***Exercise 1: Setting Up Junit***

Scenario: You need to set up JUnit in your Java project to start writing unit tests.

**Tools Used**

- IDE: Eclipse

- Build Tool: Maven

- Testing Framework: JUnit 4.13.2

- Language: Java

Add JUnit Dependency in pom.xml

<dependencies>

<dependency>

<groupId>junit</groupId>

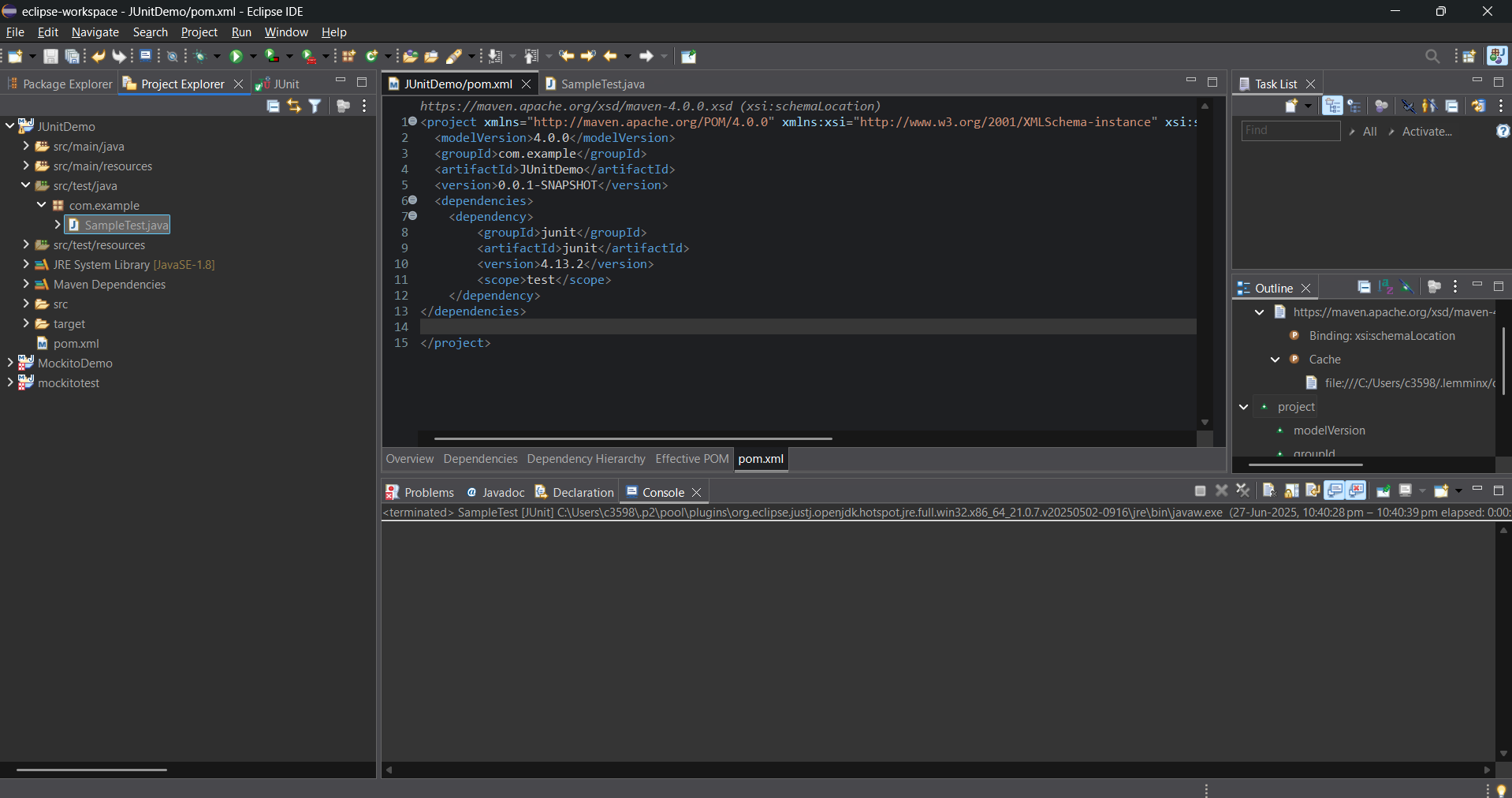
<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>



***New Test Class***

package com.example;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class SampleTest {

@Test

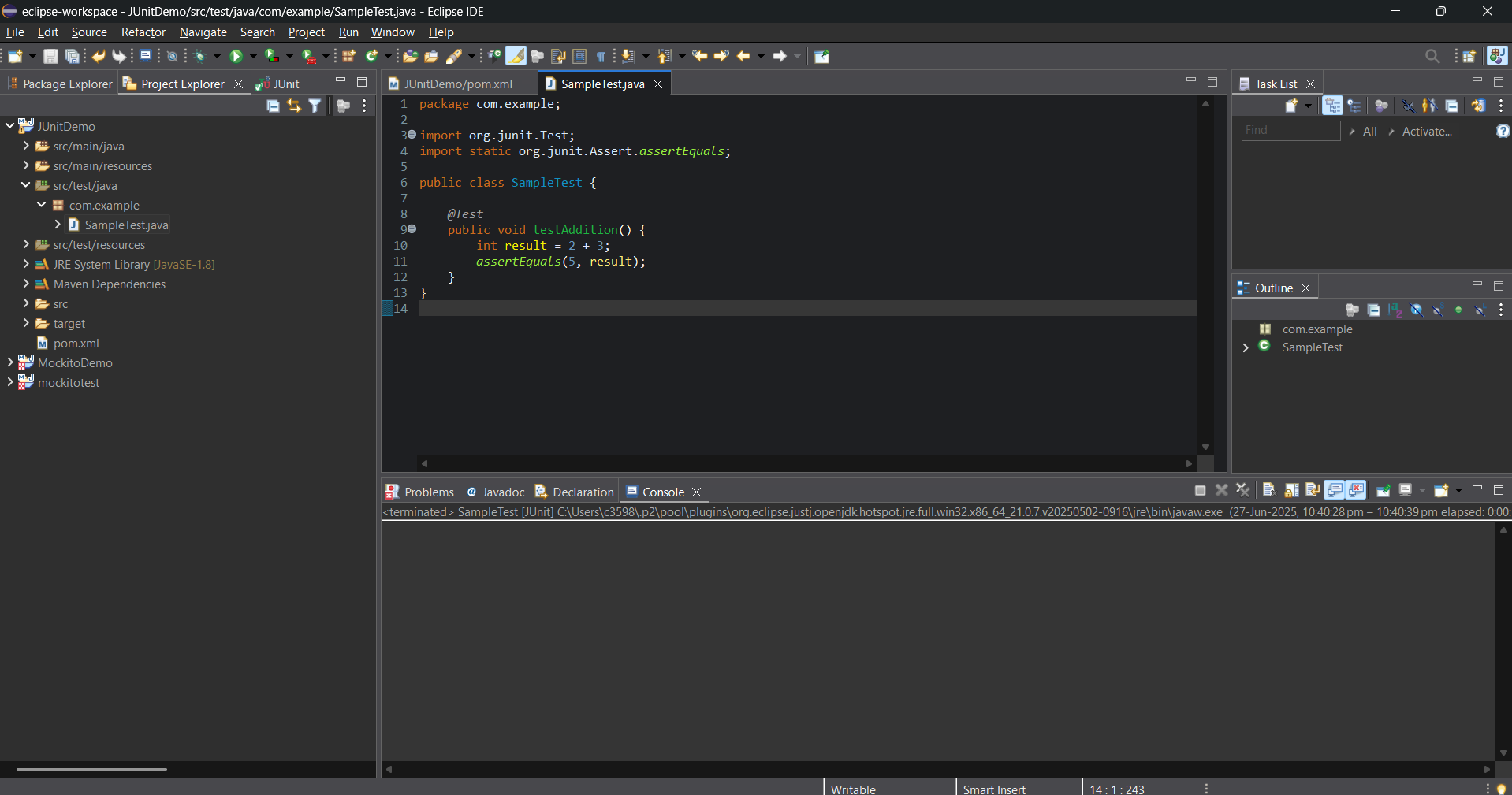
public void testAddition() {

int result = 2 + 3;

assertEquals(5, result);

}

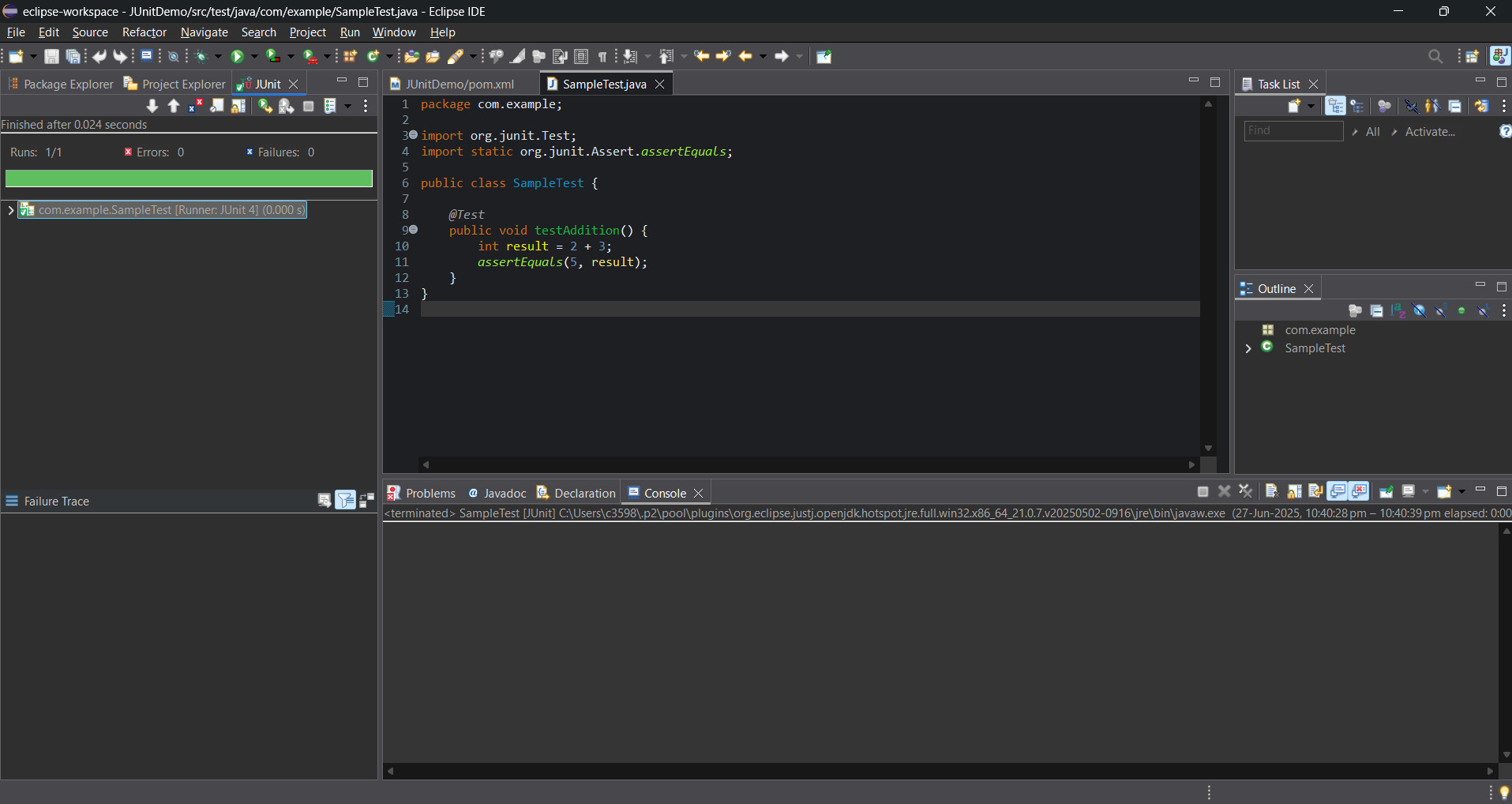
}



***OUTPUT***

The JUnit test ran successfully and the green bar confirms that the test passed.

Screenshot is pasted on the next page



***Exercise 3: Assertions in Junit***

To use various JUnit assertions to validate test results in a Java Maven project using Eclipse and JUnit

**Tools Used**

* **IDE**: Eclipse
* **Build Tool**: Maven
* **Testing Library**: JUnit 4.13.2
* **Language**: Java

Writing Test Code Using Assertions

package com.example;

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

@Test

public void testAssertions() {

// Assert equals

assertEquals(5, 2 + 3);

// Assert true

assertTrue(5 > 3);

// Assert false

assertFalse(5 < 3);

// Assert null

assertNull(null);

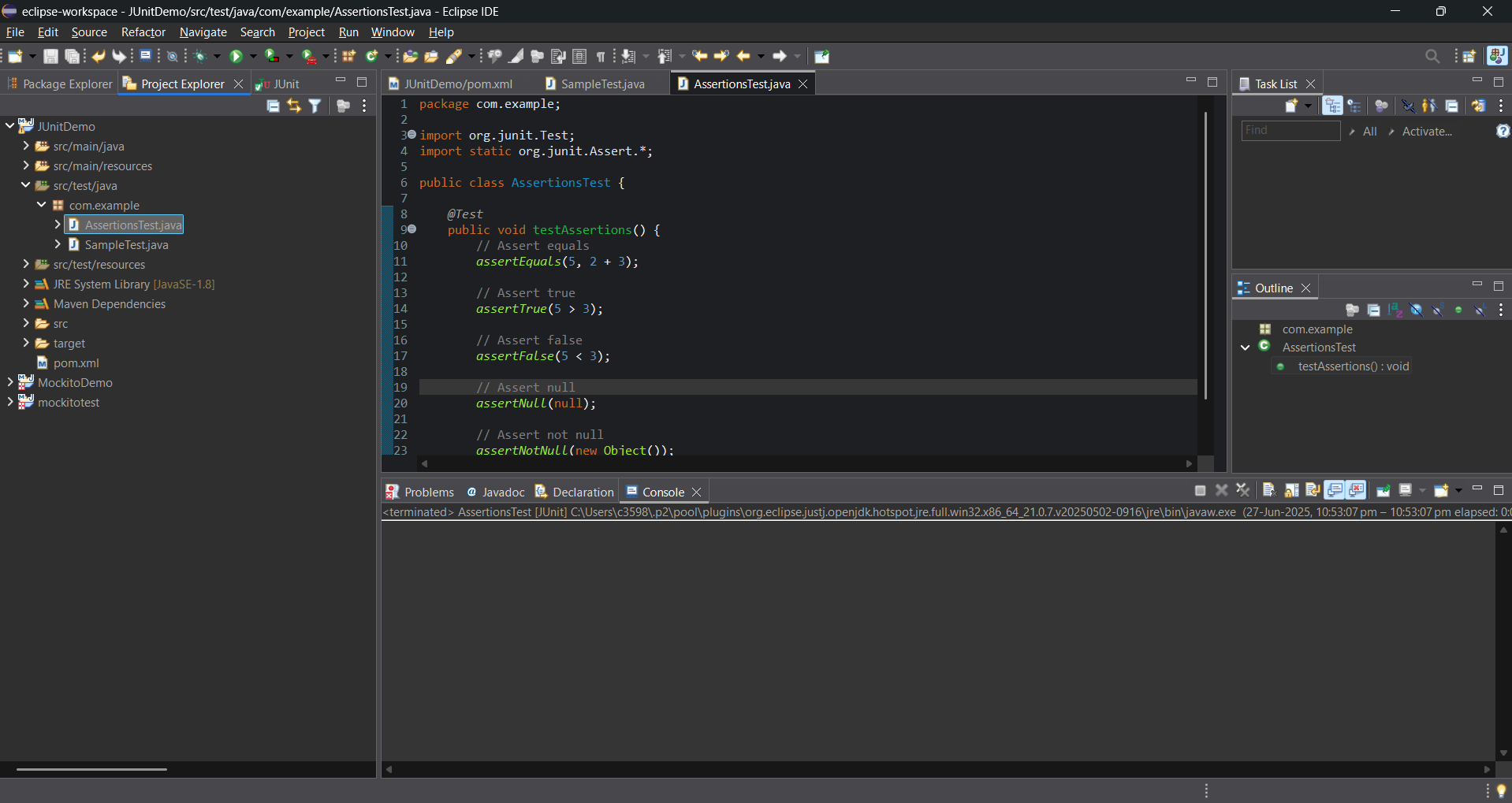
// Assert not null

assertNotNull(new Object());

}

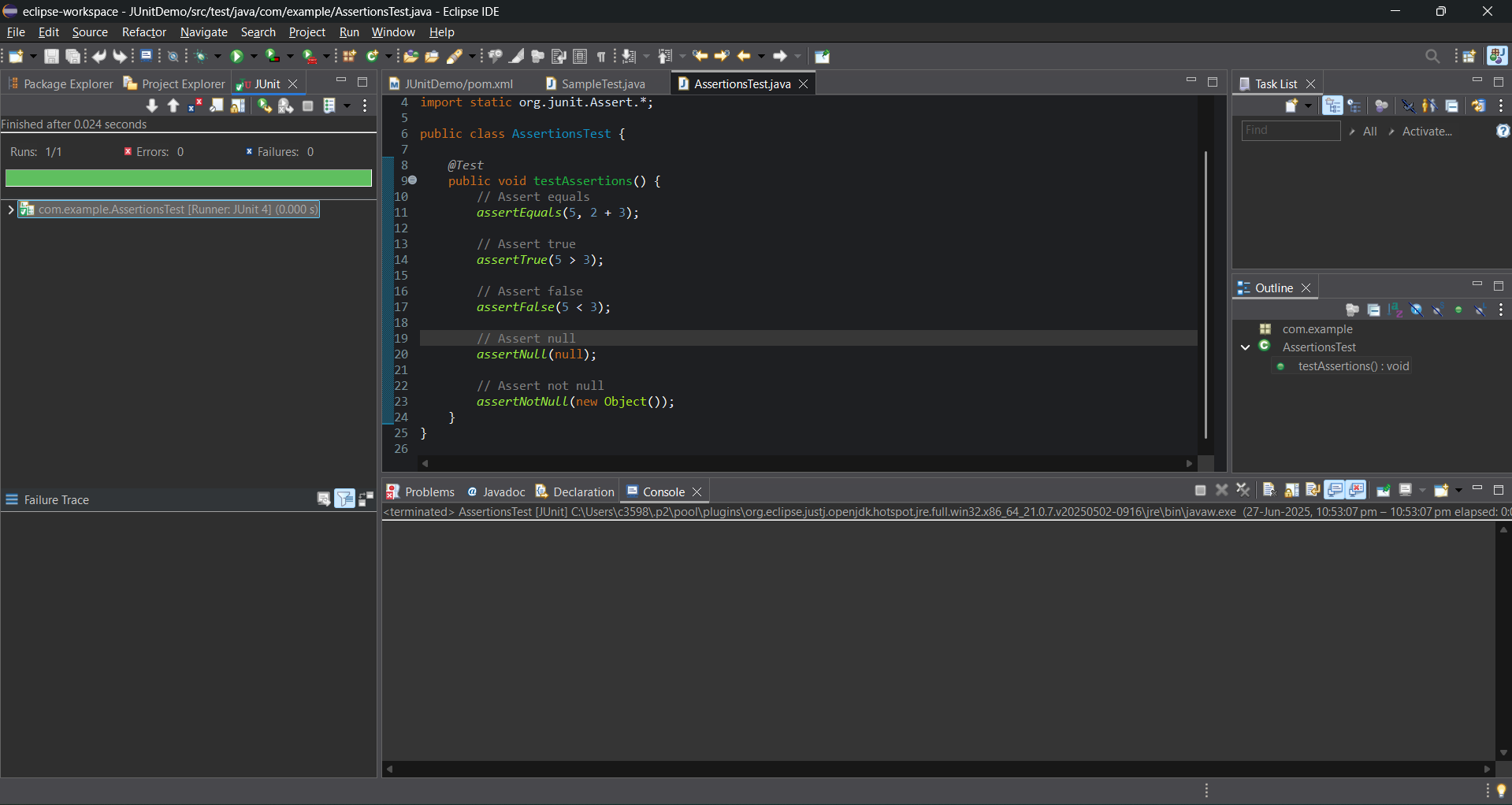
}

SCREENSHOTS ON NEXT PAGE



***OUTPUT***

All assertions passed. The green bar in Eclipse's JUnit runner confirms successful validation of test logic using JUnit assertions.



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

Scenario: You need to organize your tests using the Arrange-Act-Assert (AAA) pattern and use setup and teardown methods.

Creating Calculator.java

package com.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

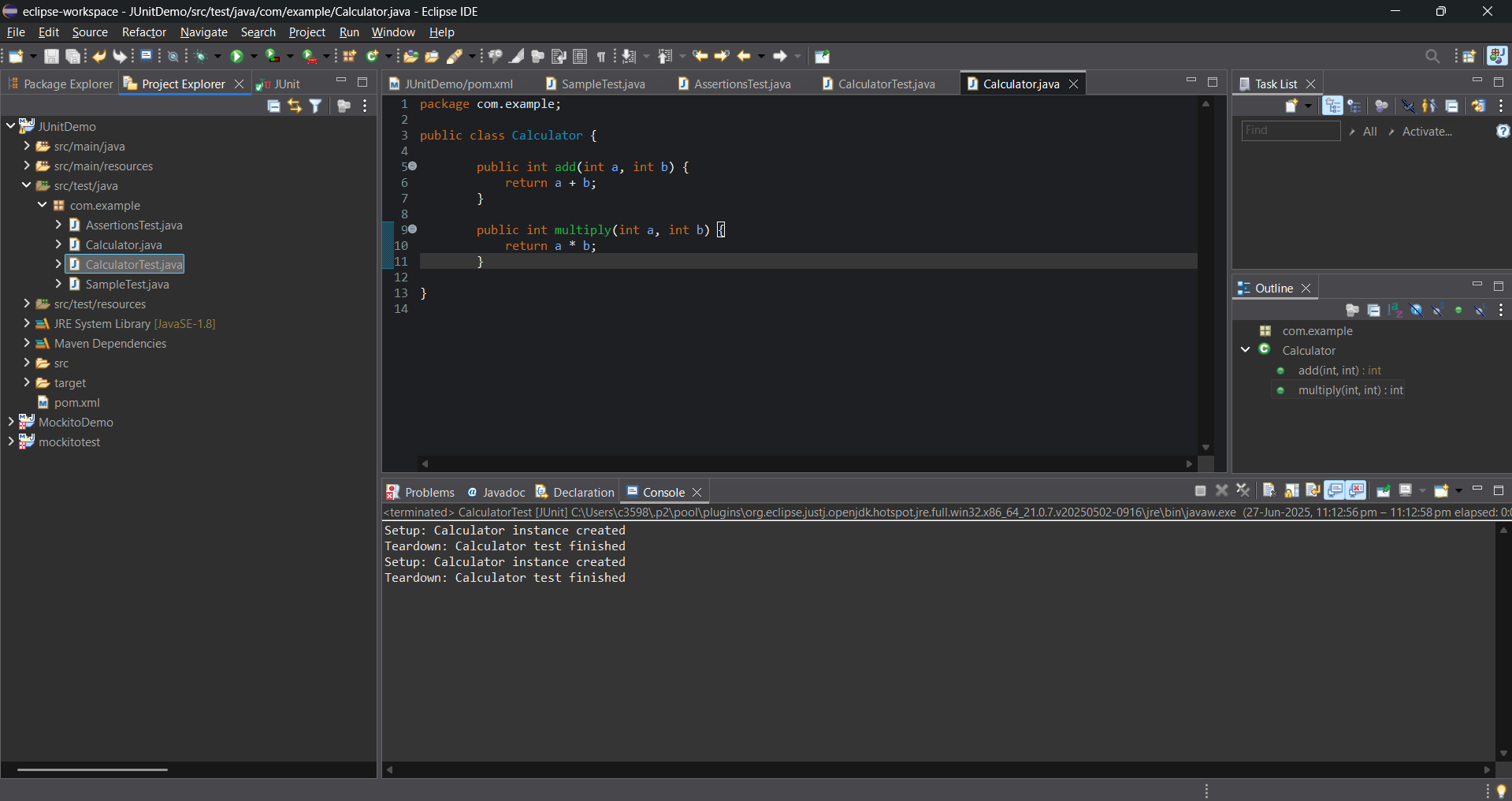
}

public int multiply(int a, int b) {

return a \* b;

}

}



**CalculatorTest.java**

package com.example;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calculator;

@Before

public void setUp() {

calculator = new Calculator(); // Arrange

System.out.println("Setup done");

}

@After

public void tearDown() {

System.out.println("Teardown done");

}

@Test

public void testAddition() {

// Arrange

int a = 5;

int b = 3;

// Act

int result = calculator.add(a, b);

// Assert

assertEquals(8, result);

}

@Test

public void testMultiplication() {

// Arrange

int a = 4;

int b = 2;

// Act

int result = calculator.multiply(a, b);

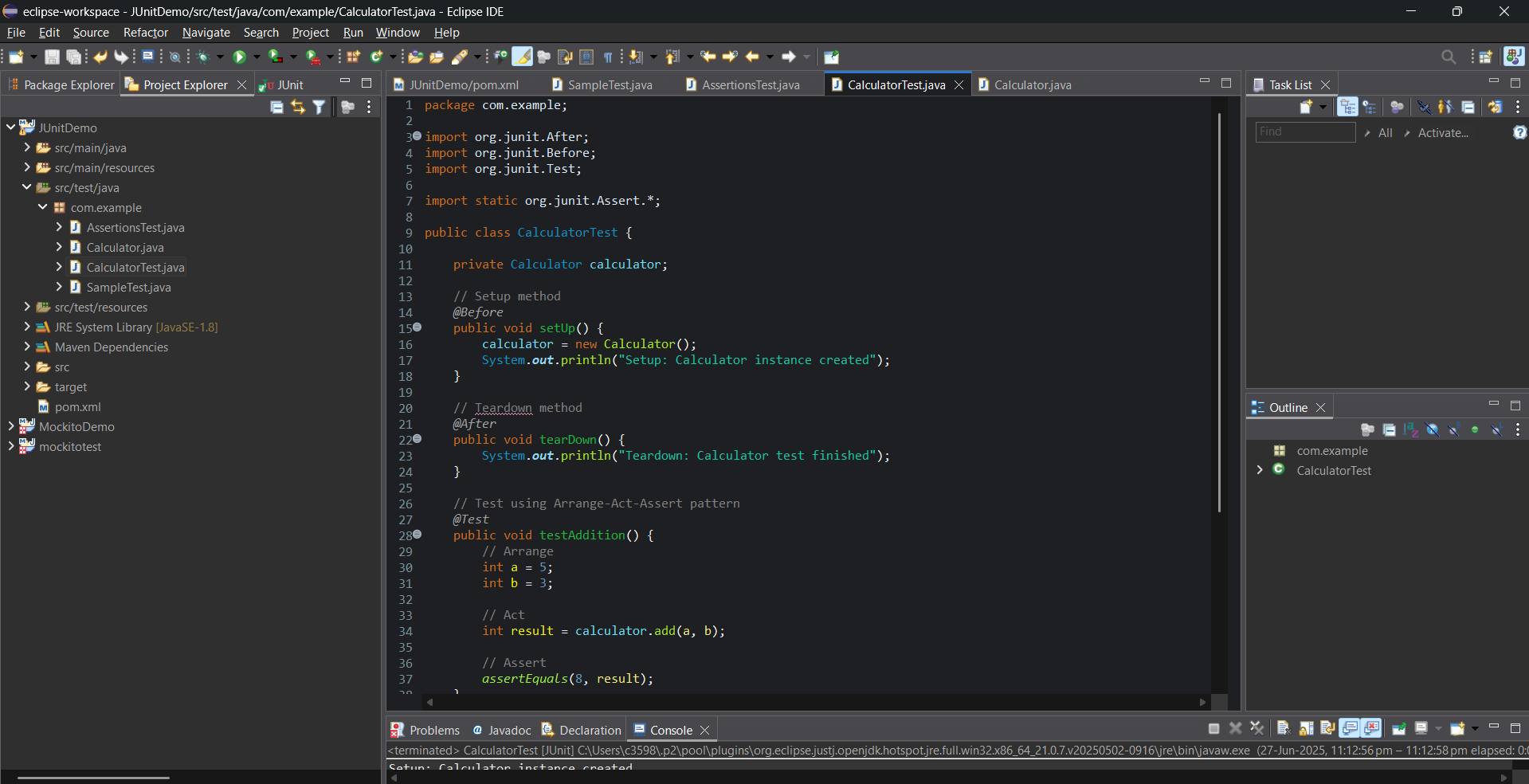
// Assert

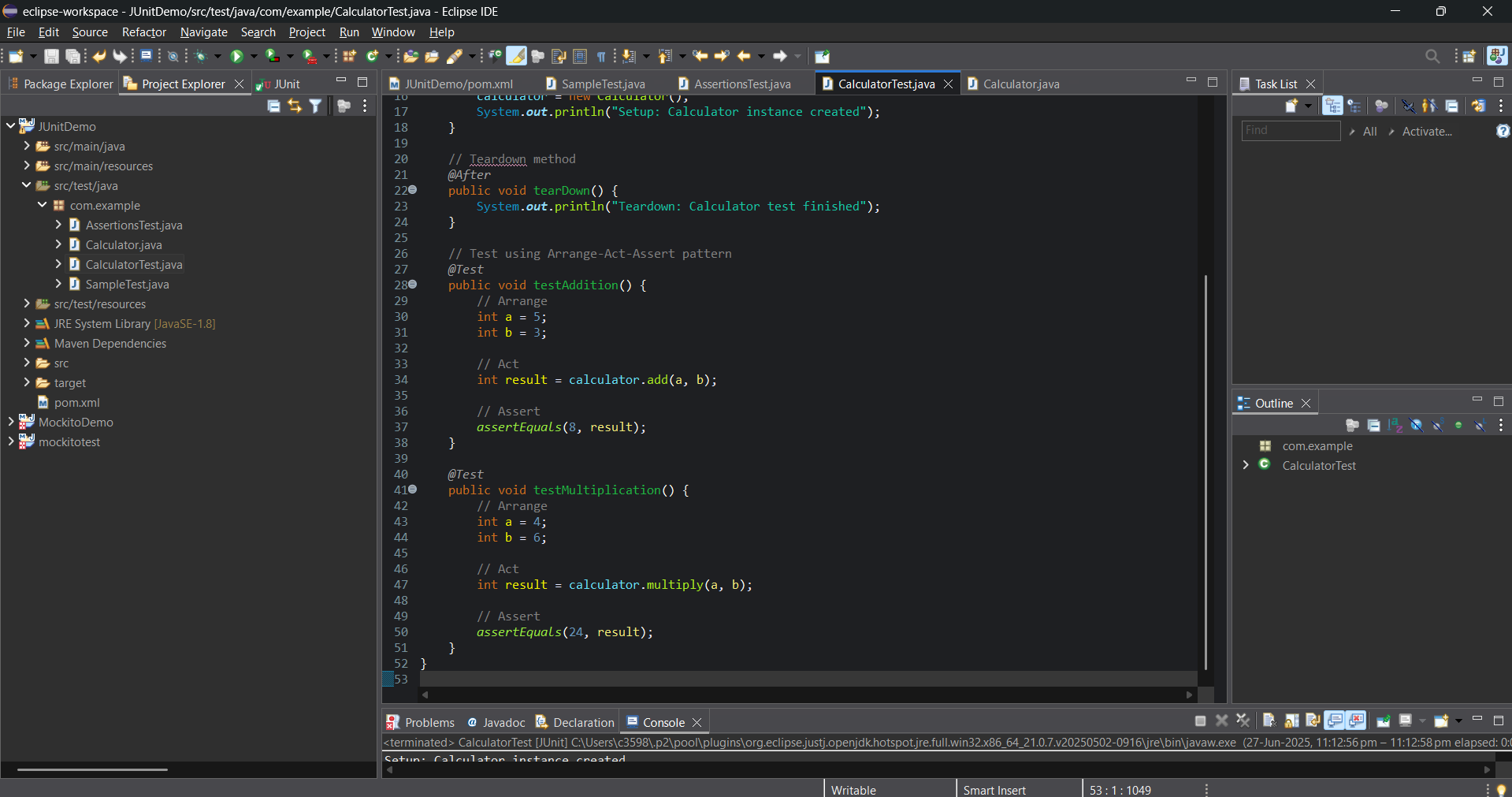
assertEquals(8, result);

}

}

SCREENSHOTS ON NEXT PAGE





OUTPUT

Setup: Calculator instance created

Teardown: Calculator test finished

Setup: Calculator instance created

Teardown: Calculator test finished

