**Experiment – 1**

**Scenario: You need to set up JUnit in your Java project to start writing unit tests**

**Steps:**

1. **Create a new Java project in your IDE**

Created a project with Group Id: com.example & Artifact Id: calculator

1. **Add JUnit dependency to your project. If you are using Maven, add the following to your pom.xml:**

<dependency>

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

<artifactId>junit-jupiter</artifactId>

<version>5.10.2</version>

<scope>test</scope>

</dependency>

1. **Create a new test class in your project.**

* **Create a new Java Class**

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 double divide(int a, int b) {

if (b == 0) throw new ArithmeticException("Cannot divide by zero.");

return (double) a / b;

}

public int modulus(int a, int b) {

if (b == 0) throw new ArithmeticException("Cannot perform modulus by zero.");

return a % b;

}

public double power(double base, double exponent) {

return Math.*pow*(base, exponent);

}

public double squareRoot(double value) {

if (value < 0) throw new ArithmeticException("Cannot take square root of a negative number.");

return Math.*sqrt*(value);

}

}

* **Create a JUnit Test Class**

package com.example;

import org.junit.jupiter.api.Test;

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

class CalculatorTest {

Calculator calc = new Calculator();

@Test

void testAddition() {

*assertEquals*(5, calc.add(2, 3));

}

@Test

void testSubtraction() {

*assertEquals*(1, calc.subtract(4, 3));

}

@Test

void testMultiplication() {

*assertEquals*(12, calc.multiply(4, 3));

}

@Test

void testDivision() {

*assertEquals*(2.0, calc.divide(6, 3));

}

@Test

void testDivisionByZero() {

*assertThrows*(ArithmeticException.class, () -> calc.divide(5, 0));

}

@Test

void testModulus() {

*assertEquals*(1, calc.modulus(10, 3));

}

@Test

void testModulusByZero() {

*assertThrows*(ArithmeticException.class, () -> calc.modulus(10, 0));

}

@Test

void testPower() {

*assertEquals*(8.0, calc.power(2, 3));

}

@Test

void testSquareRoot() {

*assertEquals*(3.0, calc.squareRoot(9));

}

@Test

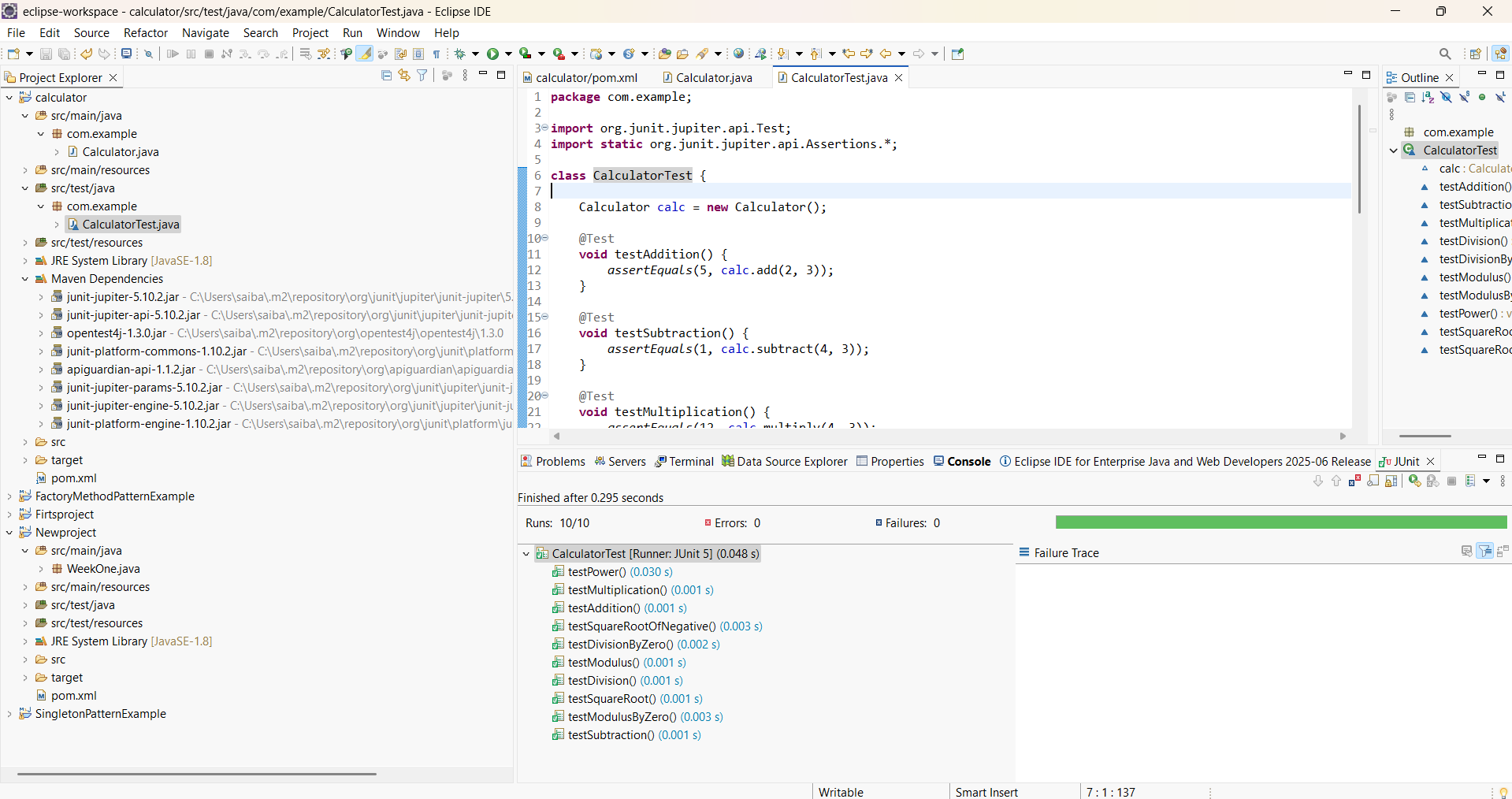
void testSquareRootOfNegative() {

*assertThrows*(ArithmeticException.class, () -> calc.squareRoot(-4));

}

}

**Output**

****

Exercise 3: Assertions in Junit

Scenario: You need to use different assertions in JUnit to validate your test results.

**Step 1:** Update the pom.xml file with Junit 5 in eclipse

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.cognizant</groupId>

<artifactId>cognizant</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- JUnit 5 dependency -->

<dependency>

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

<artifactId>junit-jupiter</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Needed to run JUnit 5 -->

<plugin>

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

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

<version>3.1.2</version>

</plugin>

</plugins>

</build>

</project>

**Step 2:** Create a test class named Assertion Test in src/test.java

package com.cognizant;

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

import org.junit.jupiter.api.Test;

public class AssertionsTest {

@Test

public void testAssertions() {

assertEquals(5, 2 + 3);

assertTrue(5 > 3);

assertFalse(5 < 3);

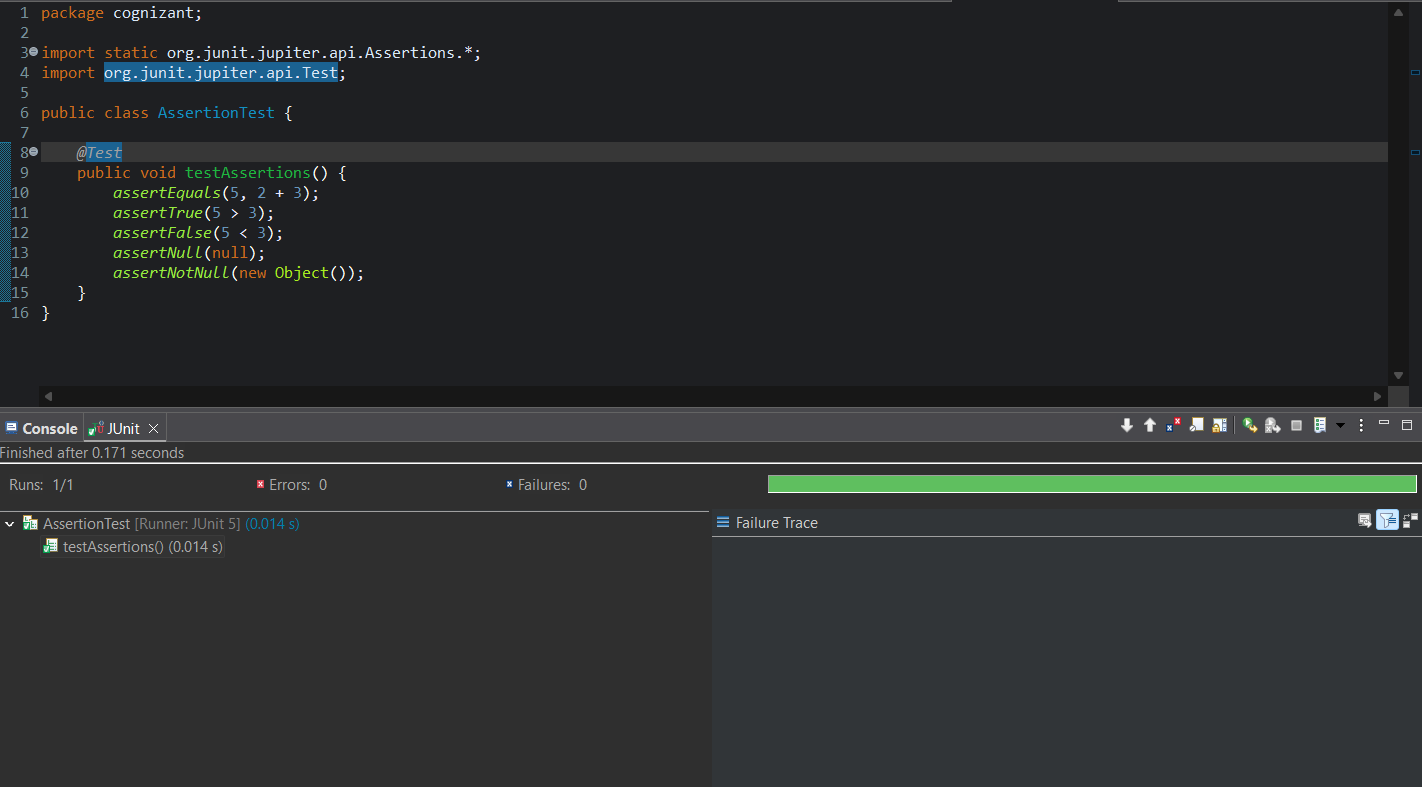
assertNull(null);

assertNotNull(new Object());

}

}

**Output:**



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

**Step 1. Create the class to test**

package com.cognizant;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

**Step 2:** **Write test using AAA pattern, setup, and teardown**

package com.cognizant;

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

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

public class CalculatorTest {

private Calculator;

// Setup method runs before each test

@BeforeEach

public void setUp() {

calculator = new Calculator();

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

}

// Teardown method runs after each test

@AfterEach

public void tearDown() {

calculator = null;

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

}

@Test

public void testAddition() {

// Arrange

int a = 5;

int b = 3;

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

assertEquals(8, result, "5 + 3 should equal 8");

}

@Test

public void testAdditionWithZero() {

int a = 0;

int b = 7;

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

assertEquals(7, result, "0 + 7 should equal 7");

}

}

Output:

