**WEEK-2**

**Exercise 3: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup.**

**1) AAA Pattern Testing:**

**CODE:**

**Calculator.java:**

**package** calculator;

**public** **class** Calculator {

**public** **int** divide(**int** a, **int** b) {

**if** (b == 0) **throw** **new** ArithmeticException("Divide by zero");

**return** a / b;

}

**public** **int** square(**int** a) {

**return** a \* a;

}

}

**CalculatorTest.java:**

**package** calculator;

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

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

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

**public** **class** CalculatorTest {

**private** Calculator calc;

@BeforeEach

**void** setUp() {

calc = **new** Calculator(); // Arrange

}

@Test

**void** testDivide() {

// Act

**int** result = calc.divide(10, 2);

// Assert

*assertEquals*(5, result, "10 divided by 2 should be 5");

}

@Test

**void** testSquare() {

// Act

**int** result = calc.square(4);

// Assert

*assertEquals*(16, result, "Square of 4 should be 16");

}

@Test

**void** testDivideByZero() {

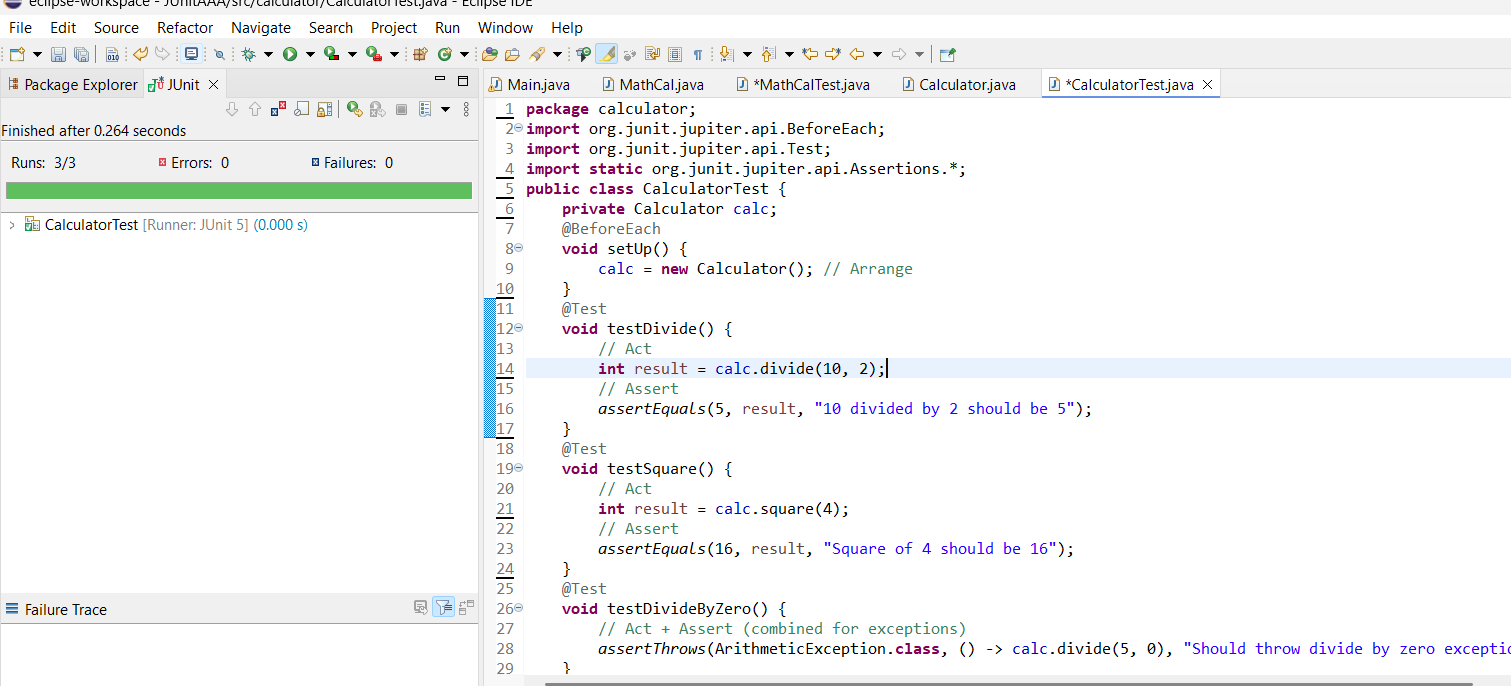
// Act + Assert (combined for exceptions)

*assertThrows*(ArithmeticException.**class**, () -> calc.divide(5, 0), "Should throw divide by zero exception");

}

}

**OUTPUT:**

****