JUnit Testing Exercises

Exercise 1: Setting Up Junit

**File: Calculator.java**

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

**File: CalculatorTest.java**

import org.junit.jupiter.api.Test;

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

class CalculatorTest {

@Test

void testAdd() {

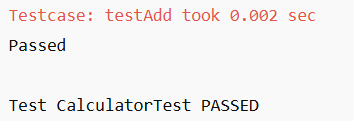
Calculator calc = new Calculator();

assertEquals(5, calc.add(2, 3));

}

}

**OUTPUT:**

****

Exercise 2: Writing Basic JUnit Tests

**File: MathOperations.java**

public class MathOperations {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

public int multiply(int a, int b) {

return a \* b;

}

}

**File: MathOperationsTest.java**

import org.junit.jupiter.api.Test;

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

class MathOperationsTest {

@Test

void testAdd() {

MathOperations m = new MathOperations();

assertEquals(10, m.add(7, 3));

}

@Test

void testSubtract() {

MathOperations m = new MathOperations();

assertEquals(4, m.subtract(9, 5));

}

@Test

void testMultiply() {

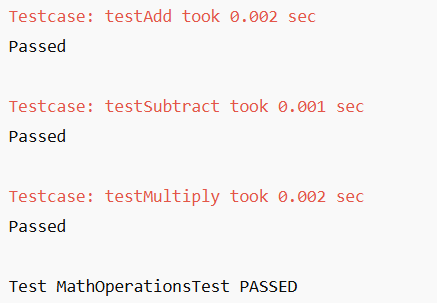
MathOperations m = new MathOperations();

assertEquals(20, m.multiply(4, 5));

}

}

**OUTPUT:**

****

**Exercise 3: Assertions in Junit**

**File: AssertionsTest.java**

import org.junit.jupiter.api.Test;

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

class AssertionsTest {

@Test

void testAssertions() {

assertEquals(5, 2 + 3);

assertTrue(5 > 3);

assertFalse(5 < 3);

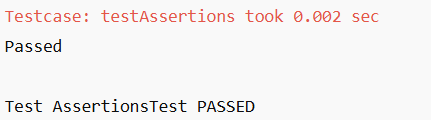
assertNull(null);

assertNotNull(new Object());

}

}

**OUTPUT:**

****

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

**File: Calculator.java**

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

}

**File: CalculatorLifecycleTest.java**

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

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

class CalculatorLifecycleTest {

Calculator calc;

@BeforeEach

void setUp() {

calc = new Calculator();

}

@Test

void testAdd() {

int result = calc.add(4, 6);

assertEquals(10, result);

}

@Test

void testSubtract() {

int result = calc.subtract(9, 2);

assertEquals(7, result);

}

@AfterEach

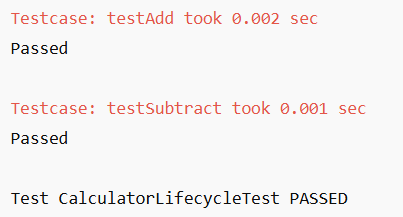
void tearDown() {

calc = null;

}

}

**OUTPUT:**

****