**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.

**Solution**

Folder structure as follows:

demo/

├── pom.xml

└── src/

├── main/java/com/example/Calculator.java

└── test/java/com/example/CalculatorTest.java

**Code for CalculatorTest.java**

package com.example;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

    private Calculator calculator;

    // Setup: runs before every @Test method

    @Before

    public void setUp() {

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

        calculator = new Calculator();

    }

    // Teardown: runs after every @Test method

    @After

    public void tearDown() {

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

        calculator = null;

    }

    @Test

    public void testAdd() {

        // Arrange

        int a = 10, b = 5;

        // Act

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

        // Assert

        assertEquals(15, result);

    }

    @Test

    public void testSubtract() {

        // Arrange

        int a = 10, b = 5;

        // Act

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

        // Assert

        assertEquals(5, result);

    }

    @Test

    public void testDivideByZero() {

        // Arrange

        int a = 10, b = 0;

        // Act + Assert

        try {

            calculator.divide(a, b);

            fail("Expected IllegalArgumentException for division by zero");

        } catch (IllegalArgumentException e) {

            assertEquals("Division by zero", e.getMessage());

        }

    }

}

**Code for 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;

    }

    public int multiply(int a, int b) {

        return a \* b;

    }

    public int divide(int a, int b) {

        if (b == 0) {

            throw new IllegalArgumentException("Division by zero");

        }

        return a / b;

    }

}

**Terminal Command**

mvn test

**OUTPUT**

A screenshot of a computer screen

AI-generated content may be incorrect.