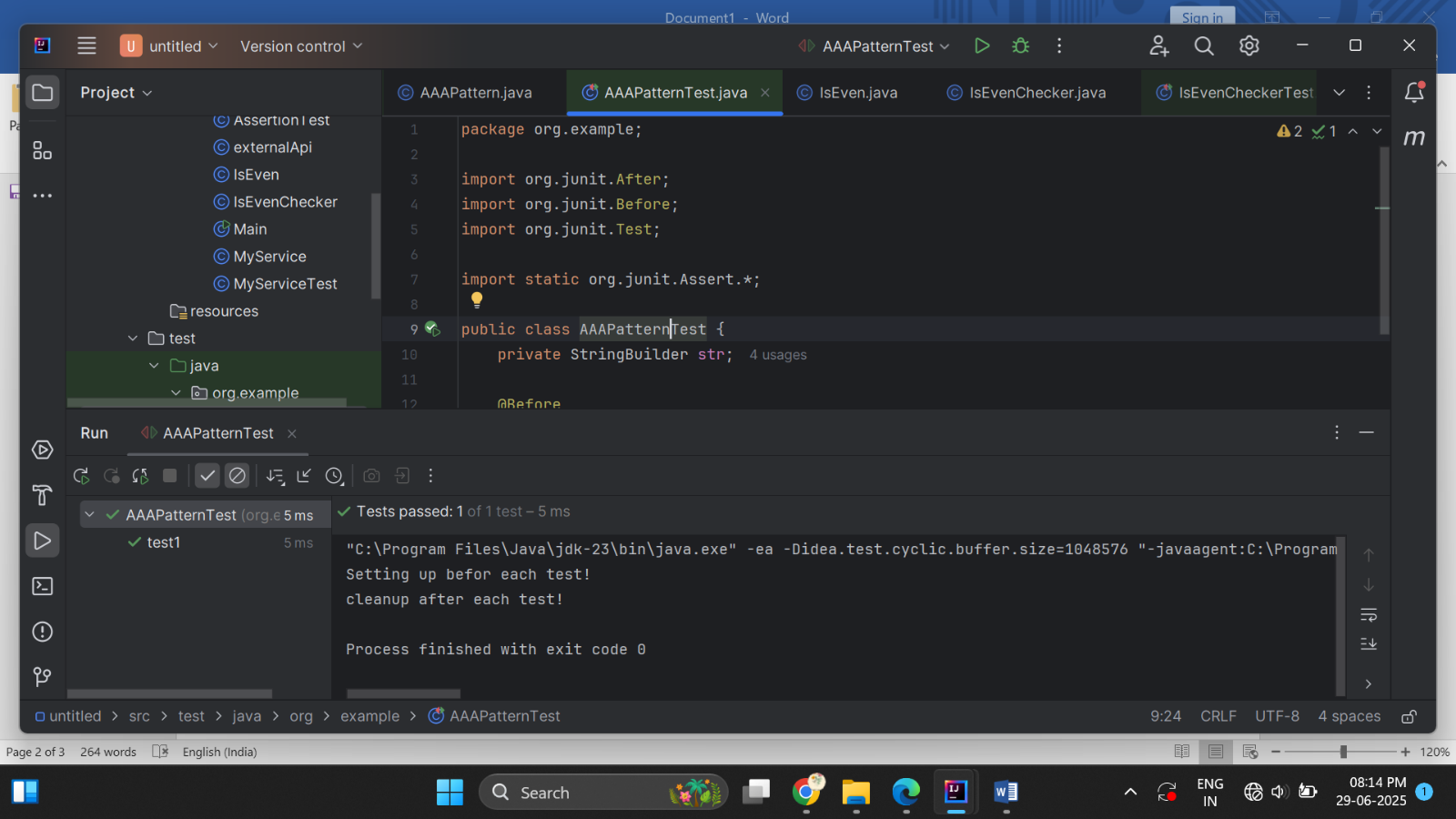
package org.example;  
  
import org.junit.Test;  
  
import static org.junit.Assert.\*;  
  
public class AssertionTestTest {  
  
 @Test  
 public void test1() {  
 *assertEquals*(5,3+2);  
 *assertTrue*(5 > 3);  
 *assertFalse*(5 < 3);  
 *assertNull*(null);  
 *assertNotNull*(new Object());  
 }  
}

output:

Ex-4 AAAPATTERN

package org.example;  
  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
  
import static org.junit.Assert.\*;  
  
public class AAAPatternTest {  
 private StringBuilder str;  
  
 @Before  
 public void setUp() throws Exception {  
 str=new StringBuilder();  
 System.*out*.println("Setting up befor each test!");  
 }  
  
 @After  
 public void tearDown() throws Exception {  
 str=null;  
 System.*out*.println("cleanup after each test!");  
 }  
  
 @Test  
 public void test1() {  
 str.append("Hello");  
 *assertEquals*(5, str.toString().length());  
 }  
}

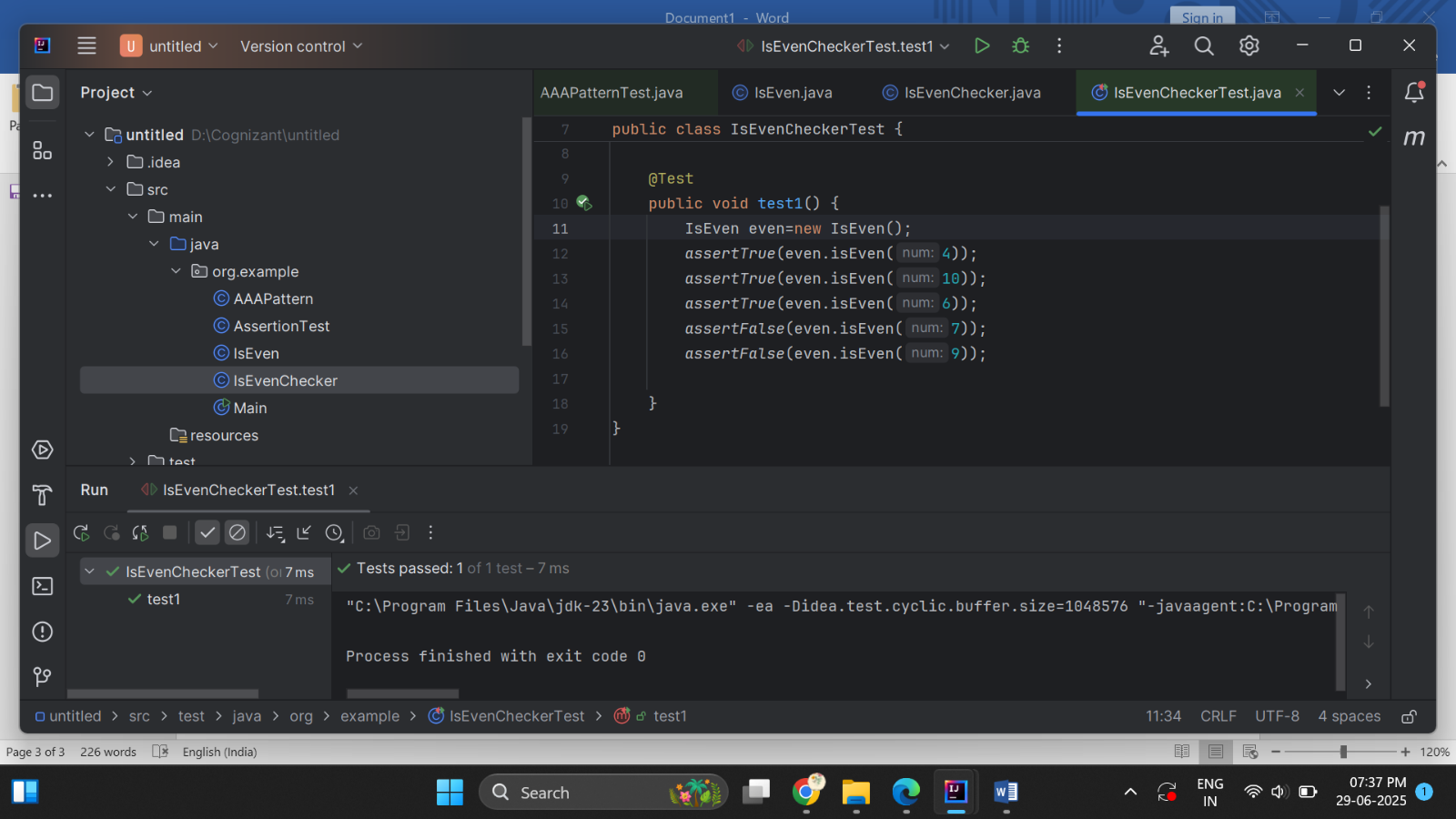
output:



**ADVANCED UNIT TESTING EXERCISE**

package org.example;  
  
import org.junit.Test;  
  
import static org.junit.Assert.\*;  
  
public class IsEvenCheckerTest {  
  
 @Test  
 public void test1() {  
 IsEven even=new IsEven();  
 *assertTrue*(even.isEven(4));  
 *assertTrue*(even.isEven(10));  
 *assertTrue*(even.isEven(6));  
 *assertFalse*(even.isEven(7));  
 *assertFalse*(even.isEven(9));  
  
 }  
}

**OUPUT:**



MOCKITO EXERCISES:

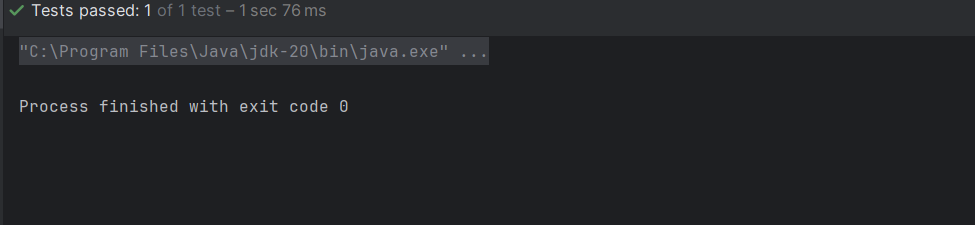
EX 1 MOCKING AND STUBBING

package org.example;  
  
public class externalApi {  
 public String getData(){  
 return "Api Found!!!" ;  
 }  
}

package org.example;  
  
public class MyService {  
 externalApi api;  
 public MyService( externalApi api){  
 this.api=api;  
 }  
 public String fetchData(){  
 return api.getData();  
 }  
}

package org.example;  
  
import junit.framework.TestCase;  
import org.junit.Test;  
import org.mockito.Mockito;  
  
import static org.junit.Assert.*assertEquals*;  
import static org.mockito.Mockito.*when*;  
  
public class MyServiceTestTest {  
 @Test  
 public void TestExternalApi(){  
 externalApi mockapi= Mockito.*mock*(externalApi.class);  
 *when*(mockapi.getData()).thenReturn("Mock Data");  
 MyService service= new MyService(mockapi);  
 String s=service.fetchData();  
 *assertEquals*("Mock Data",s);  
 }  
  
}

**OUPUT:**



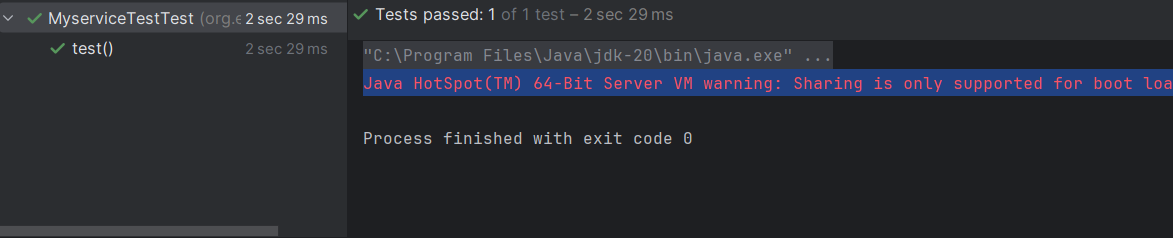
EX 2 VERIFYING INTERACTIONS

package org.example;  
  
public class externalApi {  
 public void getData() {  
  
 }  
}

package org.example;  
  
public class MyService {  
 public externalApi api;  
  
 public MyService(externalApi api){  
 this.api=api;  
 }  
  
 public void fetchData(){  
 api.getData();  
 }  
}

package org.example;  
  
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
  
import static org.junit.jupiter.api.Assertions.\*;  
import static org.mockito.Mockito.*verify*;  
  
class MyserviceTestTest {  
  
 @Test  
 void test() {  
 externalApi mockapi= Mockito.*mock*(externalApi.class);  
  
 MyService service= new MyService(mockapi);  
  
 service.fetchData();  
  
 *verify*(mockapi).getData();  
 }  
}

**OUTPUT**

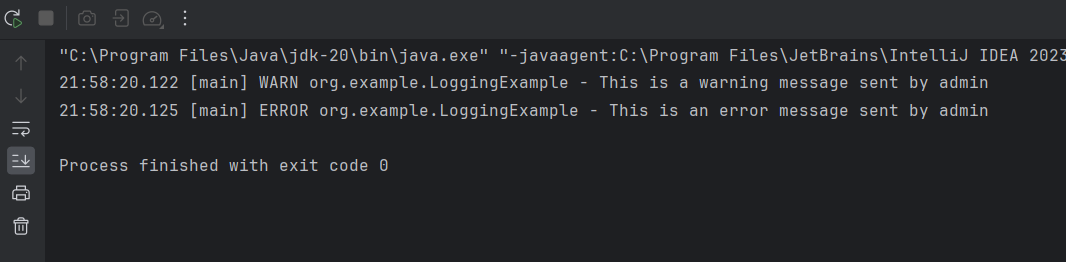
****

**SL4J LOGGING EXERCISES**

EX 1 LOGGING ERROR MESSAGES AND WARNING LEVELS

package org.example;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
  
public class LoggingExample {  
 public static final Logger *logger*=LoggerFactory.*getLogger*(LoggingExample.class);  
 public static void main(String[] args) {  
 *logger*.warn("This is a warning message sent by admin");  
 *logger*.error("This is an error message sent by admin");  
  
 }  
}

**OUTPUT**

****