**JUnit Testing Exercises**

**Exercise 1: Setting Up Junit:**

1. 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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>JUnitDemo1</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- JUnit 4.13.2 dependency for unit testing -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

1. **Calculator.java:**

package com.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}}

1. **CalculatorTest.java:**

package com.example;

import static org.junit.Assert.\*;

import org.junit.Test;

public class CalculatorTest {

@Test

public void testAdd() {

Calculator calc = new Calculator();

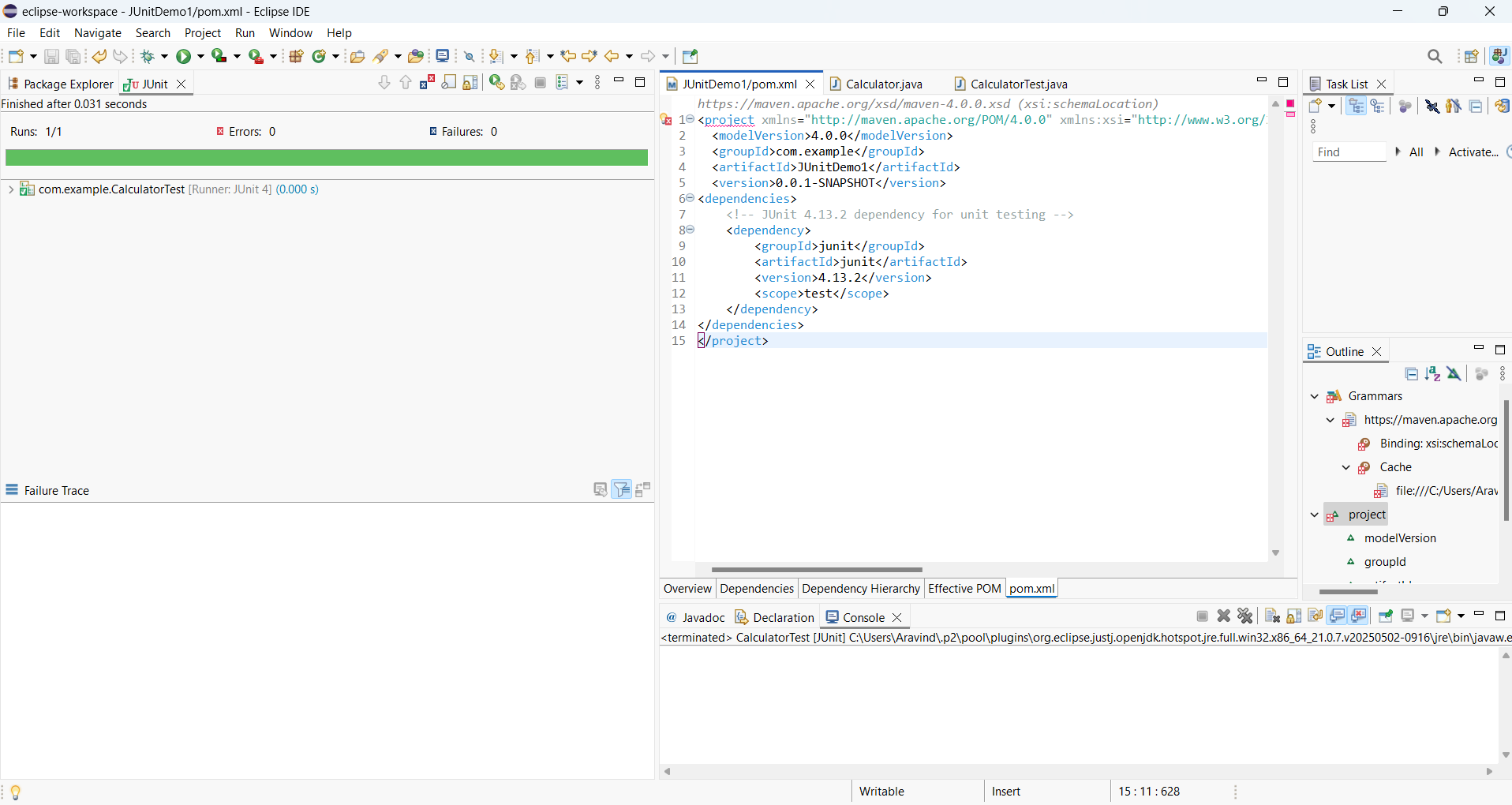
int result = calc.add(2, 3);

assertEquals(5, result);

}

}

**OUTPUT:**



#The TestCase In This Project Is Passed.

**Exercise 3: Assertions in Junit**

1. **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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>AssertionsDemo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

1. **AssertionsTest:**

package com.example;

import static org.junit.Assert.\*;

import org.junit.Test;

public class AssertionsTest {

@Test

public void testAssertions() {

// Assert that two values are equal

assertEquals(5, 2 + 3);

// Assert that the condition is true

assertTrue(5 > 3);

// Assert that the condition is false

assertFalse(5 < 3);

// Assert that the object is null

assertNull(null);

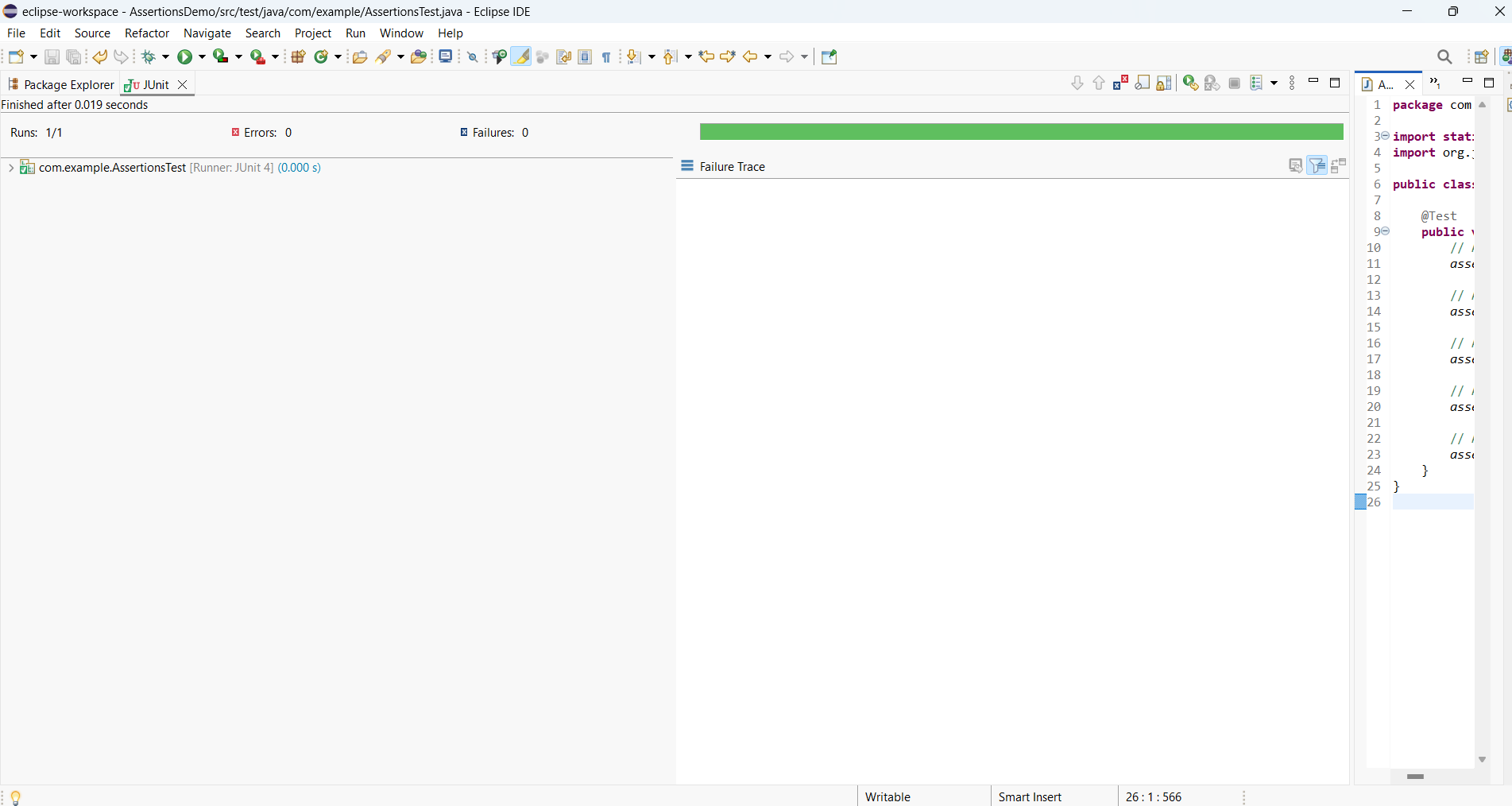
// Assert that the object is not null

assertNotNull(new Object());

}

}

**OUTPUT:**

****

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

**I 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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>JUnitFixturesDemo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

**II 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;

}

}

**III CalculatorTest.java:**

package com.example;

import static org.junit.Assert.\*;

import org.junit.\*;

public class CalculatorTest {

private Calculator calculator;

@Before

public void setUp() {

// Arrange: Set up before each test

calculator = new Calculator();

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

}

@After

public void tearDown() {

// Teardown after each test

calculator = null;

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

}

@Test

public void testAdd() {

// Act

int result = calculator.add(2, 3);

// Assert

assertEquals(5, result);

}

@Test

public void testMultiply() {

// Act

int result = calculator.multiply(4, 5);

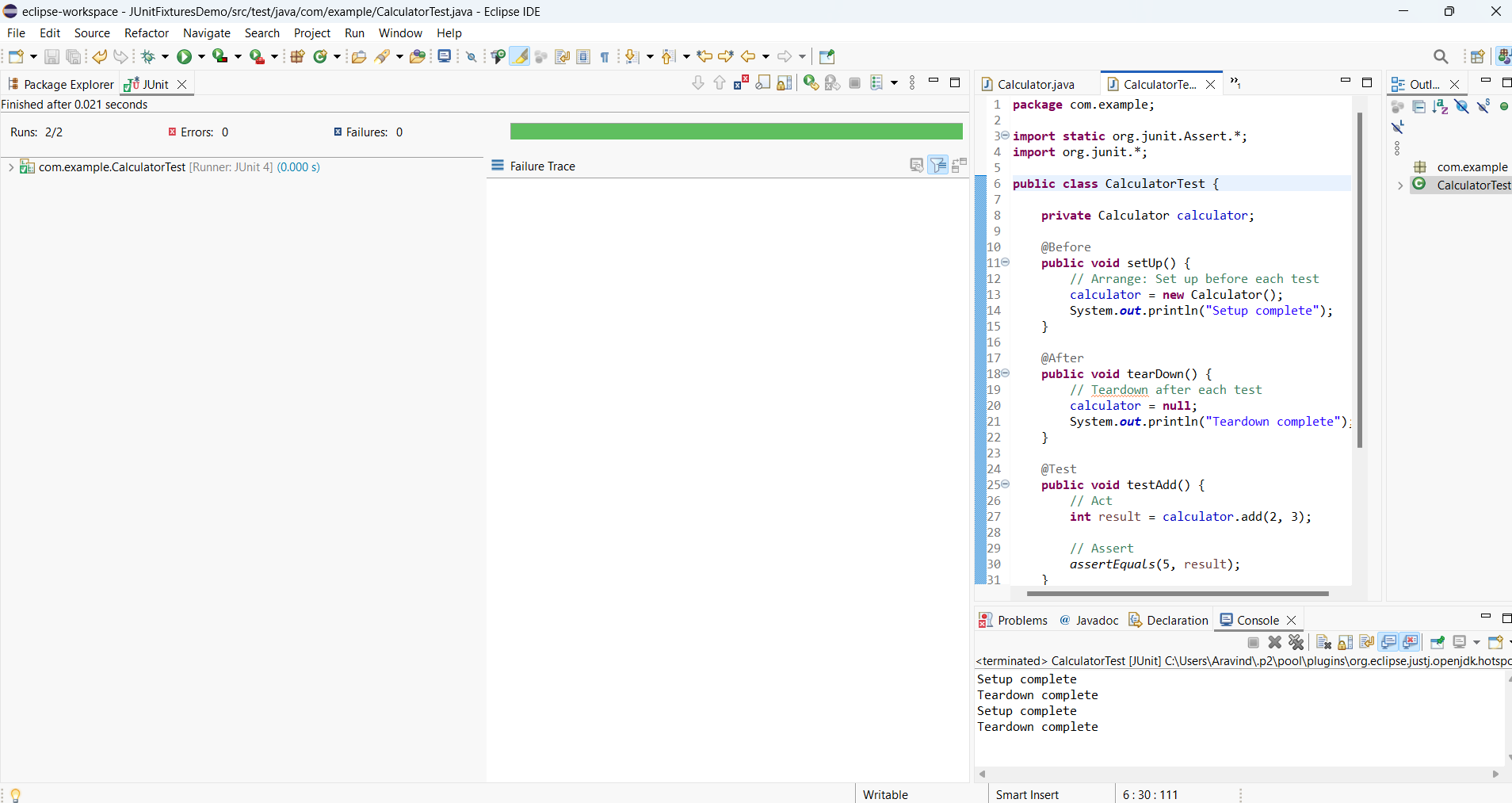
// Assert

assertEquals(20, result);

}

}

**OUTPUT:**

****