**Exercise 4 - Arrange-Act-Assert (AAA) Pattern**

**Pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.sudip</groupId>

<artifactId>Arrange-Act-Assert\_Pattern</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter</artifactId>

<version>5.10.2</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.0.0-M9</version>

</plugin>

</plugins>

</build>

</project>

**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;

}

public double divide(double a, double b) {

if (b == 0) {

throw new ArithmeticException("Division by zero");

}

return a / b;

}

}

**CalculatorTest.java**

import org.junit.jupiter.api.AfterEach;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

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

public class CalculatorTest {

private Calculator calculator;

@BeforeEach

public void setUp() {

// Initialize the Calculator instance before each test

calculator = new Calculator();

}

@AfterEach

public void tearDown() {

// Clean up by setting calculator to null after each test

calculator = null;

}

@Test

public void testAdd() {

int a = 5;

int b = 3;

int expected = 8;

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

assertEquals(expected, result, "Adding " + a + " and " + b + " should equal " + expected);

}

@Test

public void testSubtract() {

int a = 5;

int b = 3;

int expected = 2;

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

assertEquals(expected, result, "Subtracting " + b + " from " + a + " should equal " + expected);

}

@Test

public void testDivide() {

double a = 6.0;

double b = 2.0;

double expected = 3.0;

double result = calculator.divide(a, b);

assertEquals(expected, result, 0.0001, "Dividing " + a + " by " + b + " should equal " + expected);

}

@Test

public void testDivideByZero() {

double a = 5.0;

double b = 0.0;

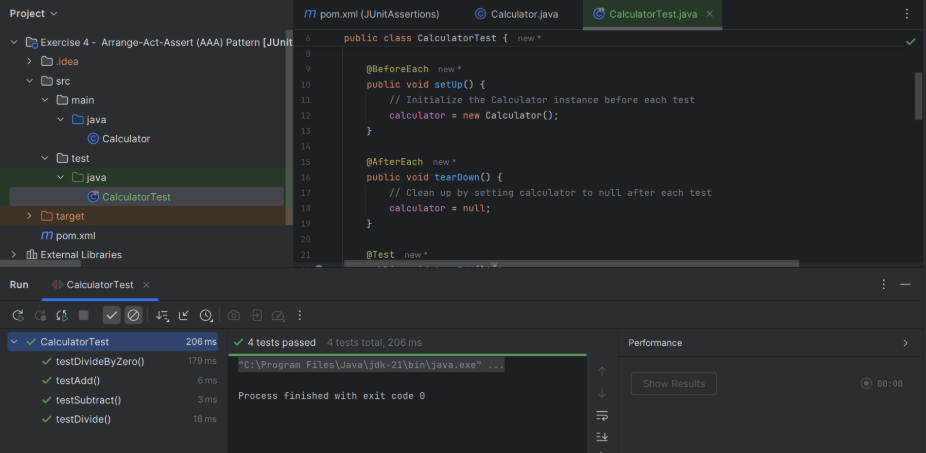
assertThrows(ArithmeticException.class, () -> calculator.divide(a, b),

"Dividing " + a + " by " + b + " should throw ArithmeticException");

}

}

**OUTPUT :**

****