**Week2: JUnit Advanced Testing Exercises**

**Exercise 1: Parameterized Tests**

**CODE :**

**EvenChecker.java**

public class EvenChecker {

public boolean isEven(int number) {

return number % 2 == 0;

}

}

**EvenCheckerTest.java**

import org.junit.jupiter.params.ParameterizedTest;

import org.junit.jupiter.params.provider.ValueSource;

import static org.junit.jupiter.api.Assertions.\*;

public class EvenCheckerTest {

EvenChecker checker = new EvenChecker();

@ParameterizedTest

@ValueSource(ints = {2, 4, 6, 8, 10})

void testIsEvenTrue(int number) {

assertTrue(checker.isEven(number));

}

@ParameterizedTest

@ValueSource(ints = {1, 3, 5, 7, 9})

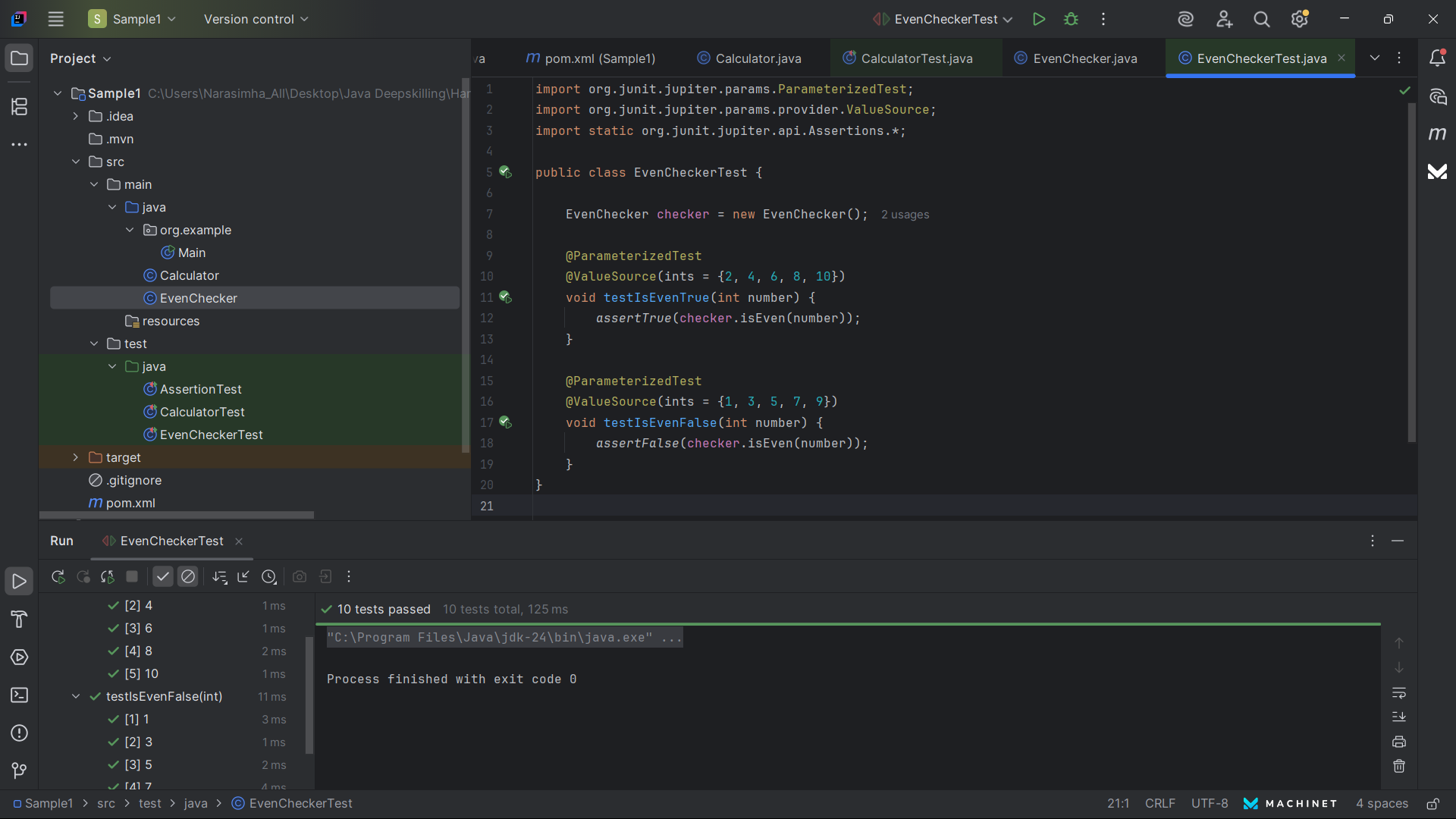
void testIsEvenFalse(int number) {

assertFalse(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**

import org.junit.jupiter.api.\*;

@TestMethodOrder(MethodOrderer.OrderAnnotation.class)

public class OrderedTests {

@Test

@Order(2)

void testSecond() {

System.out.println("Second test");

}

@Test

@Order(1)

void testFirst() {

System.out.println("First test");

}

@Test

@Order(3)

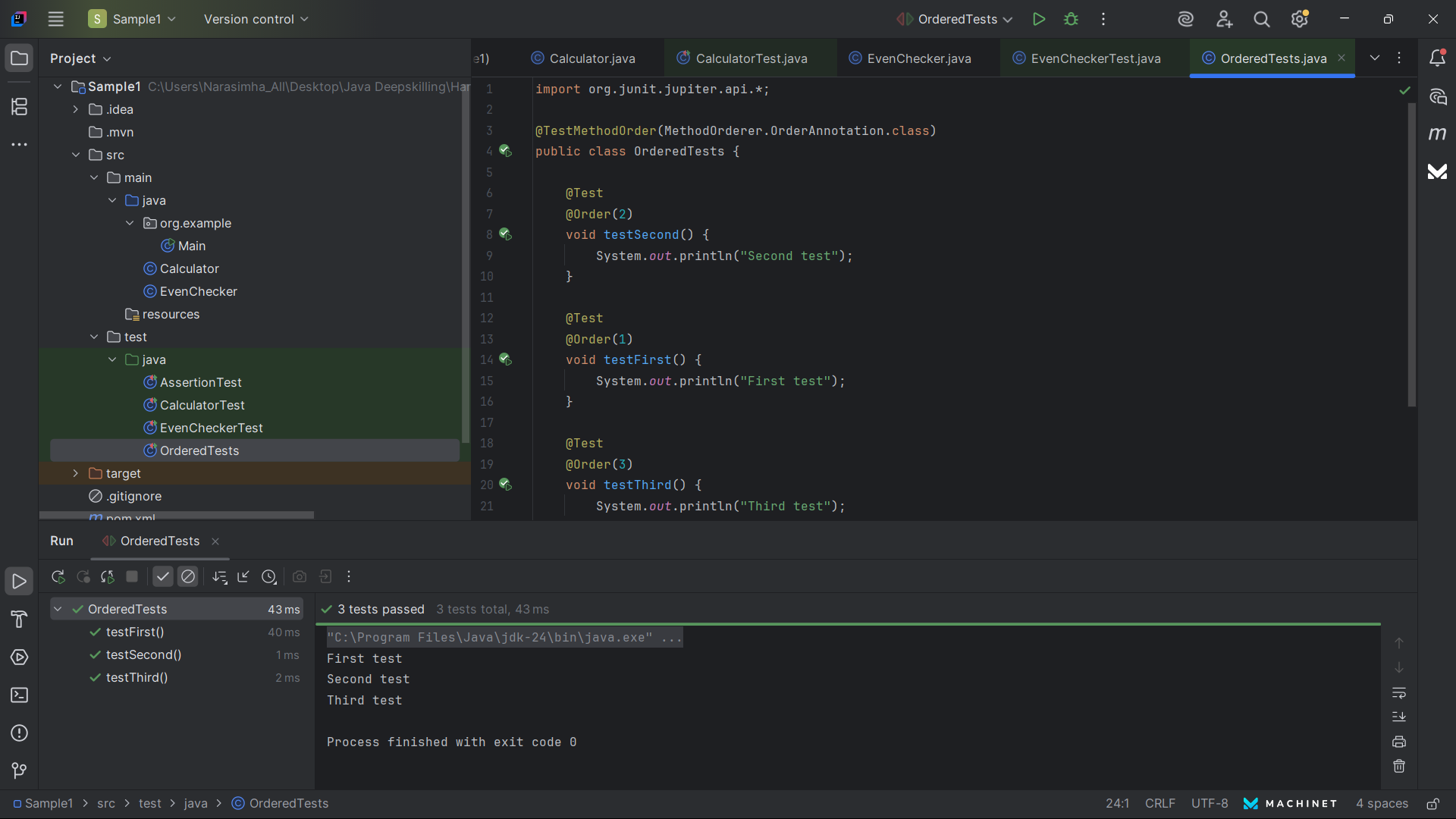
void testThird() {

System.out.println("Third test");

}

}

**Output :**



**Exercise 4: Exception Testing**

**CODE :**

**ExceptionThrower.java**

public class ExceptionThrower {

public void throwException() {

throw new IllegalArgumentException("This is an exception");

}

}

**ExceptionThrowerTest.java**

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.\*;

public class ExceptionThrowerTest {

@Test

void testExceptionThrown() {

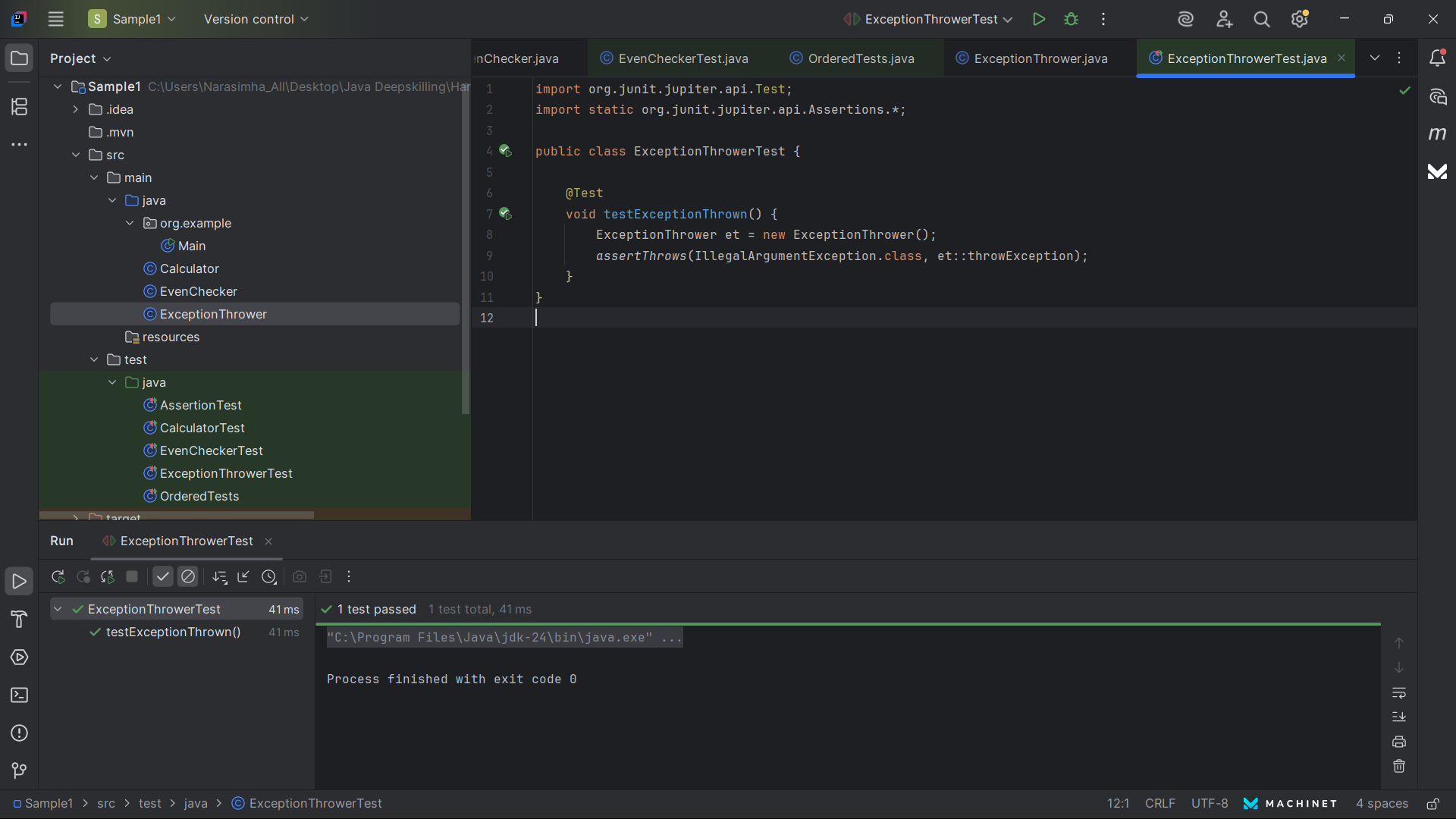
ExceptionThrower et = new ExceptionThrower();

assertThrows(IllegalArgumentException.class, et::throwException);

}

}

**Output :**



**Exercise 5: Timeout and Performance Testing**

**CODE :**

**PerformaceTester.java**

public class PerformanceTester {

public void performTask() {

try {

Thread.sleep(500); // Simulate some delay

} catch (InterruptedException e) {

Thread.currentThread().interrupt();

}

}

}

**PerformaceTesterTest.java**

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.\*;

import java.time.Duration;

public class PerformanceTesterTest {

@Test

void testPerformance() {

PerformanceTester pt = new PerformanceTester();

assertTimeout(Duration.ofSeconds(1), pt::performTask);

}

}

**Output :**

