**TDD using JUnit5**

**Exercise 1: Setting Up Junit**

**Scenario:**

To Set up JUnit in a Java project to begin writing unit tests.

**Steps**

* Created a new Java project in Eclipse named JUnitDemo.
* Added JUnit Dependency
* Created a new Test Class: File: AssertionsTest.java

**Exercise 3: Assertions in Junit**package com.example.test;

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

@Test

public void testAssertions() {

*assertEquals*(5, 2 + 3);

*assertTrue*(10 > 5);

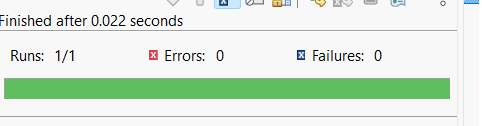
*assertFalse*(3 > 10);

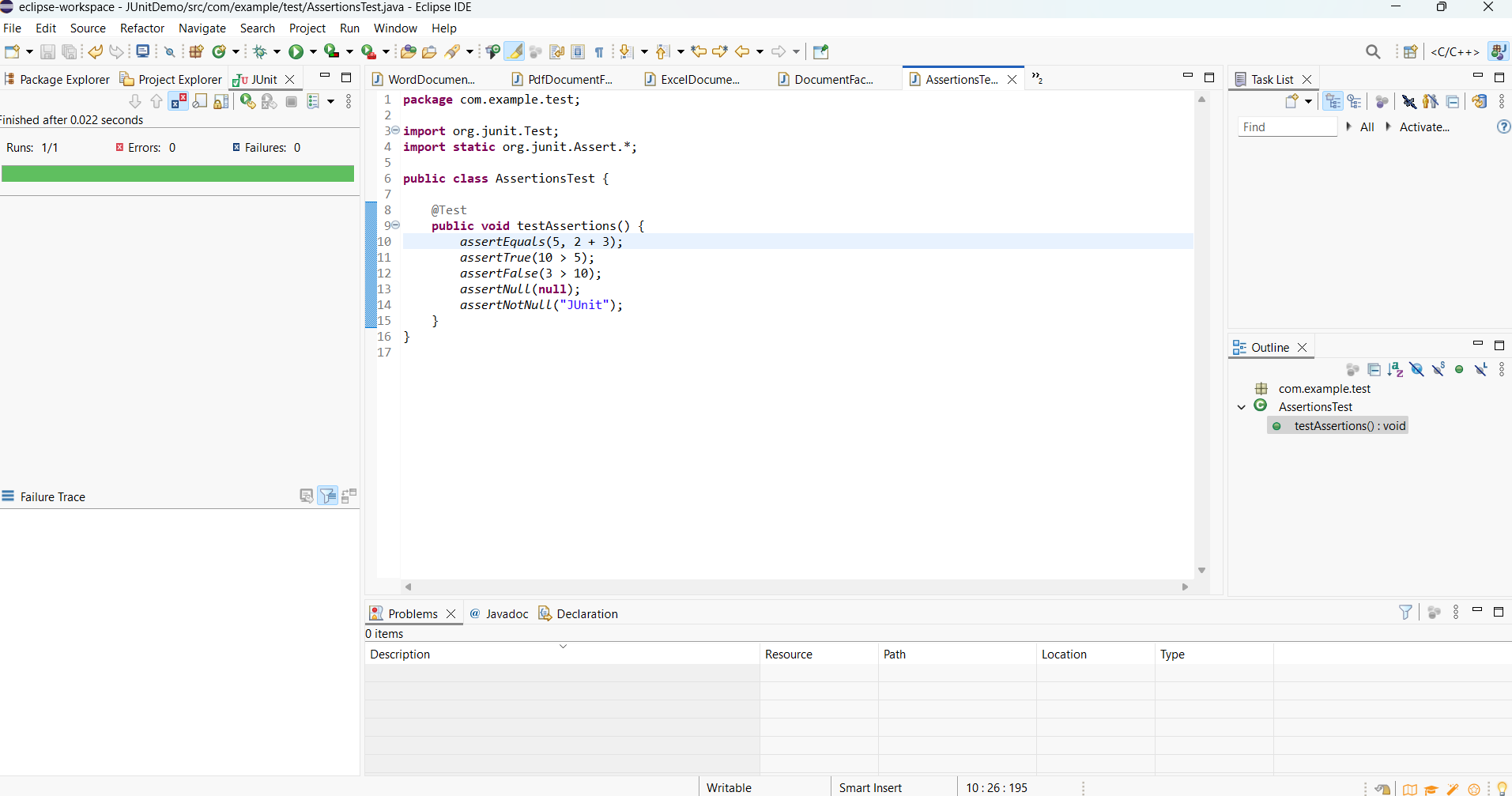
*assertNull*(null);

*assertNotNull*("JUnit");

}

}



****

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

**Step 1**

Arrange-Act-Assert (AAA) pattern to structure the test methods. This pattern improves readability and separates the test logic clearly into three parts:

* Arrange: Prepare the data or inputs.
* Act: Call the method being tested.
* Assert: Check that the actual result matches the expected result.

@Test

public void testAddition() {

// Arrange

int a = 5;

int b = 3;

// Act

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

// Assert

assertEquals(8, result);}

**Step 2:**

**Use @Before and @After annotations for setup and teardown methods**

* @Before: Used for setup logic .
* @After: Used for cleanup logic (releasing resources).

**@Before**

public void setUp() {

calculator = new Calculator();

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

}

**@After**

public void tearDown() {

calculator = null;

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

}

