**JUNIT TESTING EXERCISES**

**Exercise 1: Setting Up Junit**

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

**SOLUTION:**

Step 1: Create a Maven Project name ‘**junit-exercises’** on eclipse

Group ID: com.example  
Artifact ID: junit-exercises

Step 2 : Adding Junit dependency in **pom.xml** file

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

Step 3 : Creating a java class **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;

}

}

Step 4 : Creating a test class **CalculatorTest.java**

package com.example;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

*@Test*

public void testAdd() {

Calculator calc = new Calculator();

*assertEquals*(5, calc.add(2, 3));

}

*@Test*

public void testMultiply() {

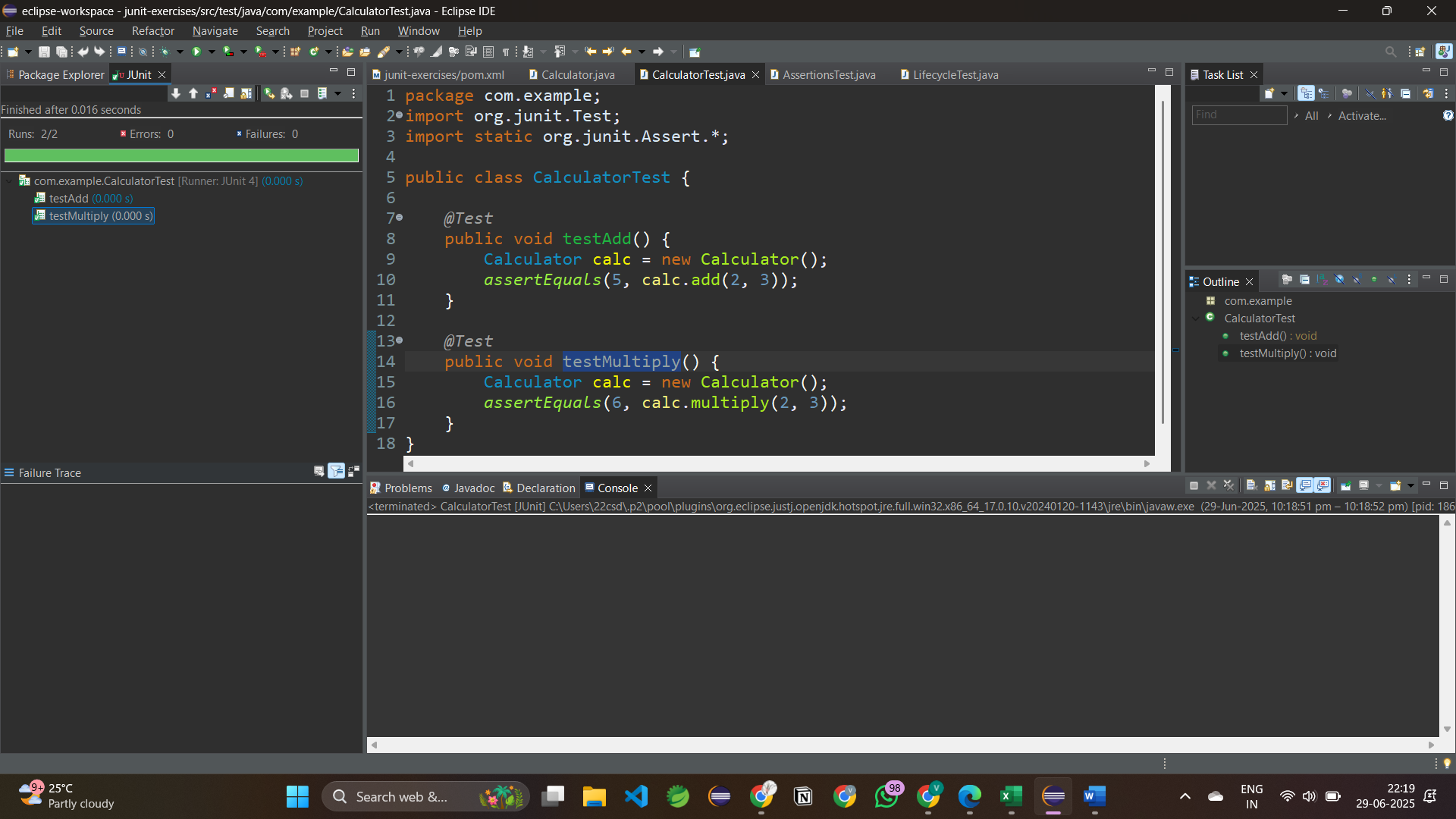
Calculator calc = new Calculator();

*assertEquals*(6, calc.multiply(2, 3));

}

}

**OUTPUT:**



**Exercise 3: Assertions in Junit**

**Scenario:** You need to use different assertions in JUnit to validate your test results.

**SOLUTION:** - **AssertionsTest.java**

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

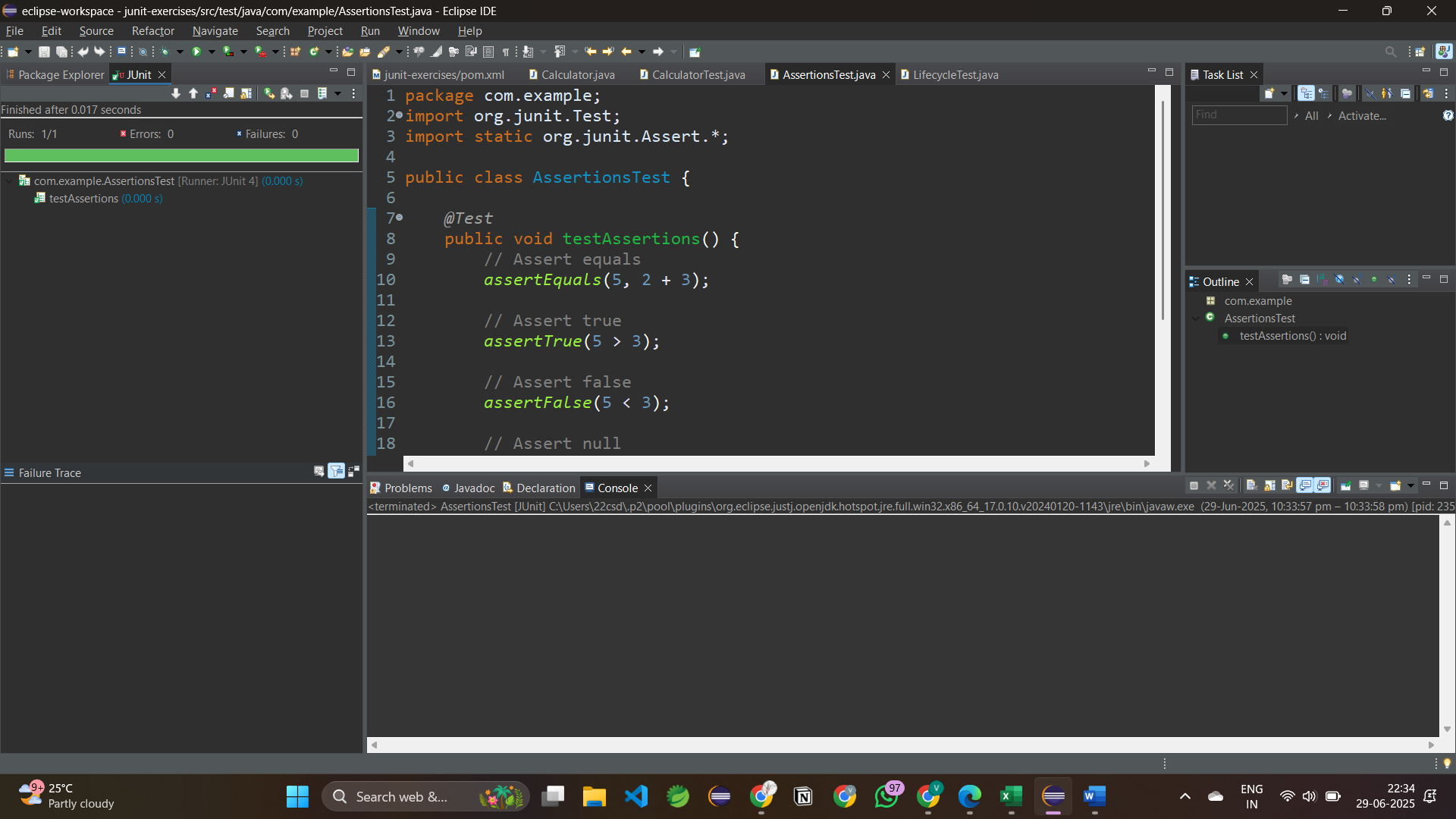
*assertNotNull*(new Object());

*assertNotSame*(5, 3);

}

}

**OUTPUT:**



**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.

Steps:

1. Write tests using the AAA pattern.

2. Use @Before and @After annotations for setup and teardown methods.

**SOLUTION:** - **LifecycleTest.java**

package com.example;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class LifecycleTest {

private Calculator calculator;

*@Before*

public void setUp() {

calculator = new Calculator(); // Arrange

System.*out*.println("Setup before each test");

}

*@After*

public void tearDown() {

System.*out*.println("Cleanup after each test");

}

*@Test*

public void testAdd\_AAA() {

// Act

int result = calculator.add(10, 20);

// Assert

*assertEquals*(30, result);

}

*@Test*

public void testMultiply\_AAA() {

int result = calculator.multiply(3, 4);

*assertEquals*(12, result);

}

}

**OUTPUT:**

