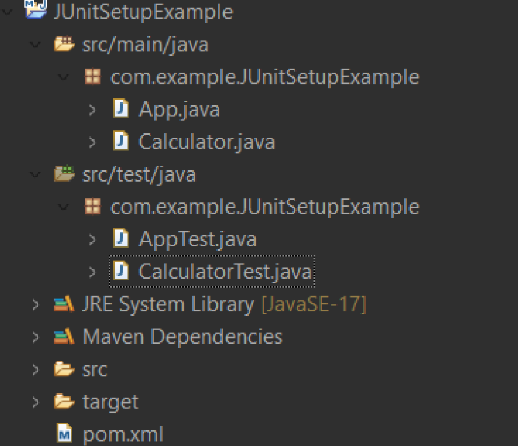
Exercise 1: Setting Up JUnit

Scenario:

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

Steps:

1. Create a new Java project in your IDE (e.g., IntelliJ IDEA, Eclipse).

1. Add JUnit dependency to your project. If you are using Maven, add the following to your pom.xml:

<dependency>

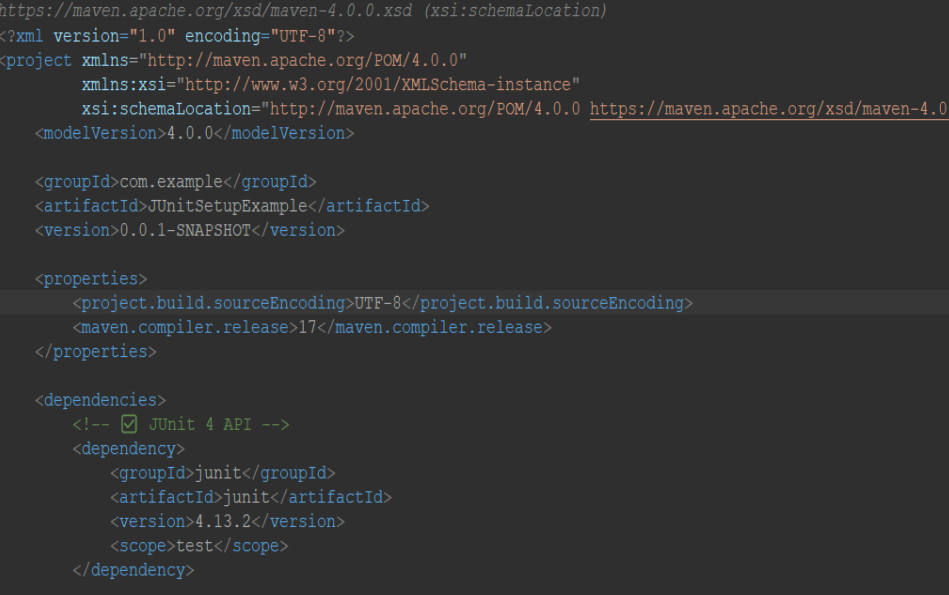
<groupId>junit</groupId>

<artifactId>junit</artifactId>

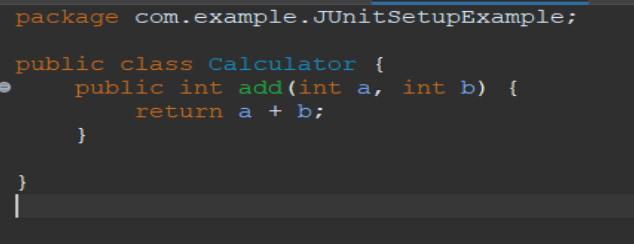
<version>4.13.2</version>

<scope>test</scope>

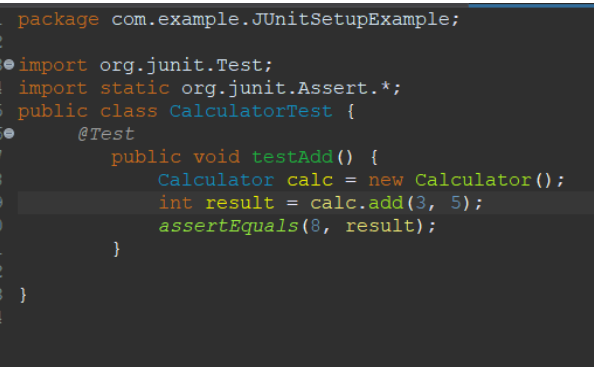
</dependency>



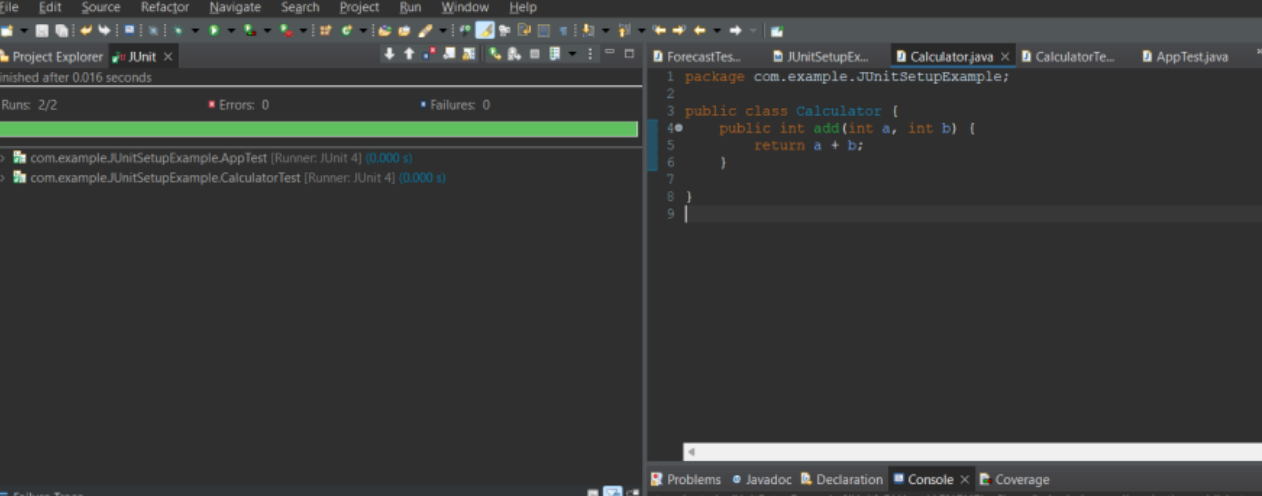
3.Create a new test class in your project.

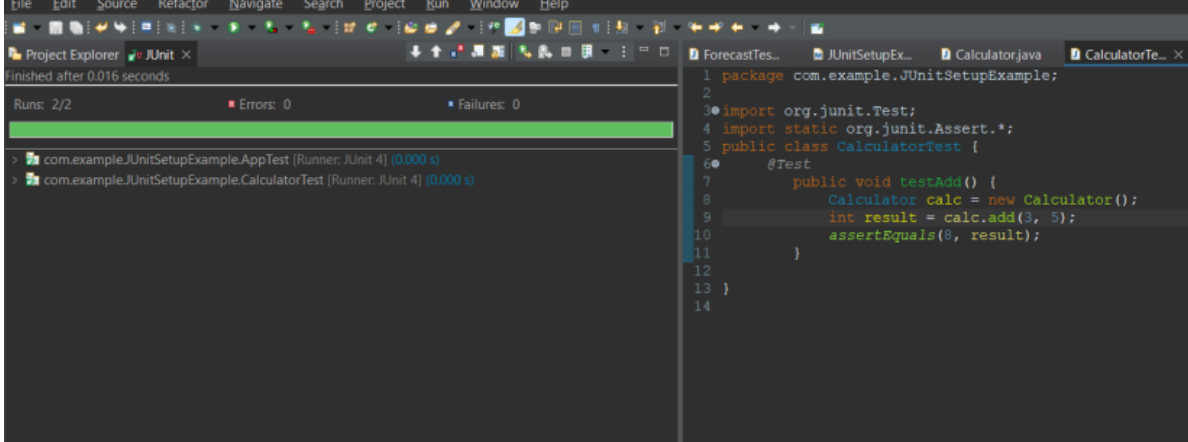
**Class: Calculator.**

**Test Class: CalculatorTest**

****

**Output ScreenShot:**

****

****

**Exercise 3: Assertions in JUnit Scenario: You need to use different assertions in JUnit to validate your test results. Steps: 1.Write tests using various JUnit assertions.**

**Code:**

import static org.junit.Assert

import org.junit.Test;

public class AssertionsTest {

@Test

public void testAssertions() {

assertEquals(5, 2 + 3);

assertTrue(5 > 3);

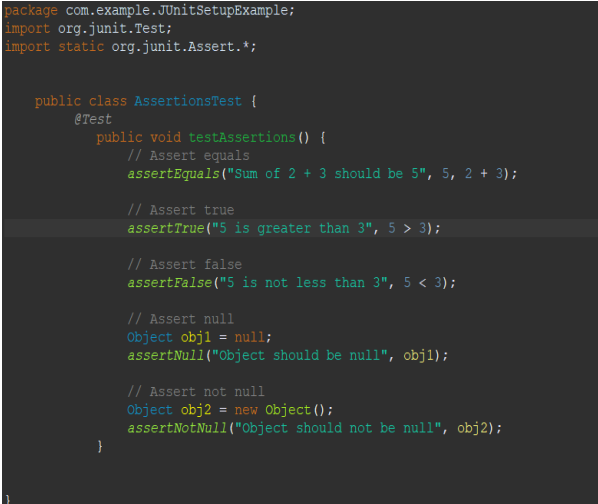
assertFalse(5 < 3);

assertNull(null);

assertNotNull(new Object());

}

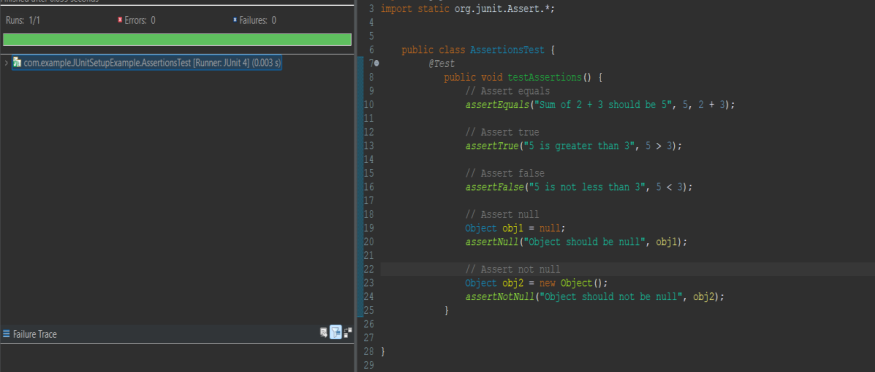
}

**Test class: AssertionsTest**

**2.Test the code by run the class as :**

**Run as> JUnit Test**

**Output Screenshots:**

****

**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.

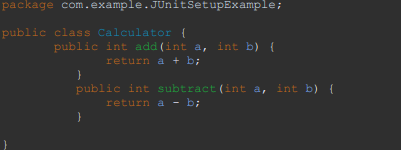
Steps:

1. Write tests using the AAA pattern.

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

3. Run as JUnit Test to get the result for the class.

**Calculator Class:**

****

**CalculatorTest class:**

package com.example.JUnitSetupExample;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calculator;

// Setup method (runs before each test)

@Before

public void setUp() {

System.out.println("Setting up...");

calculator = new Calculator(); // Arrange

}

// Teardown method (runs after each test)

@After

public void tearDown() {

System.out.println("Cleaning up...");

calculator = null;

}

@Test

public void testAddition() {

// Arrange (done in setUp)

// Act

int result = calculator.add(10, 5);

// Assert

assertEquals(15, result);

}

@Test

public void testSubtraction() {

// Arrange (done in setUp)

// Act

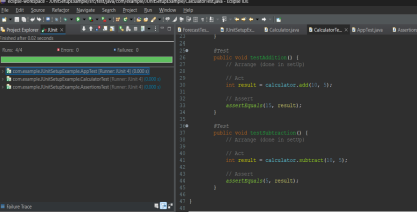
int result = calculator.subtract(10, 5);

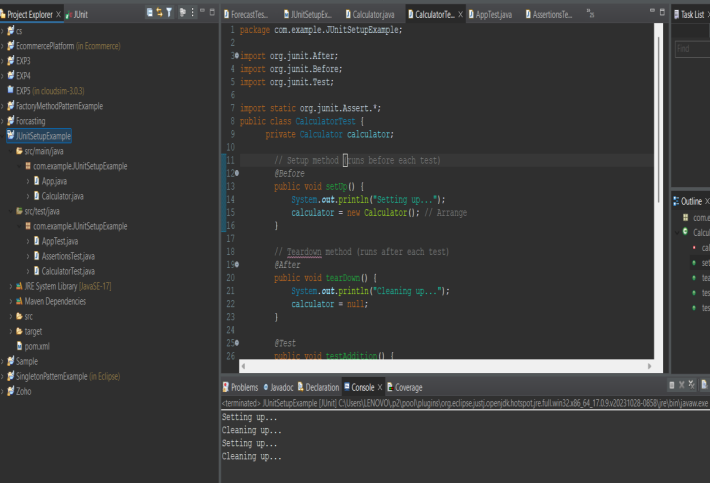
// Assert

assertEquals(5, result);

}

}

**OUTPUT SCREENSHOT:**

****