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.

package jUnitExercises;

public class Calculator {

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;

}

public int divide(int a, int b) {

return a / b;

}

}

package jUnitExercises;

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

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

public class CalculatorTestClass {

private Calculator c;

@BeforeEach

public void setUp() {

c = new Calculator();

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

}

@AfterEach

public void tearDown() {

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

}

@Test

public void testAddition() {

int res = c.add(2, 3);

assertEquals(5, res, "2 + 3 should be equal to 5");

}

@Test

public void testSubtraction() {

int res = c.subtract(5, 2);

assertEquals(3, res, "5 - 2 should be equal to 3");

}

@Test

public void testMultiplication() {

int res = c.multiply(2, 3);

assertEquals(6, res, "2 \* 3 should be equal to 6");

}

@Test

public void testDivision() {

int res = c.divide(6, 2);

assertEquals(3, res, "6 / 2 should be equal to 3");

}

}