**TDD USING Junit5 AND MOCKITO**

**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

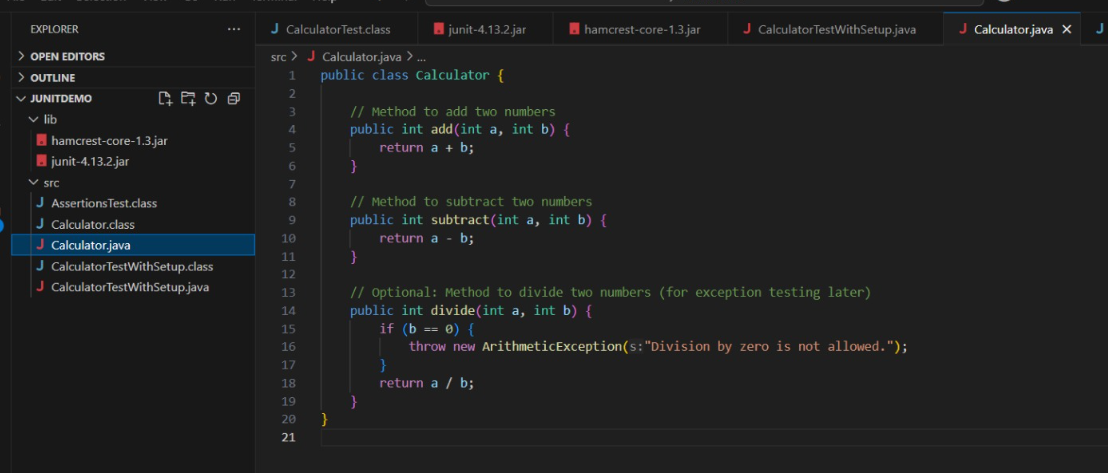
and teardown methods.

**Steps:**

1. Write tests using the AAA pattern.

2. Use @Before and @After annotations for setup and teardown method

**FOLDER STRUCTURE**

****

**PROGRAM**

**CalculatorTestWithSetup.java**

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTestWithSetup {

private Calculator calc; // Test Fixture (common test object)

@Before

public void setUp() {

// Setup - Runs before every @Test method

calc = new Calculator();

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

}

@After

public void tearDown() {

// Teardown - Runs after every @Test method

calc = null;

System.out.println("Teardown: Calculator instance cleared.");

}

@Test

public void testAddition() {

// Arrange

int a = 5;

int b = 3;

// Act

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

// Assert

assertEquals(8, result);

}

@Test

public void testSubtraction() {

// Arrange

int a = 10;

int b = 4;

// Act

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

// Assert

assertEquals(6, result);

}

}

**Calculator.java**

public class Calculator {

// Method to add two numbers

public int add(int a, int b) {

return a + b;

}

// Method to subtract two numbers

public int subtract(int a, int b) {

return a - b;

}

// Method to divide two numbers (for exception testing later)

public int divide(int a, int b) {

if (b == 0) {

throw new ArithmeticException("Division by zero is not allowed.");

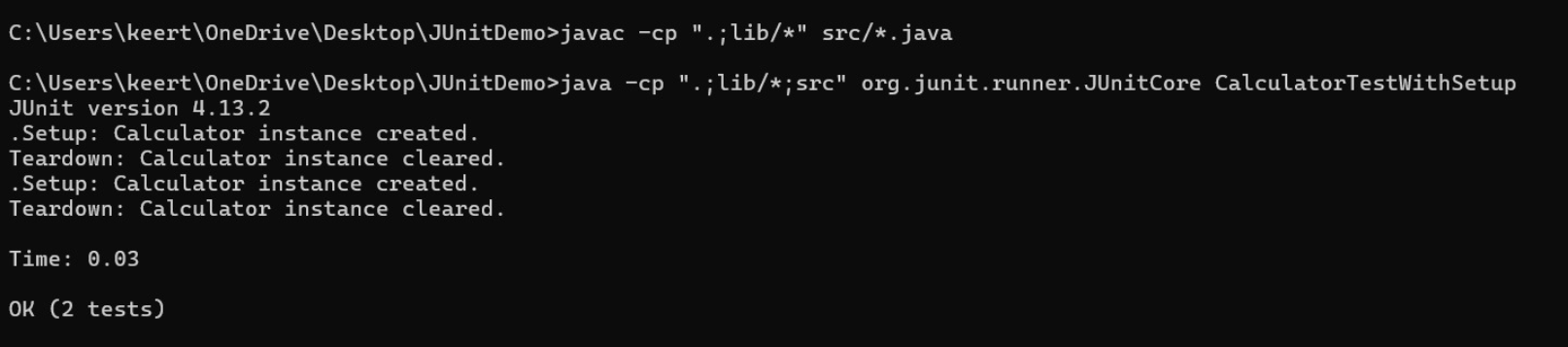
}

return a / b;

}

}

**OUTPUT**

****