**Exercise 1: Setting Up Junit**

**Pom.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<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 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.example</groupId>  
 <artifactId>JUnitExample</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <properties>  
 <maven.compiler.source>24</maven.compiler.source>  
 <maven.compiler.target>24</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.13.2</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
</project>

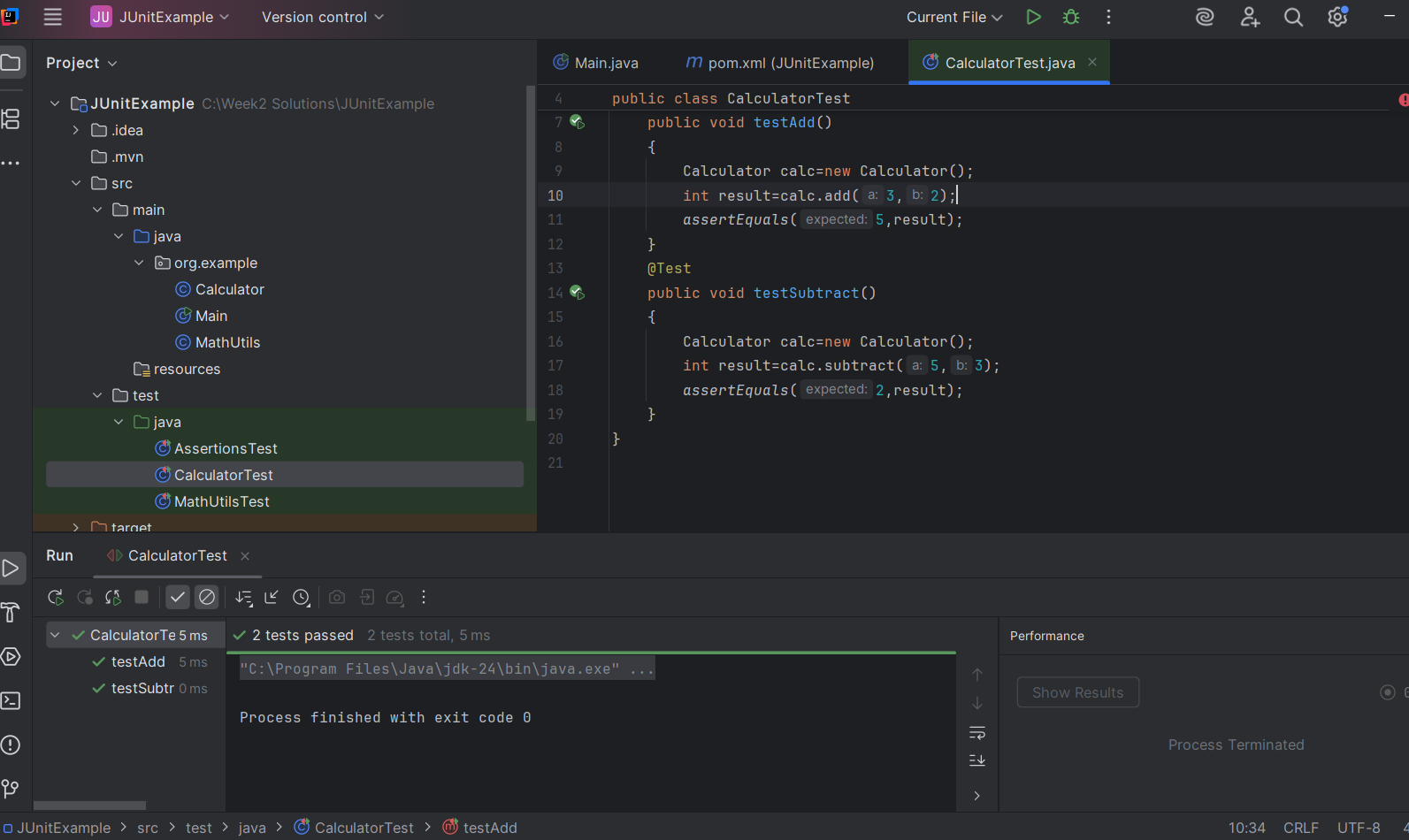
**Calculator.java**

package org.example;  
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**

package org.example;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
public class CalculatorTest  
{  
 @Test  
 public void testAdd()  
 {  
 Calculator calc=new Calculator();  
 int result=calc.add(3,2);  
 *assertEquals*(5,result);  
 }  
 @Test  
 public void testSubtract()  
 {  
 Calculator calc=new Calculator();  
 int result=calc.subtract(5,3);  
 *assertEquals*(2,result);  
 }  
}

**Output:**

****

**Exercise 2: Writing Basic JUnit Tests**

**MathUtils.java**

package org.example;  
public class MathUtils

{  
 public int square(int number)

{  
 return number\*number;  
 }  
 public boolean isEven(int number)

{  
 return number%2==0;  
 }  
 public int max(int a,int b)

{  
 return a>b?a:b;  
 }  
}

**MathUtilsTest.java**

package org.example;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
public class MathUtilsTest

{  
 @Test  
 public void testSquare()

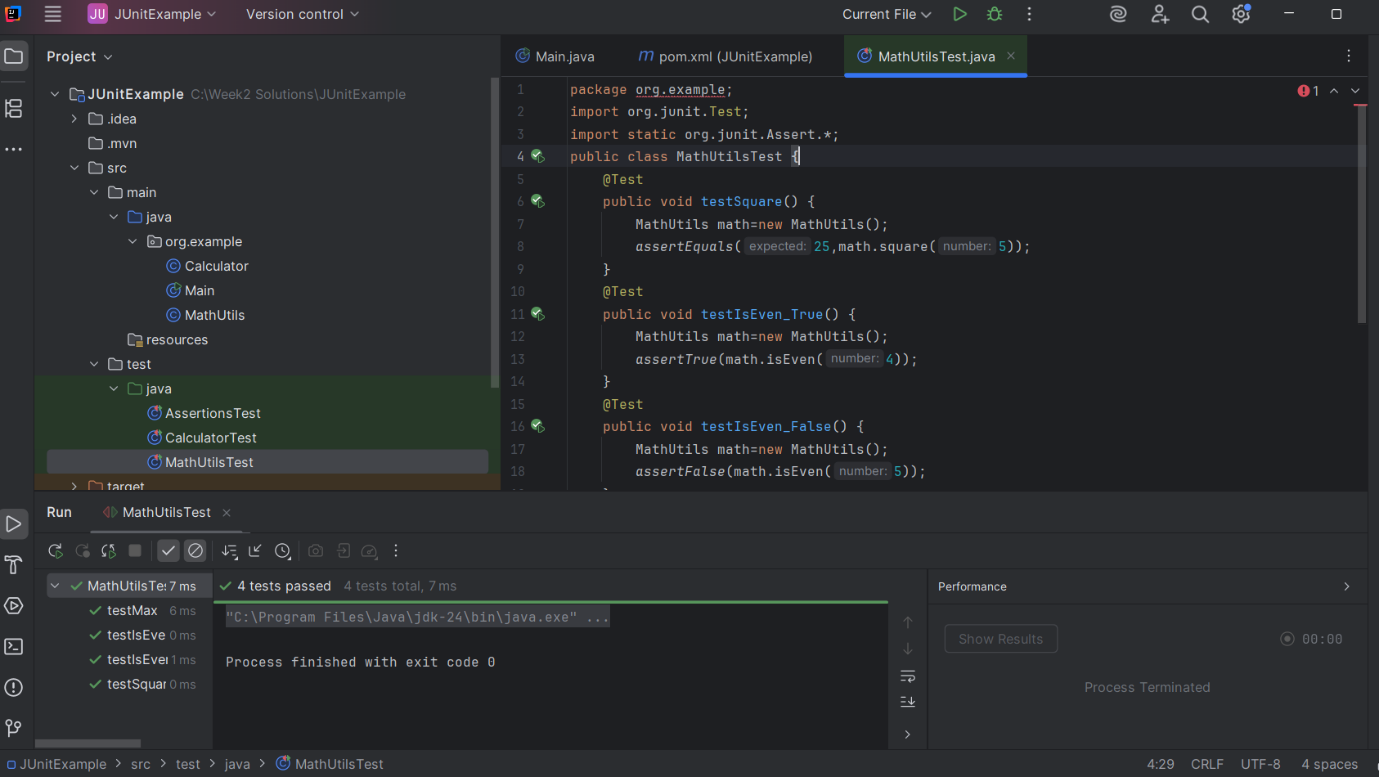
{  
 MathUtils math=new MathUtils();  
 *assertEquals*(25,math.square(5));  
 }  
 @Test  
 public void testIsEven\_True()

{  
 MathUtils math=new MathUtils();  
 *assertTrue*(math.isEven(4));  
 }  
 @Test  
 public void testIsEven\_False()

{  
 MathUtils math=new MathUtils();  
 *assertFalse*(math.isEven(5));  
 }  
 @Test  
 public void testMax()

{  
 MathUtils math=new MathUtils();  
 *assertEquals*(10,math.max(10,3));  
 *assertEquals*(20,math.max(15,20));  
 }  
}

**Output**

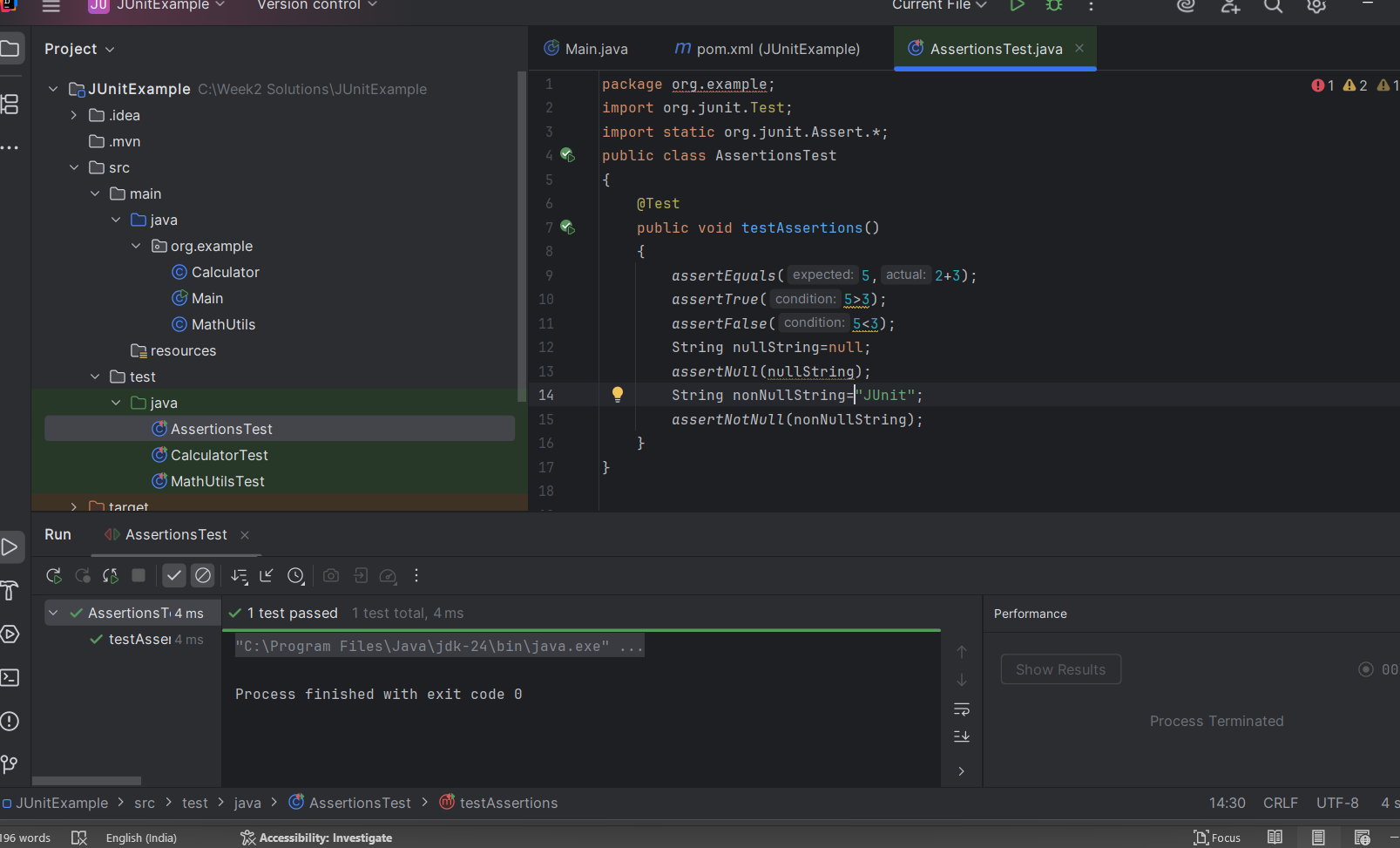
****

**Exercise 3: Assertions in Junit**

**AssertionsTest.java**

package org.example;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
public class AssertionsTest  
{  
 @Test  
 public void testAssertions()  
 {  
 *assertEquals*(5,2+3);  
 *assertTrue*(5>3);  
 *assertFalse*(5<3);  
 String nullString=null;  
 *assertNull*(nullString);  
 String nonNullString="JUnit";  
 *assertNotNull*(nonNullString);  
 }  
}

**Output**

****

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

**Calculator.java**

package org.example;  
public class Calculator  
{  
 public int add(int a,int b)  
 {  
 return a+b;  
 }  
 public int multiply(int a,int b)  
 {  
 return a\*b;  
 }  
}

**CalculatorTest.java**

package org.example;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
public class CalculatorTest  
{  
 private Calculator calculator;  
 @Before  
 public void setUp()  
 {  
 calculator=new Calculator();  
 System.*out*.println("Setting up...");  
 }  
 @After  
 public void tearDown()  
 {  
 calculator=null;  
 System.*out*.println("Cleaning up...");  
 }  
 @Test  
 public void testAdd() {  
 int result=calculator.add(2,3);  
 *assertEquals*(5, result);  
 }  
 @Test  
 public void testMultiply() {  
 int result=calculator.multiply(4,5);  
 *assertEquals*(20, result);  
 }  
}

**Output**

