**Exercise 1: Parameterized Tests**

**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>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-params</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
</project>

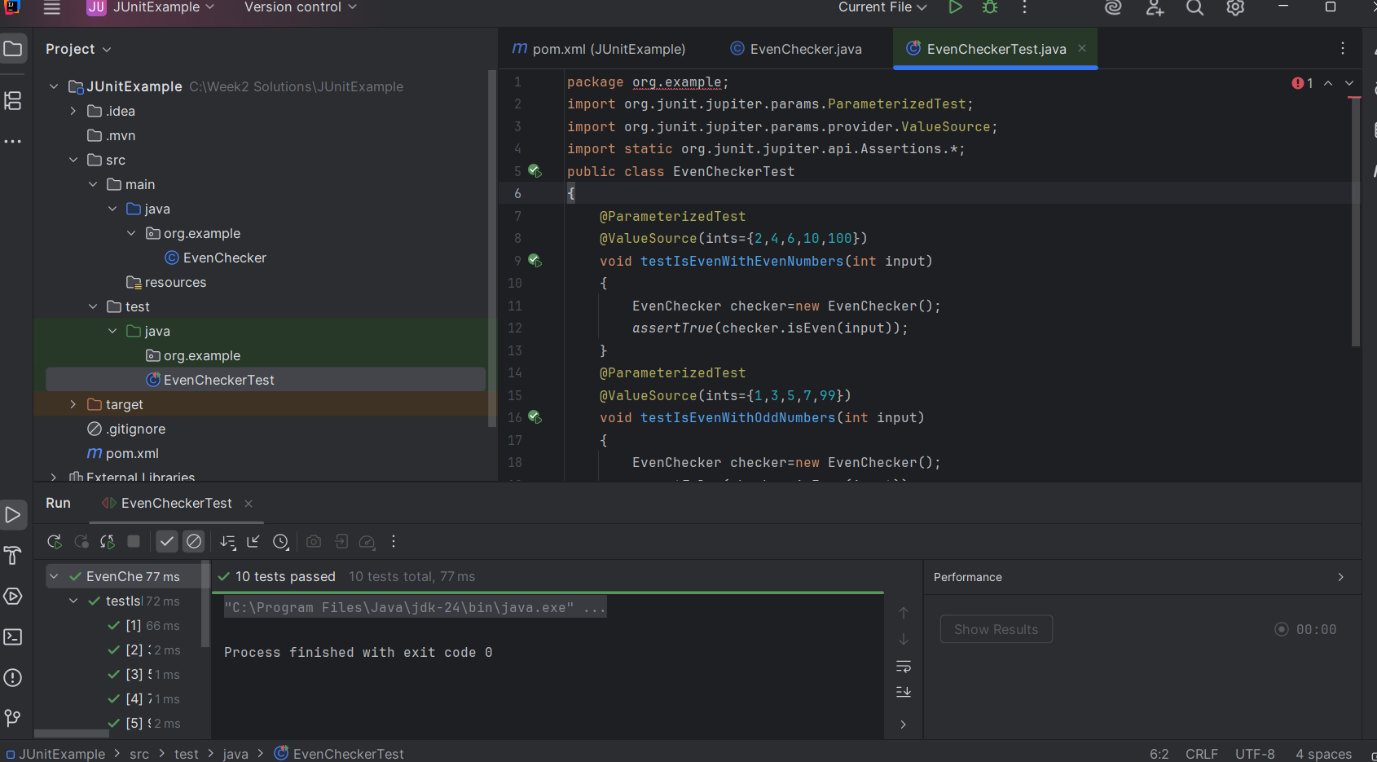
**EvenChecker.java**

package org.example;  
public class EvenChecker  
{  
 public boolean isEven(int number)  
 {  
 return number%2==0;  
 }  
}

**EvenCheckerTest.java**

package org.example;  
import org.junit.jupiter.params.ParameterizedTest;  
import org.junit.jupiter.params.provider.ValueSource;  
import static org.junit.jupiter.api.Assertions.\*;  
public class EvenCheckerTest  
{  
 @ParameterizedTest  
 @ValueSource(ints={2,4,6,10,100})  
 void testIsEvenWithEvenNumbers(int input)  
 {  
 EvenChecker checker=new EvenChecker();  
 *assertTrue*(checker.isEven(input));  
 }  
 @ParameterizedTest  
 @ValueSource(ints={1,3,5,7,99})  
 void testIsEvenWithOddNumbers(int input)  
 {  
 EvenChecker checker=new EvenChecker();  
 *assertFalse*(checker.isEven(input));  
 }  
}

**Output**

****

**Exercise 2: Test Suites and Categories**

**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>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.platform</groupId>  
 <artifactId>junit-platform-suite</artifactId>  
 <version>1.10.0</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;  
 }  
}

**MathUtils.java**

package org.example;  
public class MathUtils {  
 public int square(int number) {  
 return number\*number;  
 }  
 public int multiply(int a,int b) {  
 return a\*b;  
 }  
}

**CalculatorTest.java**

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
public class CalculatorTest  
{  
 Calculator calculator=new Calculator();  
 @Test  
 void testAdd() {  
 *assertEquals*(5,calculator.add(2,3));  
 }  
 @Test  
 void testSubtract() {  
 *assertEquals*(1,calculator.subtract(3,2));  
 }  
}

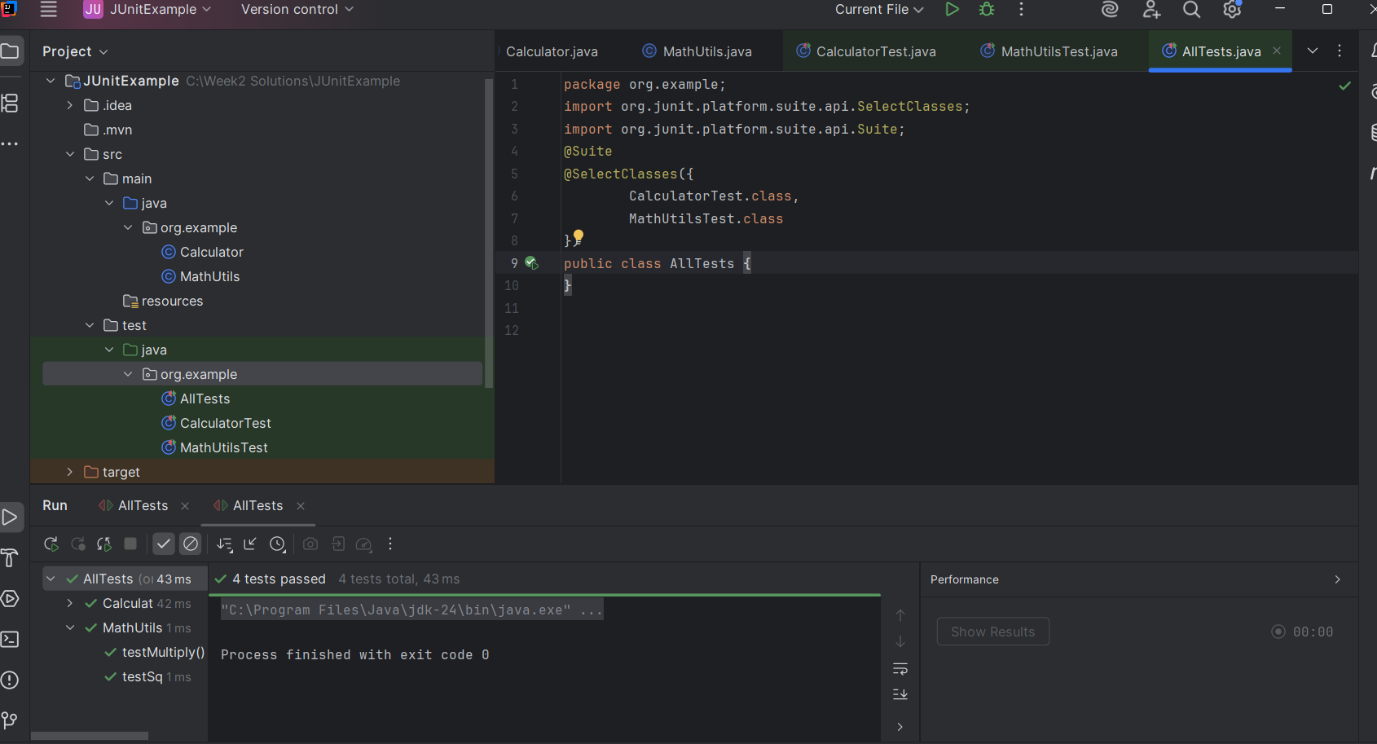
**MathUtilsTest.java**

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
public class MathUtilsTest {  
 MathUtils mathUtils=new MathUtils();  
 @Test  
 void testSquare() {  
 *assertEquals*(9,mathUtils.square(3));  
 }  
 @Test  
 void testMultiply() {  
 *assertEquals*(12,mathUtils.multiply(3,4));  
 }  
}

**AllTests.java**

package org.example;  
import org.junit.platform.suite.api.SelectClasses;  
import org.junit.platform.suite.api.Suite;  
@Suite  
@SelectClasses({  
 CalculatorTest.class,  
 MathUtilsTest.class  
})  
public class AllTests {  
}

**Output**

****

**Exercise 3: Test Execution Order**

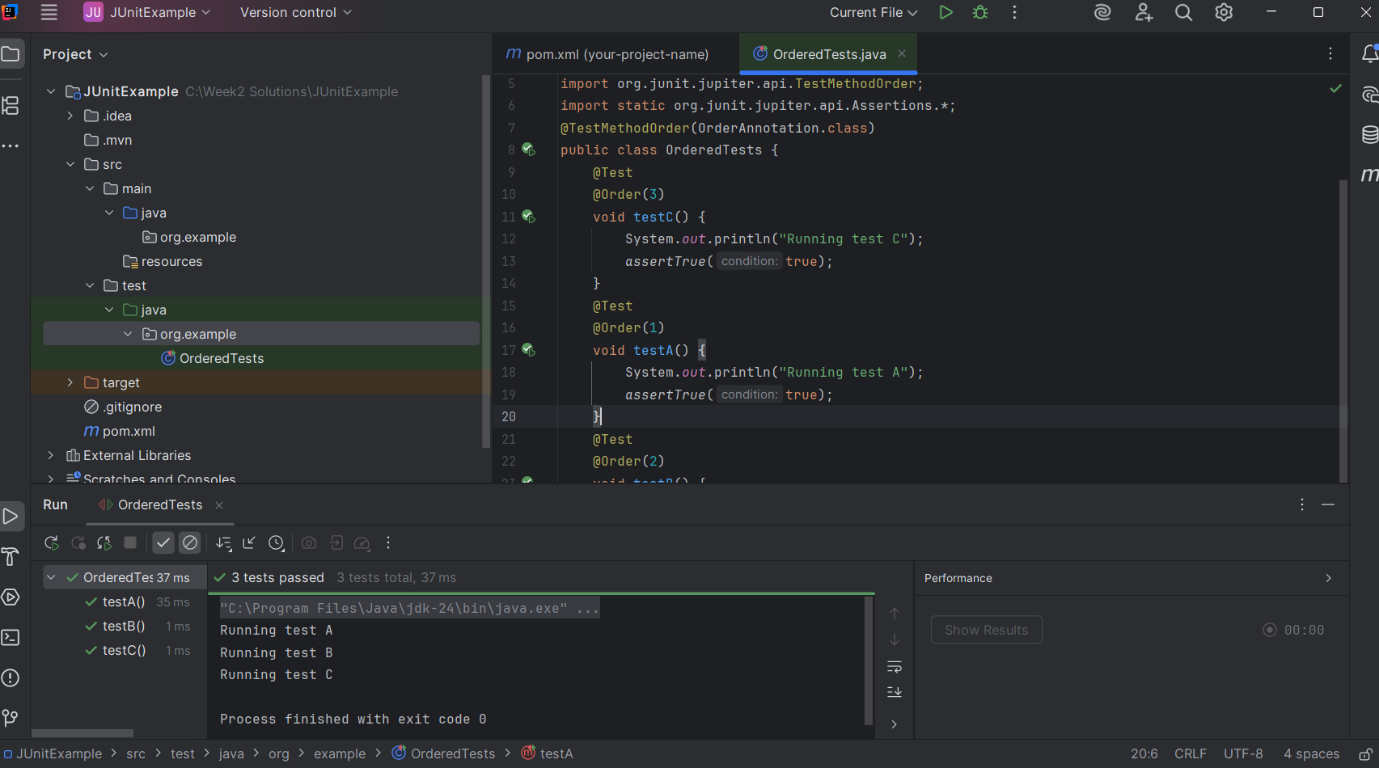
**Pom.xml**

<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  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.example</groupId>  
 <artifactId>your-project-name</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <dependencies>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.platform</groupId>  
 <artifactId>junit-platform-suite</artifactId>  
 <version>1.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.0.0</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**OrderedTests.java**

package org.example;  
import org.junit.jupiter.api.MethodOrderer.OrderAnnotation;  
import org.junit.jupiter.api.Order;  
import org.junit.jupiter.api.Test;  
import org.junit.jupiter.api.TestMethodOrder;  
import static org.junit.jupiter.api.Assertions.\*;  
@TestMethodOrder(OrderAnnotation.class)  
public class OrderedTests {  
 @Test  
 @Order(3)  
 void testC() {  
 System.*out*.println("Running test C");  
 *assertTrue*(true);  
 }  
 @Test  
 @Order(1)  
 void testA() {  
 System.*out*.println("Running test A");  
 *assertTrue*(true);  
 }  
 @Test  
 @Order(2)  
 void testB() {  
 System.*out*.println("Running test B");  
 *assertTrue*(true);  
 }  
}

**Output**

****

**Exercise 4: Exception Testing**

**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  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.example</groupId>  
 <artifactId>junit-demo</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <properties>  
 <maven.compiler.source>17</maven.compiler.source>  
 <maven.compiler.target>17</maven.compiler.target>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.0.0</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>

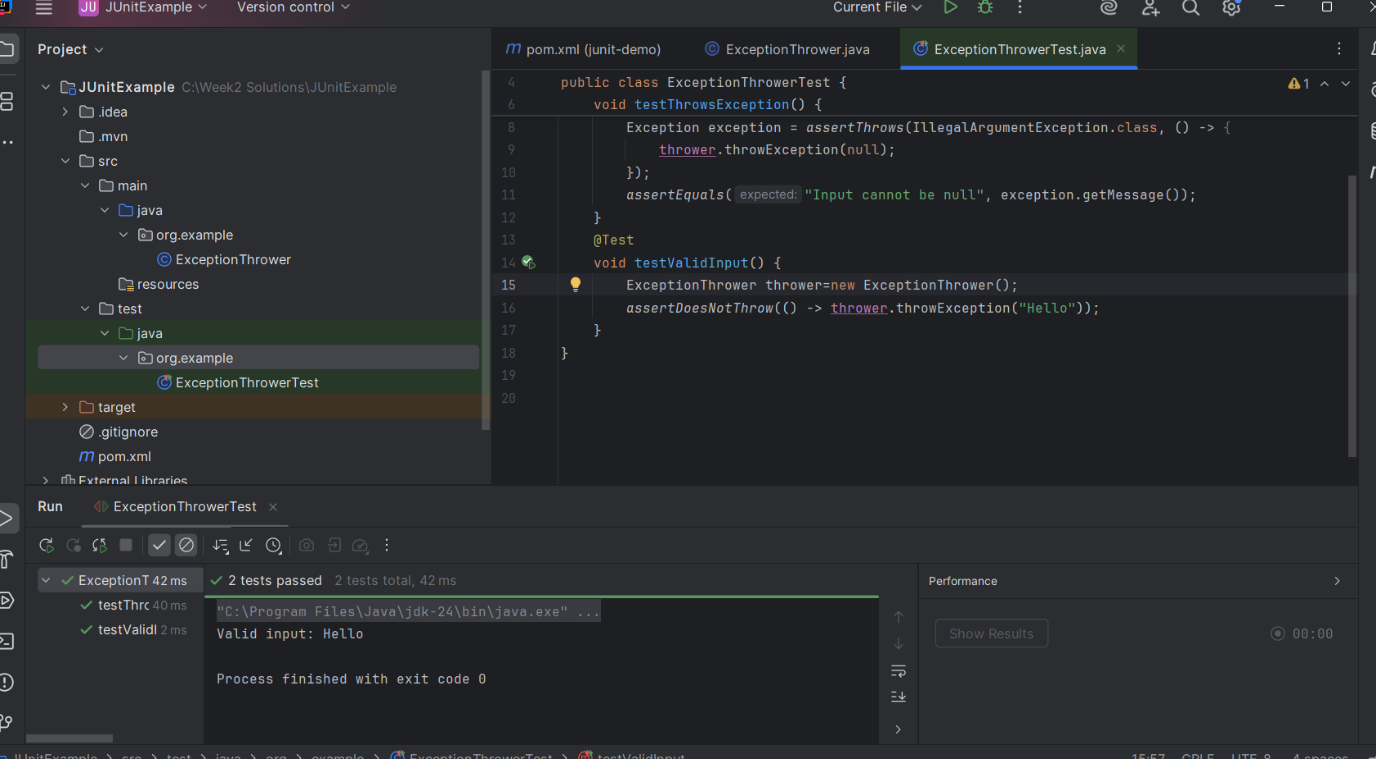
**ExceptionThrower.java**

package org.example;  
public class ExceptionThrower {  
 public void throwException(String input) {  
 if (input==null) {  
 throw new IllegalArgumentException("Input cannot be null");  
 }  
 System.*out*.println("Valid input: " + input);  
 }  
}

**ExceptionThrowerTest.java**

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
public class ExceptionThrowerTest {  
 @Test  
 void testThrowsException() {  
 ExceptionThrower thrower=new ExceptionThrower();  
 Exception exception = *assertThrows*(IllegalArgumentException.class, () -> {  
 thrower.throwException(null);  
 });  
 *assertEquals*("Input cannot be null", exception.getMessage());  
 }  
 @Test  
 void testValidInput() {  
 ExceptionThrower thrower=new ExceptionThrower();  
 *assertDoesNotThrow*(() -> thrower.throwException("Hello"));  
 }  
}

**Output**

****

**Exercise 5: Timeout and Performance Testing**

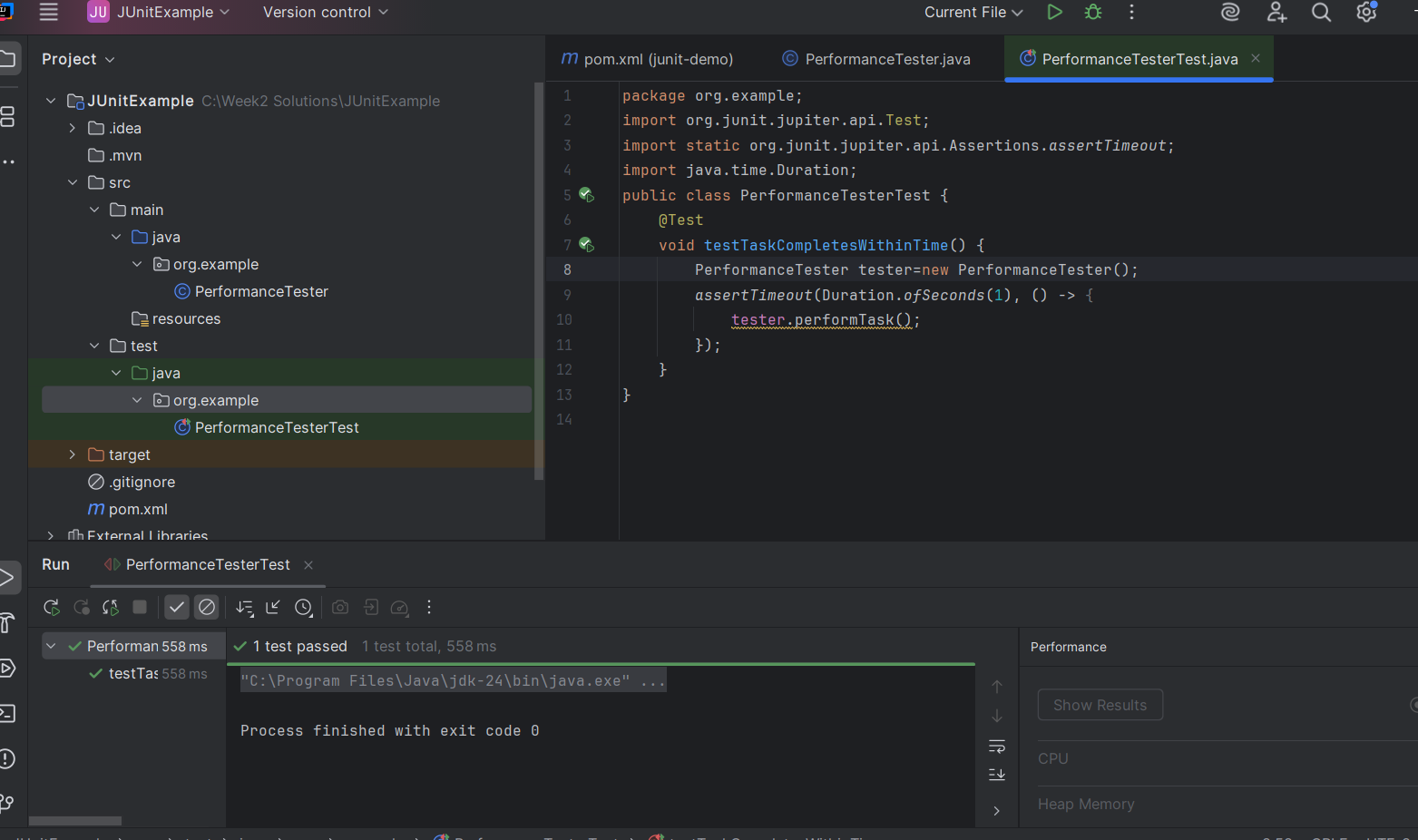
**PerformanceTester.java**

package org.example;  
public class PerformanceTester {  
 public void performTask() {  
 try {  
 Thread.*sleep*(500);  
 } catch (InterruptedException e) {  
 Thread.*currentThread*().interrupt();  
 }  
 }  
}

**PerformanceTesterTest.java**

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.*assertTimeout*;  
import java.time.Duration;  
public class PerformanceTesterTest {  
 @Test  
 void testTaskCompletesWithinTime() {  
 PerformanceTester tester=new PerformanceTester();  
 *assertTimeout*(Duration.*ofSeconds*(1), () -> {  
 tester.performTask();  
 });  
 }  
}

**Output**

****