**Exercise 1: Setting Up JUnit**

**CODE:**

**Addition Class:**

package Calculator;

public class Add {

public int add(int a, int b) {

return a + b ;

}

}

**Test Class:**

package CalculatorTest;

import Calculator.Add;

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

import org.junit.jupiter.api.Test;

class AddTest {

Add a = new Add();

@Test

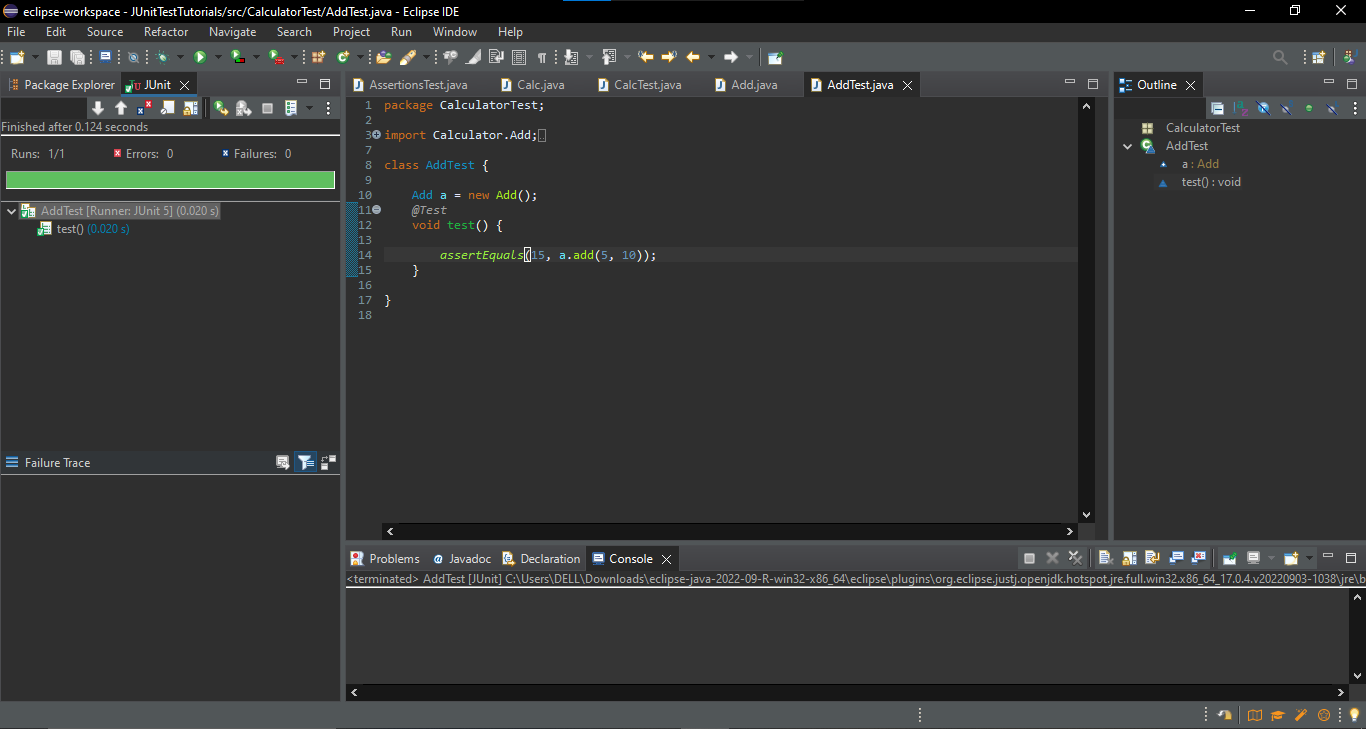
void test() {

assertEquals(15, a.add(5, 10));

}

}

**OUTPUT:**

****

**Exercise 3: Assertions in JUnit**

**CODE:**

package CalculatorTest;

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

import org.junit.jupiter.api.Test;

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);

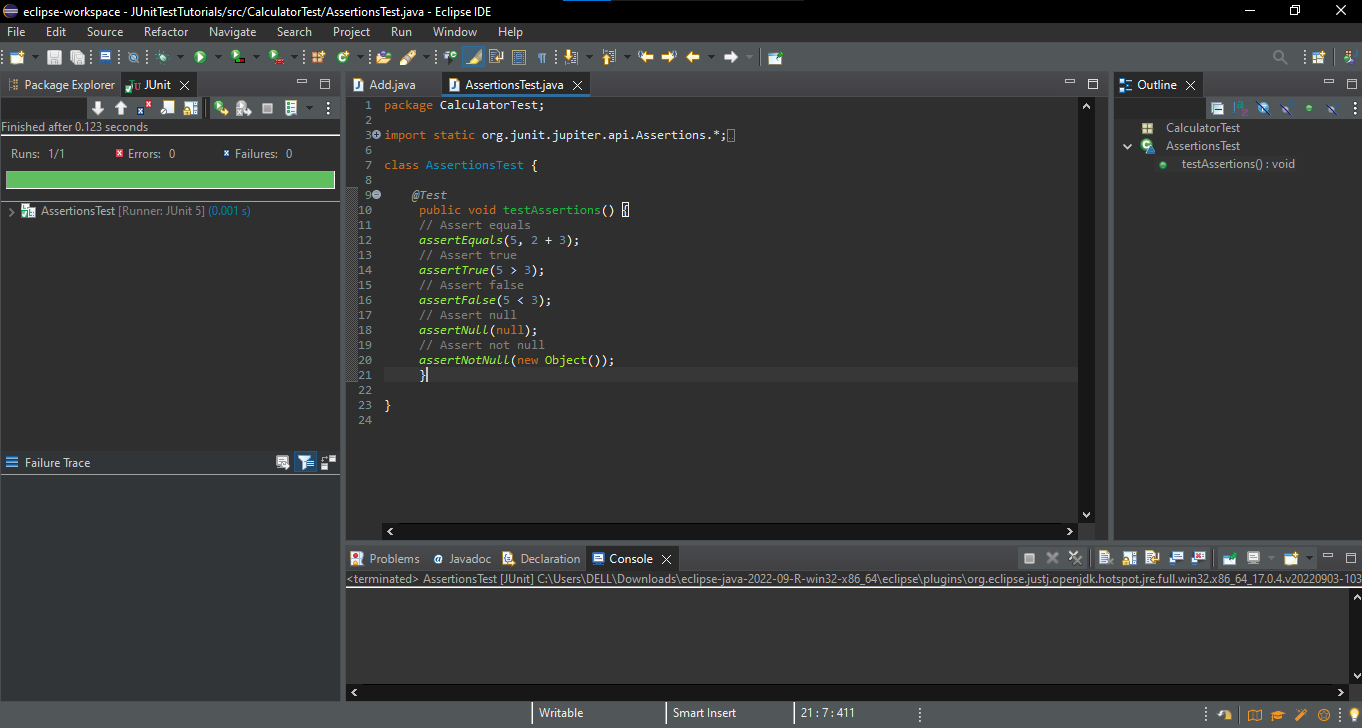
// Assert not null

assertNotNull(new Object());

}

}

**OUTPUT:**

****

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

**CODE:**

**Calculator Class:**

package Calculator;

public class Calc {

public int add (int a, int b) {

return a+b ;

}

public int sub(int a, int b) {

return a-b ;

}

}

**Test Class:**

package CalculatorTest;

import Calculator.Calc;

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

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

class CalcTest {

Calc calculator;

@BeforeEach

public void setup() {

System.out.println("Setup: Creating new Calculator instance...");

calculator = new Calc(); // Test fixture

}

@AfterEach

public void teardown() {

System.out.println(" Teardown: Test completed.\n");

}

@Test

public void testAddition() {

//Arrange

int a = 10, b = 5;

//Act

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

//Assert

assertEquals(15, result, "Addition should return correct sum");

}

@Test

public void testSubtraction() {

// Arrange

int a = 10, b = 3;

// Act

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

//Assert

assertEquals(7, result, "Subtraction should return correct difference");

}

}

