**Exercise 1: Setting Up JUnit**

**Solution:**

calculator.java –

public class calculator {

    public int add(int a, int b) {

        return a + b;

    }

}

test1.java –

import org.junit.\*;

public class test1 {

    @Test

    public void testAdd() {

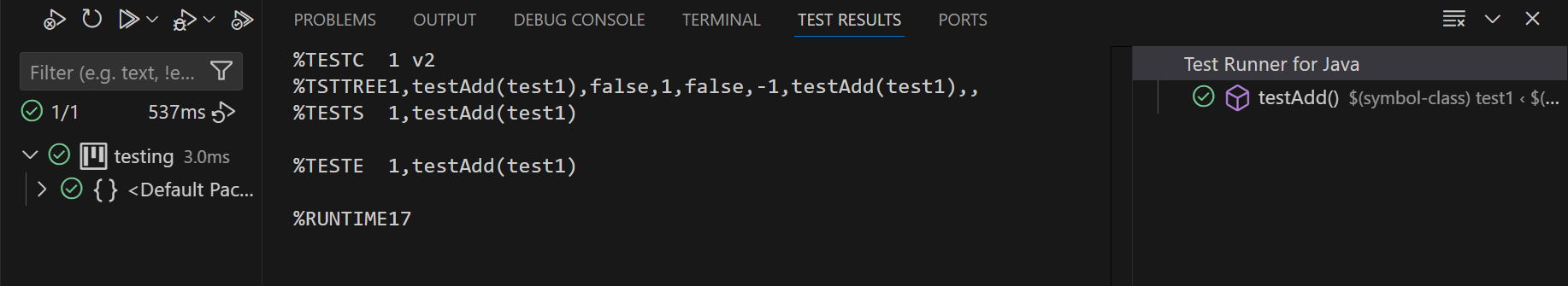
        calculator calc = new calculator();

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

    }

}

**Output :**

****

**Exercise 3: Assertions in JUnit**

**Solution:**

import org.junit.\*;

public class assertionstest {

    @Test

    public void testAssertions() {

        // Assert equals

        Assert.assertEquals(5, 2 + 3);

        // Assert true

        Assert.assertTrue(5 > 3);

        // Assert false

        Assert.assertFalse(5 < 3);

        // Assert null

        Assert.assertNull(null);

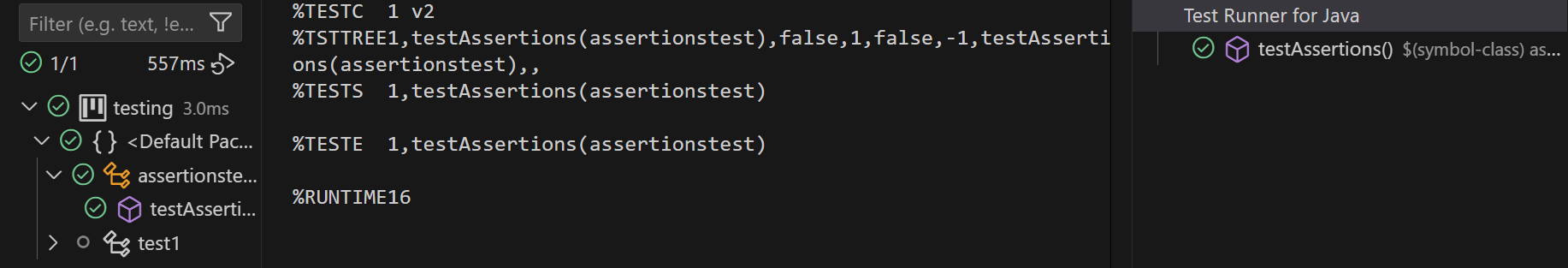
        // Assert not null

        Assert.assertNotNull(new Object());

    }

}

**Output:**

****

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

**Solution:**

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.AfterEach;

import org.junit.jupiter.api.Assertions;

import org.junit.jupiter.api.Test;

public class AAA {

    private Calculator calculator;

    @BeforeEach

    void setUp() {

        calculator = new Calculator(); // Arrange

    }

    @AfterEach

    void tearDown() {

        calculator = null;

    }

    @Test

    void testAddition() {

        // Act

        int result = calculator.add(2, 3);

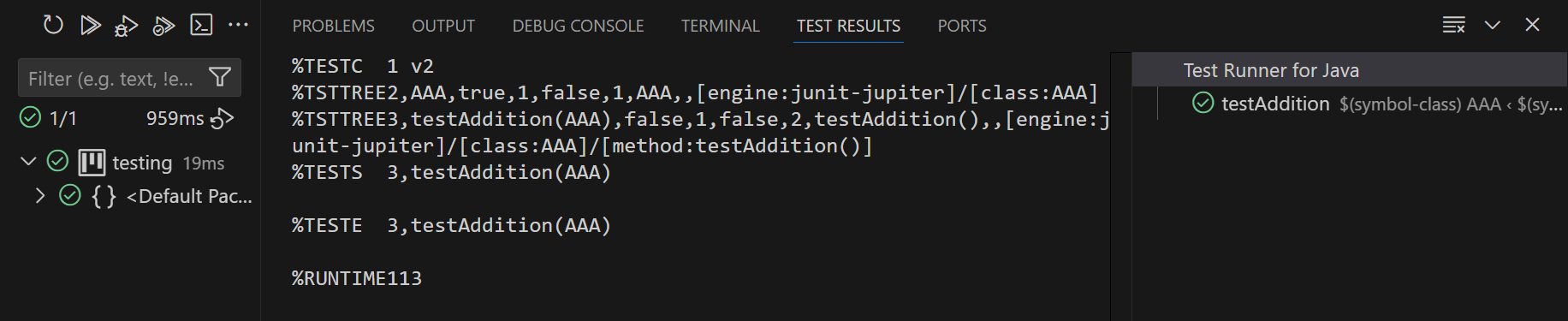
        // Assert

        Assertions.assertEquals(5, result);

    }

}

**Output:**

****