**Week2: JUnit Advanced Testing Exercises**

**Exercise 1: Parameterized Tests**

**CODE :**

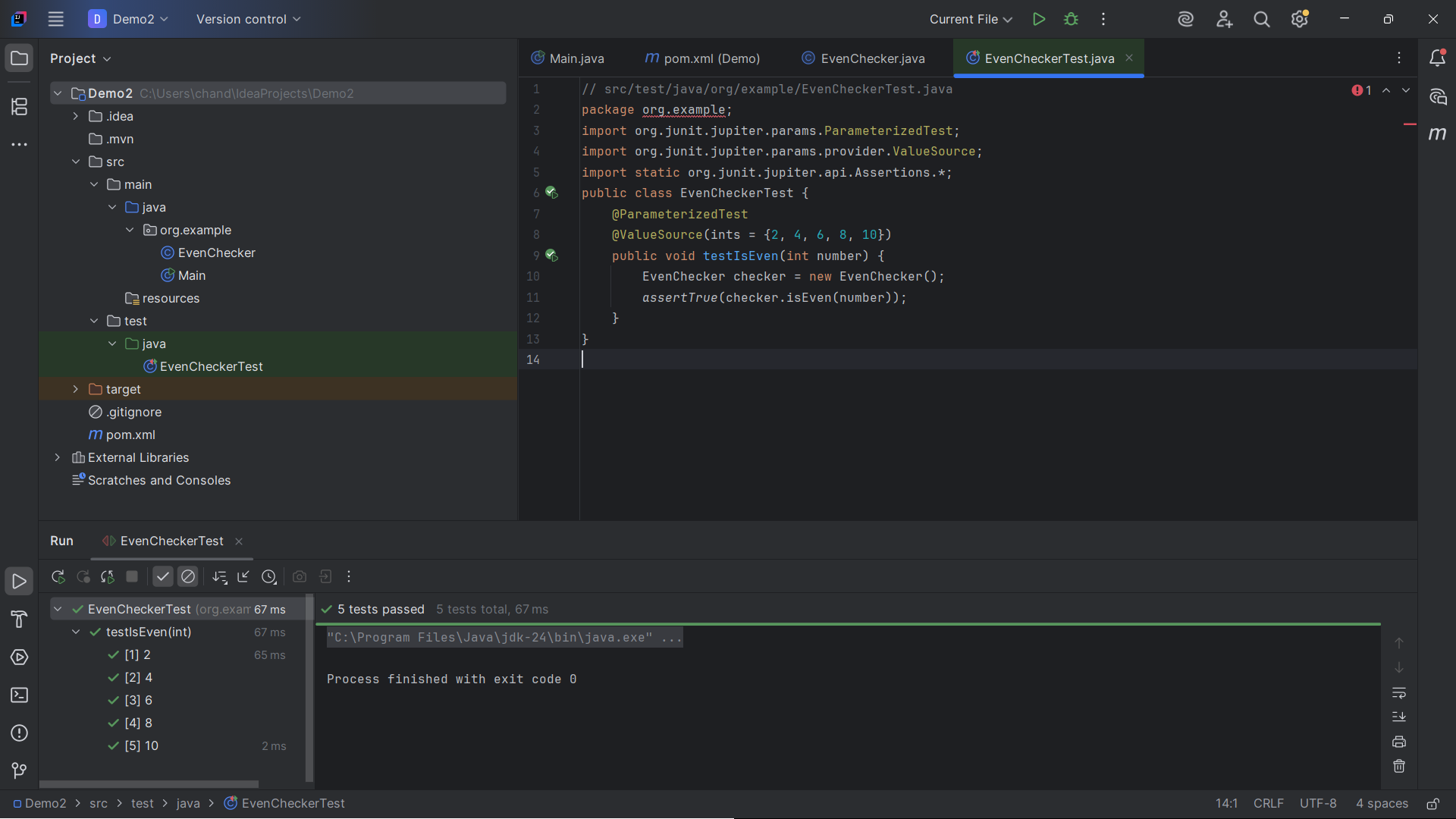
**EvenChecker.java**

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, 8, 10})  
 public void testIsEven(int number) {  
 EvenChecker checker = new EvenChecker();  
 *assertTrue*(checker.isEven(number));  
 }  
}

**OUTPUT:**



**Exercise 2: Test Suites and Categories**

**CODE :**

import org.junit.platform.suite.api.SelectClasses;

import org.junit.platform.suite.api.Suite;

@Suite

@SelectClasses({EvenCheckerTest.class, OrderedTests.class, ExceptionThrowerTest.class})

public class AllTests {

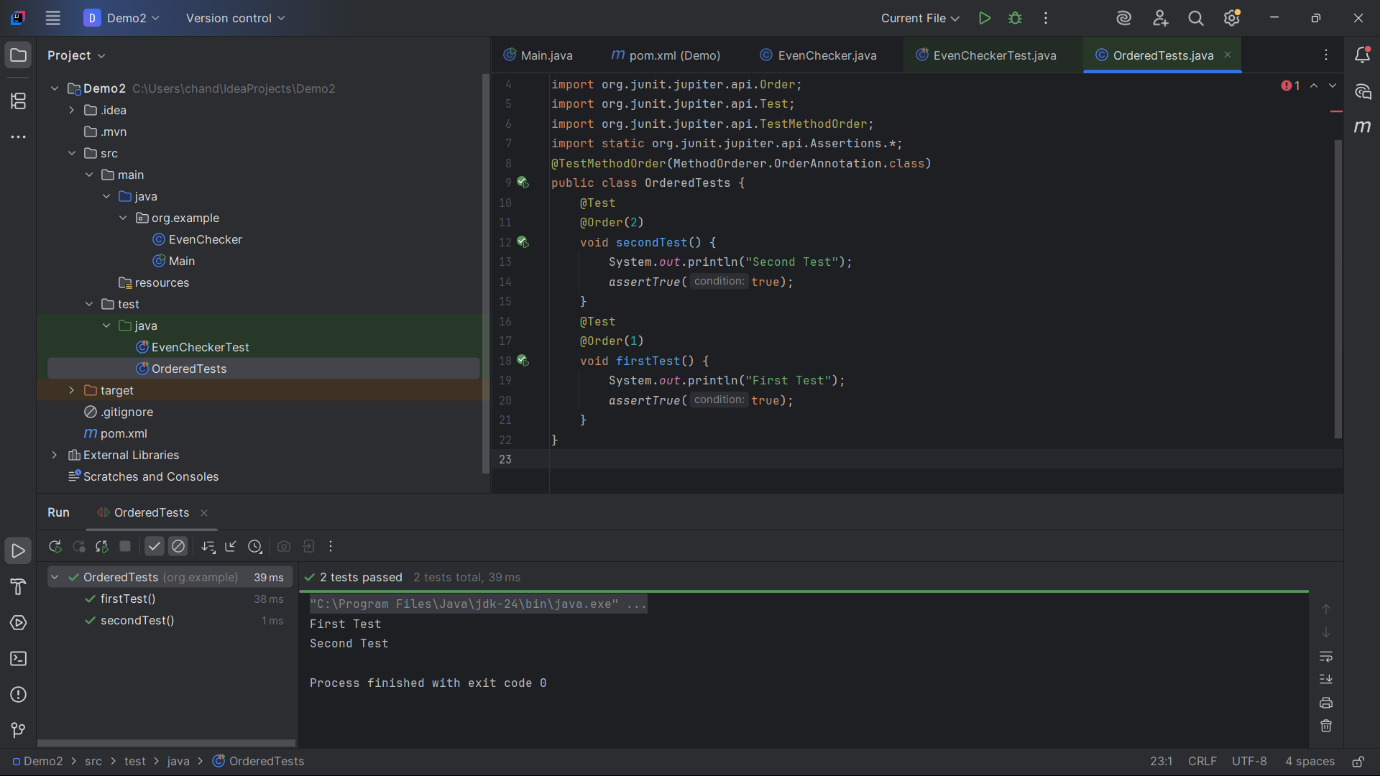
}

**Exercise 3: Test Execution Order**

**CODE :**

**OrderedTests.java**  
package org.example;  
import org.junit.jupiter.api.MethodOrderer;  
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(MethodOrderer.OrderAnnotation.class)  
public class OrderedTests {  
 @Test  
 @Order(2)  
 void secondTest() {  
 System.*out*.println("Second Test");  
 *assertTrue*(true);  
 }  
 @Test  
 @Order(1)  
 void firstTest() {  
 System.*out*.println("First Test");  
 *assertTrue*(true);  
 }  
}

**OUTPUT:**



**Exercise 4: Exception Testing**

**CODE :**

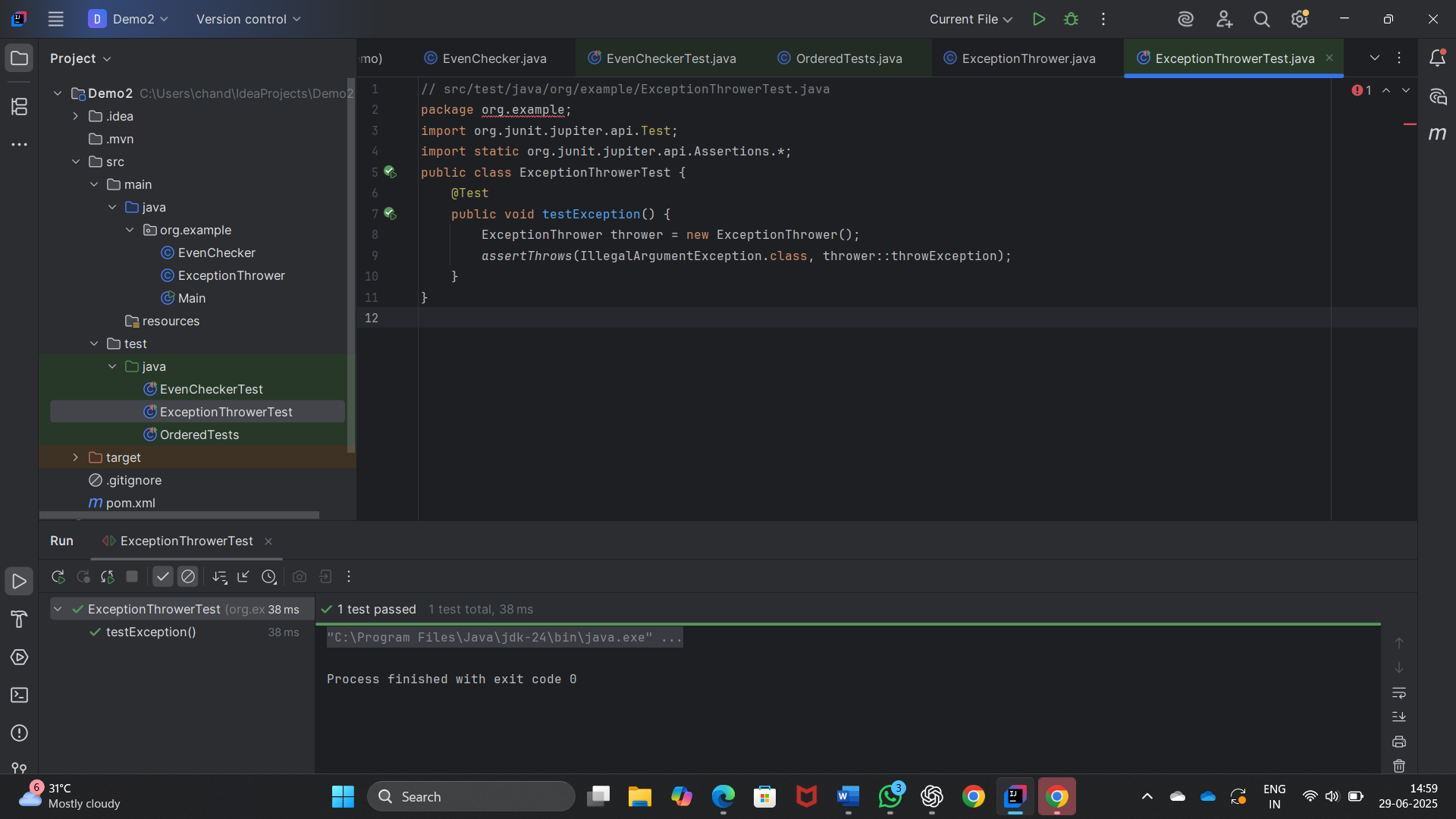
**ExceptionThrower.java**

package org.example;  
public class ExceptionThrower {  
 public void throwException() {  
 throw new IllegalArgumentException("Invalid value!");  
 }  
}

**ExceptionThrowerTest.java**

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
public class ExceptionThrowerTest {  
 @Test  
 public void testException() {  
 ExceptionThrower thrower = new ExceptionThrower();  
 *assertThrows*(IllegalArgumentException.class, thrower::throwException);  
 }  
}

**OUTPUT:**



**Exercise 5: Timeout and Performance Testing**

**CODE :**

**PerformaceTester.java**

package org.example;  
public class PerformanceTester {  
 public void performTask() throws InterruptedException {  
 Thread.*sleep*(200); // simulate work for 200 ms  
 }  
}

**PerformaceTesterTest.java**

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
import java.time.Duration;  
public class PerformanceTesterTest {  
 @Test  
 public void testPerformance() {  
 PerformanceTester tester = new PerformanceTester();  
 *assertTimeout*(Duration.*ofMillis*(500), () -> {  
 tester.performTask();  
 });  
 }  
}

**OUTPUT:**

