**Exercise 1: Setting Up Junit**

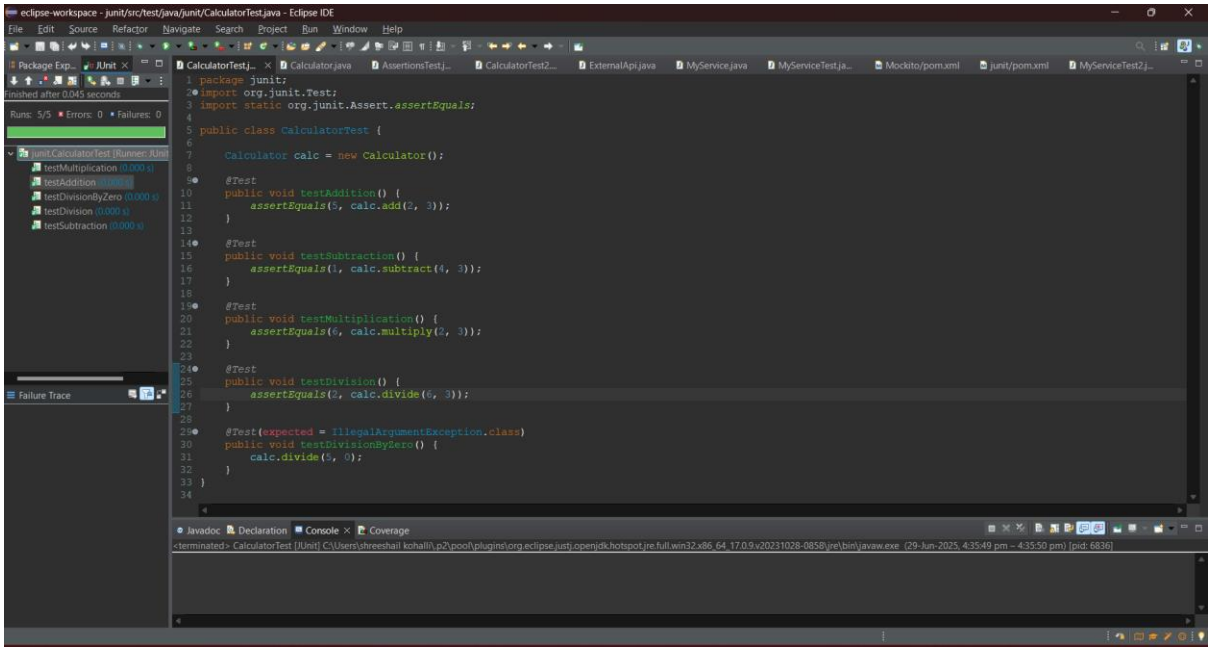
```
//main

package junit;

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("Cannot divide by zero.");
        }
        return a / b;
    }
}
```

```
//testobject

package junit;
import org.junit.Test;
import static org.junit.Assert.assertEquals;

public class CalculatorTest {

    Calculator calc = new Calculator();

    @Test
    public void testAddition() {
        assertEquals(5, calc.add(2, 3));
    }

    @Test
    public void testSubtraction() {
        assertEquals(1, calc.subtract(4, 3));
    }

    @Test
    public void testMultiplication() {
        assertEquals(6, calc.multiply(2, 3));
    }

    @Test
    public void testDivision() {
        assertEquals(2, calc.divide(6, 3));
    }

    @Test(expected = IllegalArgumentException.class)
    public void testDivisionByZero() {
        calc.divide(5, 0);
```

```
    }
}
```

```
//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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <dependencies>
        <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.13.2</version>
    <scope>test</scope>
</dependency>

  </dependencies>
</project>
```

Output:

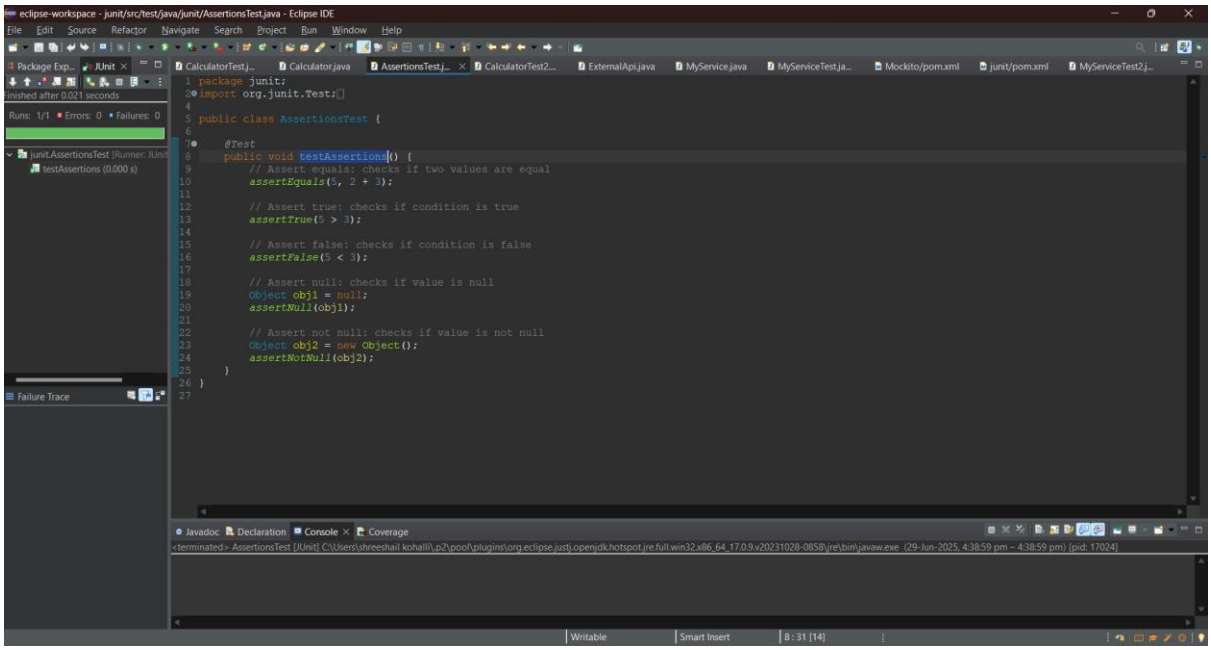


**Exercise 3: Assertions in Junit**

**//AssertionsTest.java**

```
package junit;
import org.junit.Test;
import static org.junit.Assert.*;

public class AssertionsTest {

    @Test
    public void testAssertions() {
        // Assert equals: checks if two values are equal
        assertEquals(5, 2 + 3);

        // Assert true: checks if condition is true
        assertTrue(5 > 3);

        // Assert false: checks if condition is false
        assertFalse(5 < 3);

        // Assert null: checks if value is null
        Object obj1 = null;
        assertNull(obj1);

        // Assert not null: checks if value is not null
        Object obj2 = new Object();
        assertNotNull(obj2);
    }
}
```

Output for assertion test:




**Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in Junit:**

**Test:**

```
package junit;


import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.assertEquals;


public class CalculatorTest2 {


    private Calculator calc;


    @Before

    public void setUp() {
```

```
    // Arrange: Initialize Calculator before each test
    calc = new Calculator();
    System.out.println("Setup: Calculator initialized");
}

@After
public void tearDown() {
    // Teardown: Runs after each test
    calc = null;
    System.out.println("Teardown: Calculator reset");
}

@Test
public void testAddition() {
    int result = calc.add(10, 5);
    assertEquals(15, result);
}

@Test
public void testSubtraction() {
    int result = calc.subtract(10, 5);
    assertEquals(5, result);
}

@Test
public void testMultiplication() {
    int result = calc.multiply(3, 4);
    assertEquals(12, result);
```

```
    }

    @Test
    public void testDivision() {
        int result = calc.divide(20, 4);
        assertEquals(5, result);
    }
}
```

Output: