**JUNIT**

**Exercise 1: Setting Up JUnit**

**Pom.xml**

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

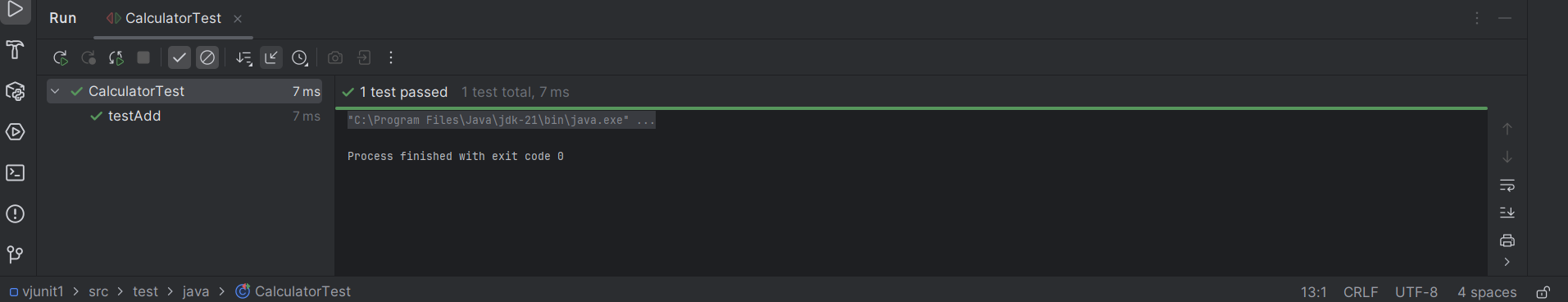
</dependencies>

**Calculator.java**

public class Calculator {  
 public int add(int a, int b) {  
 return a + b;  
 }  
}

**CalculatorTest.java**  
import org.junit.Test;  
import static org.junit.Assert.*assertEquals*;  
  
public class CalculatorTest {  
 @Test  
 public void testAdd() {  
 Calculator calculator = new Calculator();  
 int result = calculator.add(3, 4);  
 *assertEquals*(7, result);  
 }  
}

**Output:**

****

**Exercise 3: Assertions in JUnit**

**Pom.xml**

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

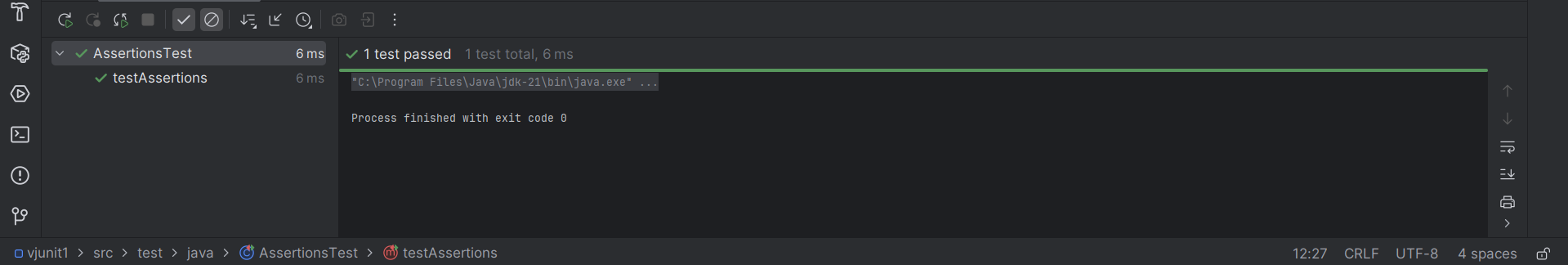
<scope>test</scope>

</dependency>

**AssertionsTest.java**

=import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class AssertionsTest {  
  
 @Test  
 public void testAssertions() {  
 // Check if the sum is correct  
 *assertEquals*(5, 2 + 3);  
  
 // Check if a condition is true  
 *assertTrue*(5 > 3);  
  
 // Check if a condition is false  
 *assertFalse*(5 < 3);  
  
 // Check if a value is null  
 *assertNull*(null);  
  
 // Check if an object is not null  
 *assertNotNull*(new Object());  
 }  
}

**Output:**

****

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

**Calculator.java**

public class Calculator {  
 public int add(int a, int b) {  
 return a + b;  
 }  
  
 public int subtract(int a, int b) {  
 return a - b;  
 }  
}

**CalculatorTest.java**

import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class CalculatorTest {  
  
 private Calculator calculator;  
  
 // Setup - runs before each test  
 @Before  
 public void setUp() {  
 calculator = new Calculator(); // Arrange  
 System.*out*.println("Setup completed");  
 }  
  
 // Teardown - runs after each test  
 @After  
 public void tearDown() {  
 calculator = null;  
 System.*out*.println("Teardown completed");  
 }  
  
 @Test  
 public void testAddition() {  
 // Arrange - already done in setUp()  
  
 // Act  
 int result = calculator.add(10, 5);  
  
 // Assert  
 *assertEquals*(15, result);  
 }  
  
 @Test  
 public void testSubtraction() {  
 // Act  
 int result = calculator.subtract(10, 5);  
  
 // Assert  
 *assertEquals*(5, result);  
 }  
}

**Output:**

