**Junit Exercise**

**Exercise 1: Setting Up Junit**

Calculator.java

package com.example;

public class Calculator {

public int add(int a ,int b) {

return a+b;

}

}

TestClass.java

package com.example;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

class CaculatorTest {

*@Test*

public void testAdd() {

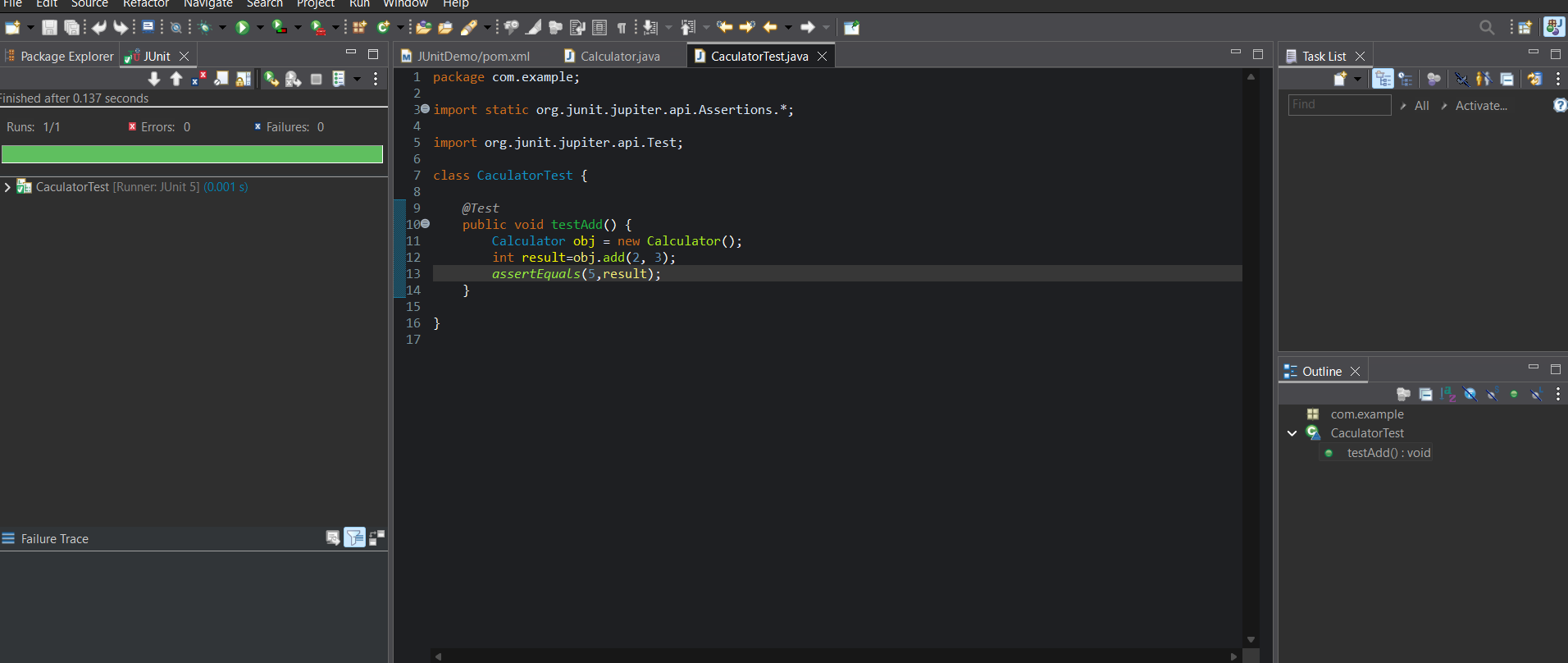
Calculator obj = new Calculator();

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

*assertEquals*(5,result);

}

}



**Exercise 3: Assertion in Junit4**

Added to pom.xml:

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest</artifactId>

<version>2.2</version>

<scope>test</scope>

</dependency>

AssertionTest Class:

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());

}

}



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

Updated Calculator.java

package com.example;

public class Calculator {

public int add(int a ,int b) {

return a+b;

}

public int subtract(int a, int b) {

return a-b;

}

}

Updated CalculatorTest.java

package com.example;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calculator;

// Setup - runs before each test

@Before

public void setUp() {

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

calculator = new Calculator();

}

// Teardown - runs after each test

@After

public void tearDown() {

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

calculator = null;

}

@Test

public void testAddition() {

// Arrange

int a = 5;

int b = 3;

// Act

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

// Assert

assertEquals(8, result);

}

@Test

public void testSubtraction() {

// Arrange

int a = 10;

int b = 4;

// Act

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

// Assert

assertEquals(6, result);

}

}

