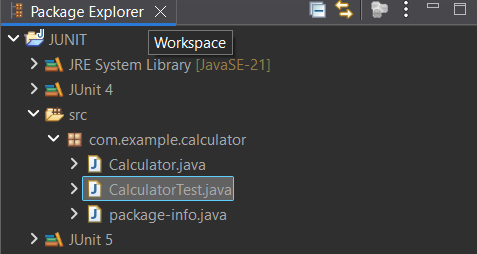
**Exercise 1: Setting Up JUnit**

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

**FOLDER STRUCTURE**

****

**Calculator.java**

package com.example.calculator;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

**CalculatorTest.java**

package com.example.calculator;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

*@Test*

public void testAdd() {

Calculator calc = new Calculator();

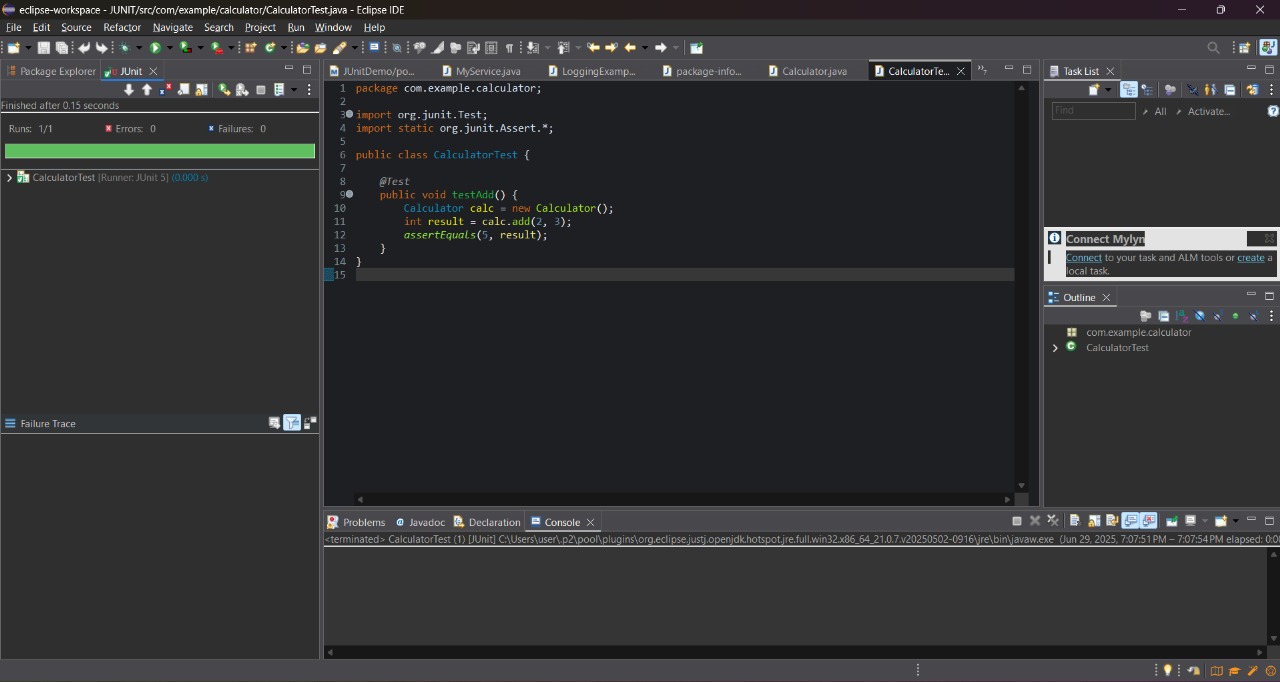
int result = calc.add(2, 3);

*assertEquals*(5, result);

}

}

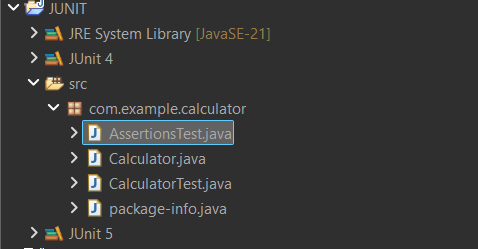
**OUTPUT**



**Exercise 3: Assertions in JUnit**

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

**FOLDER STRUCTURE**

****

**AssertionsTest.java**

package com.example.calculator;

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

*@Test*

public void testAssertions() {

*assertEquals*(10, 5 + 5);

*assertTrue*("Hello".startsWith("H"));

*assertFalse*(100 < 50);

String str = null;

*assertNull*(str);

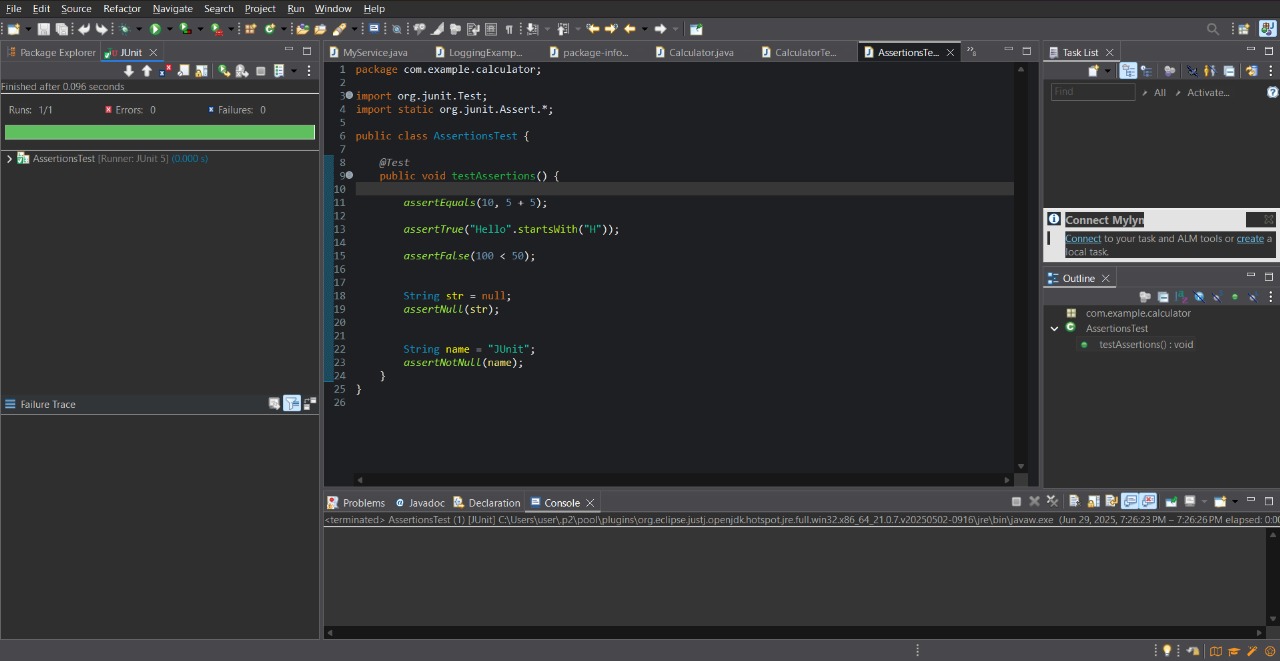
String name = "JUnit";

*assertNotNull*(name);

}

}

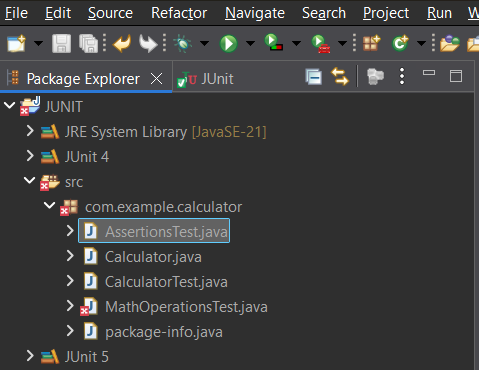
**OUTPUT**



**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.**

**FOLDER STRUCTURE**



**AssertionsTest.java**

package com.example.calculator;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class MathOperationsTest {

private Calculator calculator;

*@Before*

public void setUp() {

calculator = new Calculator();

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

}

*@After*

public void tearDown() {

System.***out***.println("TearDown Complete.");

}

*@Test*

public void testAdditionPositiveNumbers() {

/

int result = calculator.add(7, 8);

*assertEquals*(15, result);

}

*@Test*

public void testAdditionWithZero() {

int result = calculator.add(0, 10);

*assertEquals*(10, result);

}

*@Test*

public void testAdditionWithNegativeNumbers() {

int result = calculator.add(-4, -6);

*assertEquals*(-10, result);

}

}

**OUTPUT**

