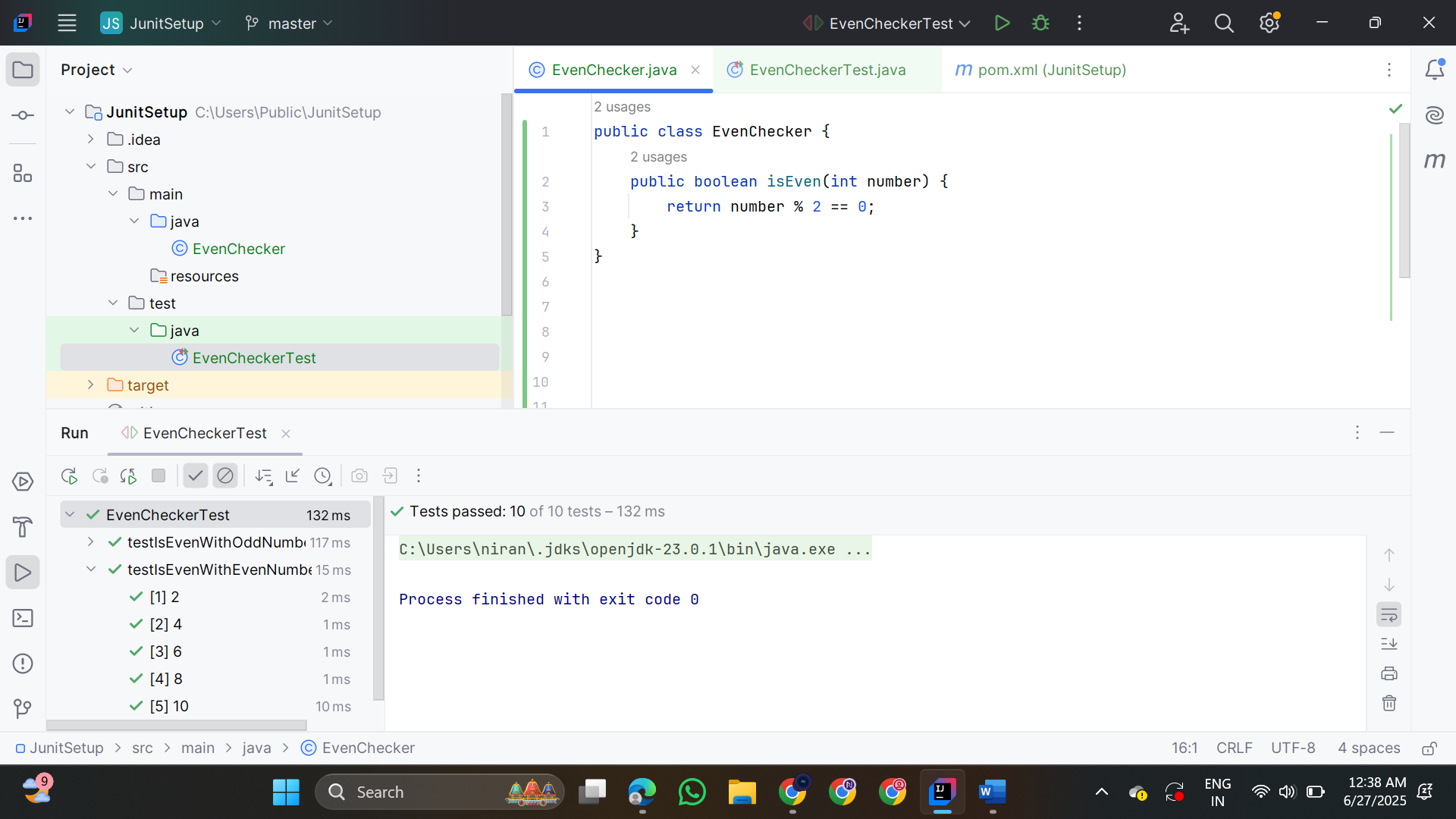
**Junit Advanced Testing  
  
 1.Parameterized Tests**

**EvenChecker.java**public class EvenChecker {  
 public boolean isEven(int number) {  
 return number % 2 == 0;  
 }  
}

**EvenChecker.java**

import org.junit.jupiter.params.ParameterizedTest;  
import org.junit.jupiter.params.provider.ValueSource;  
import static org.junit.jupiter.api.Assertions.assertTrue;  
import static org.junit.jupiter.api.Assertions.assertFalse;  
  
public class EvenCheckerTest {  
  
 EvenChecker checker = new EvenChecker();  
  
 @ParameterizedTest  
 @ValueSource(ints = {2, 4, 6, 8, 10})  
 void testIsEvenWithEvenNumbers(int number) {  
 assertTrue(checker.isEven(number));  
 }  
  
 @ParameterizedTest  
 @ValueSource(ints = {1, 3, 5, 7, 9})  
 void testIsEvenWithOddNumbers(int number)

{  
 assertFalse(checker.isEven(number));  
 }  
}



**2.Testsuites and Categories**

**AllTests.java**

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

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

@Suite

@SelectClasses({

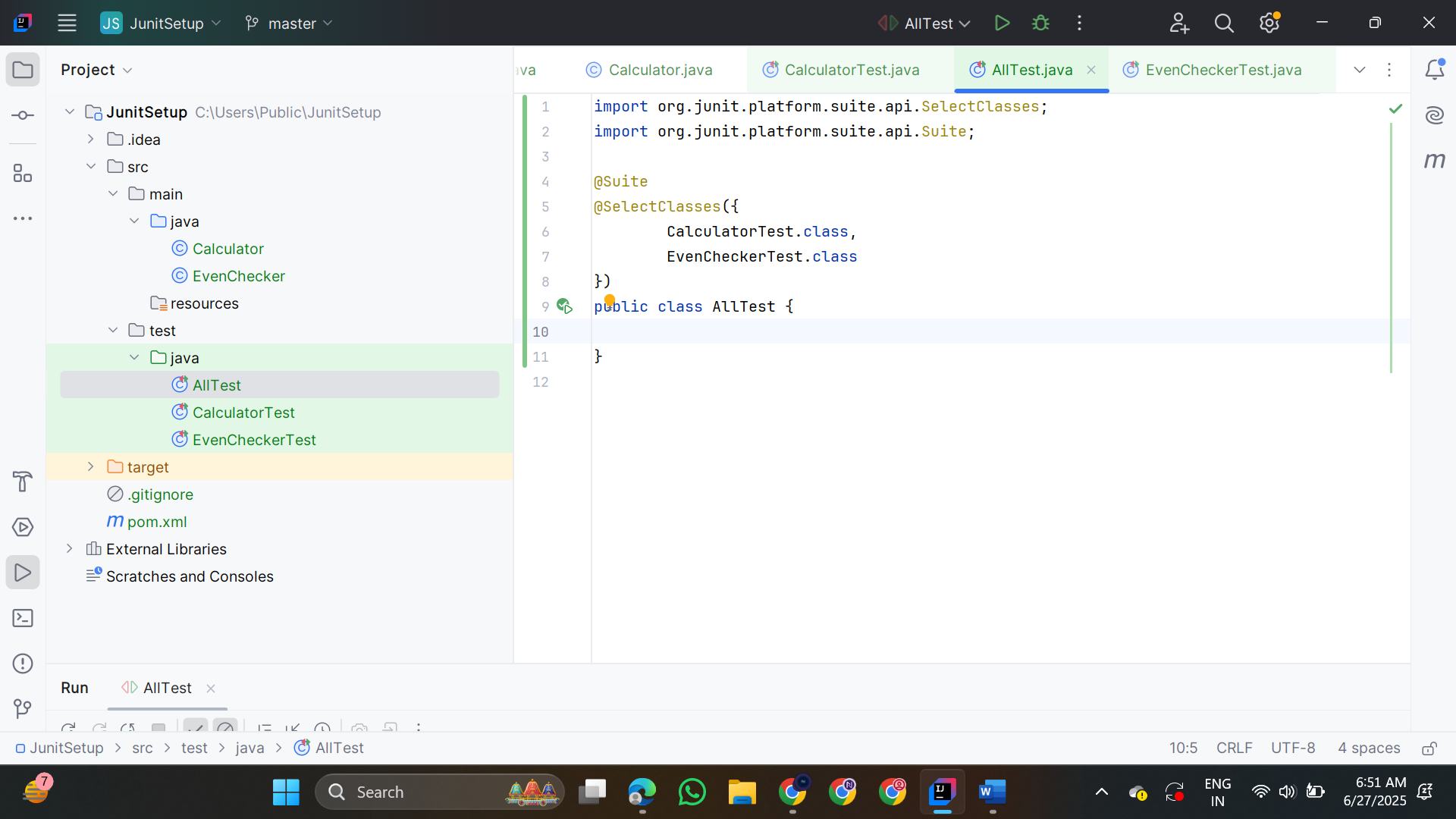
CalculatorTest.class,

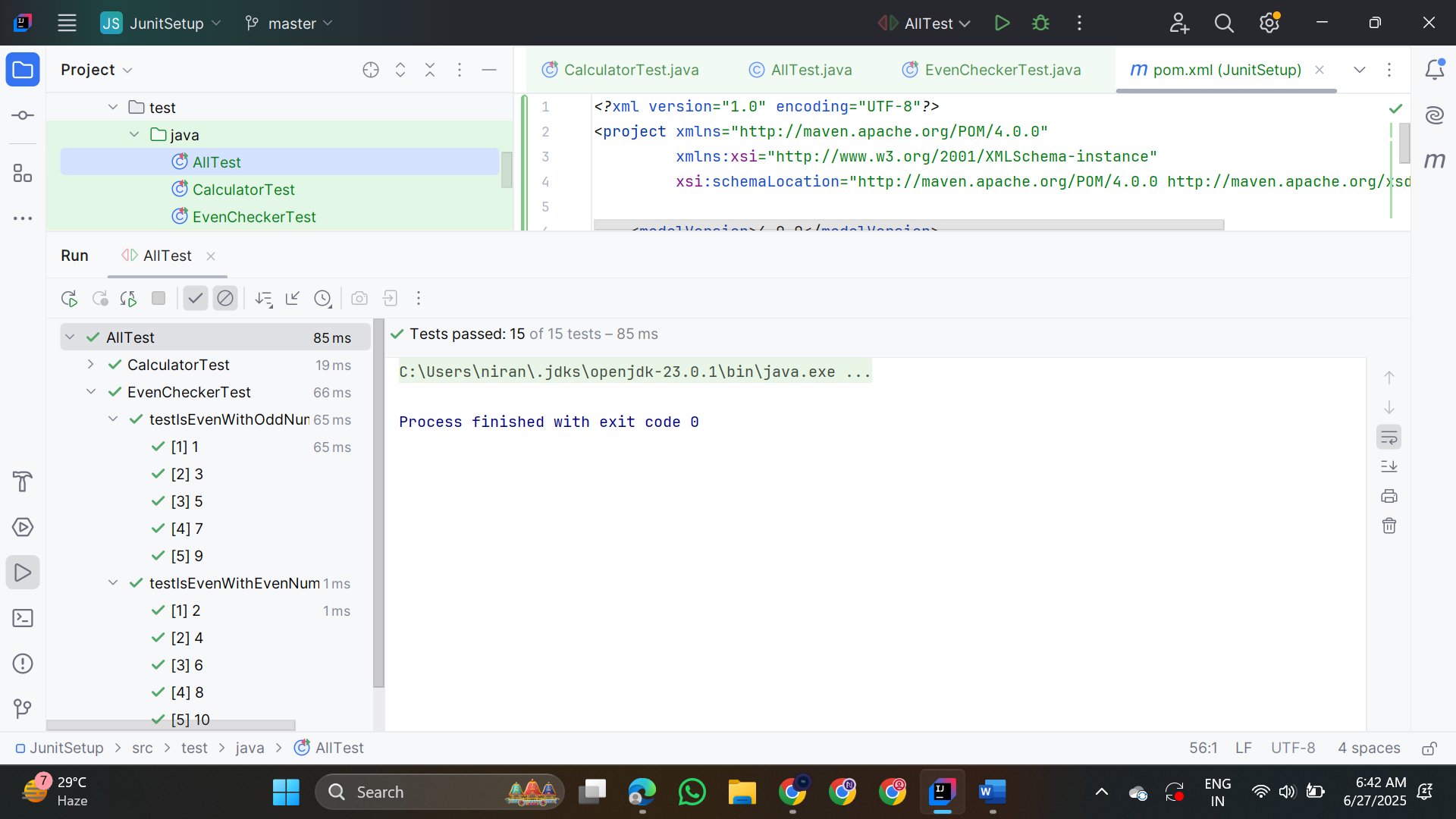
EvenCheckerTest.class

})

public class AllTests {

}





**3.Test Execution Order**

**OrderedTests.java**

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.assertTrue;

@TestMethodOrder(OrderAnnotation.class)

public class OrderedTests {

@Test

@Order(2)

void secondTest() {

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

assertTrue(true);

}

@Test

@Order(1)

void firstTest() {

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

assertTrue(true);

}

@Test

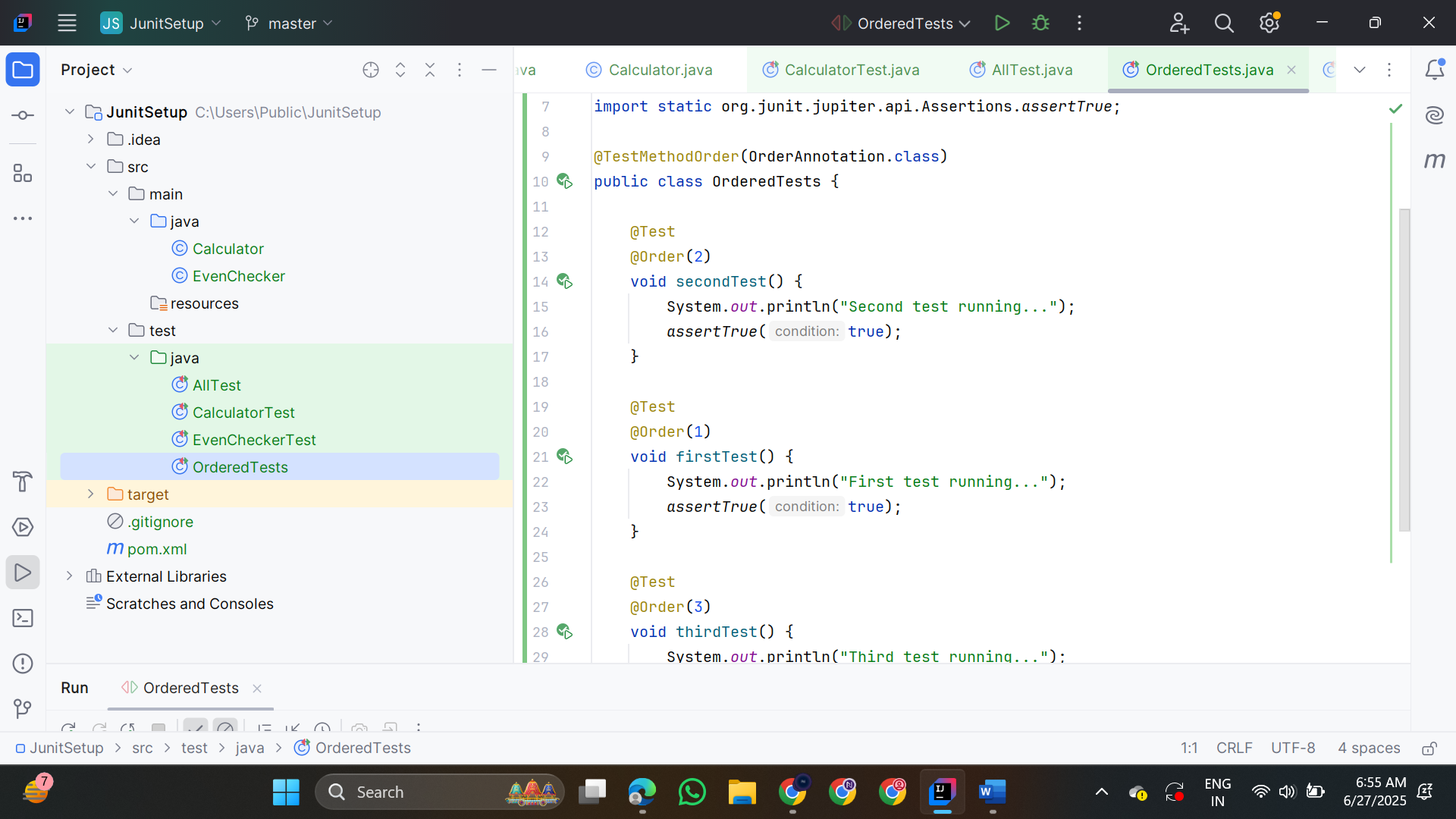
@Order(3)

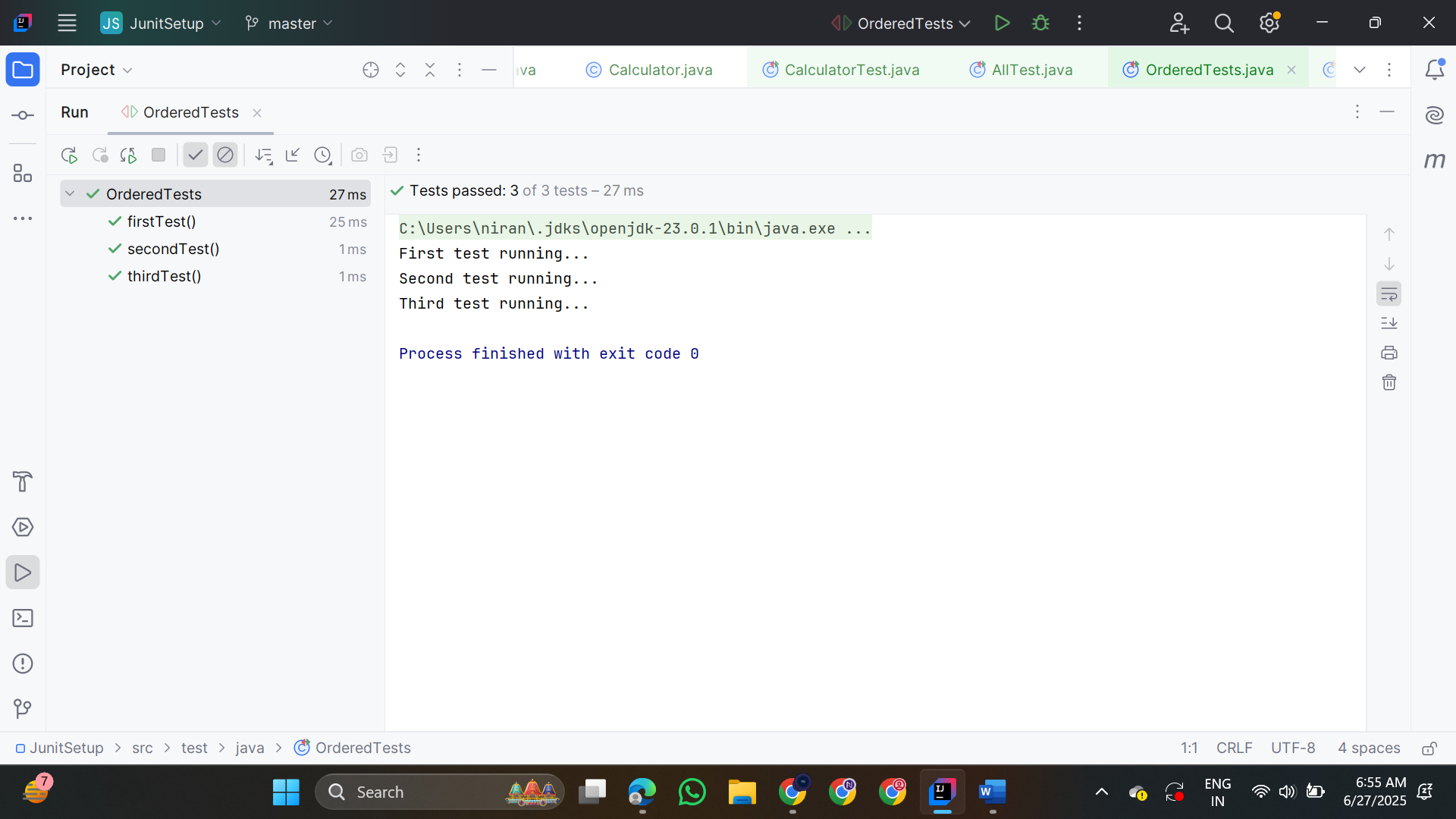
void thirdTest() {

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

assertTrue(true);

}}





**4.Exception Testing**

**ExceptionThrower.java**

public class ExceptionThrower {

public void throwException() throws IllegalArgumentException {

throw new IllegalArgumentException("Invalid input!");

}

}

**ExceptionThrowerTest.java**

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertThrows;

public class ExceptionThrowerTest {

@Test

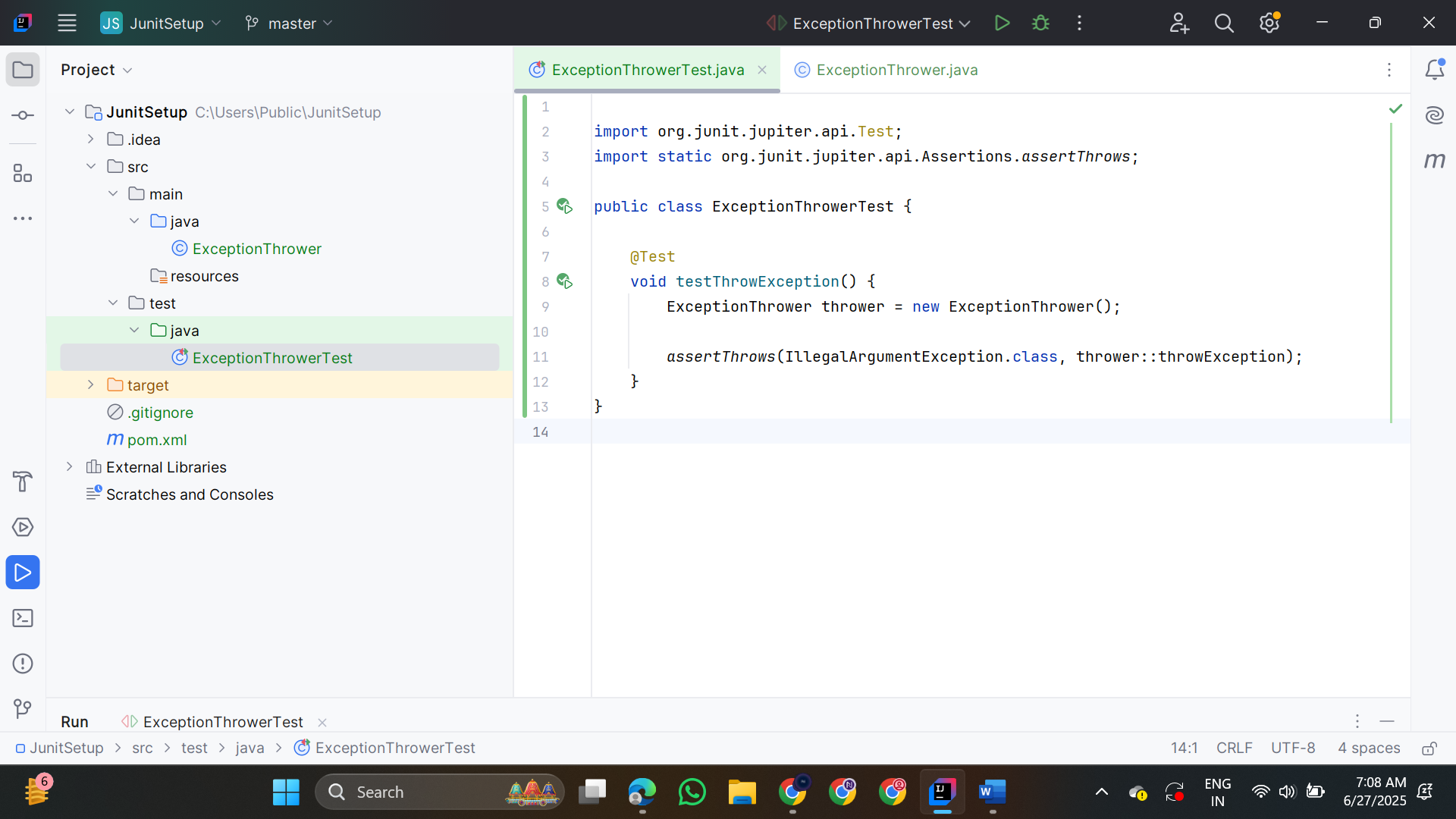
void testThrowException() {

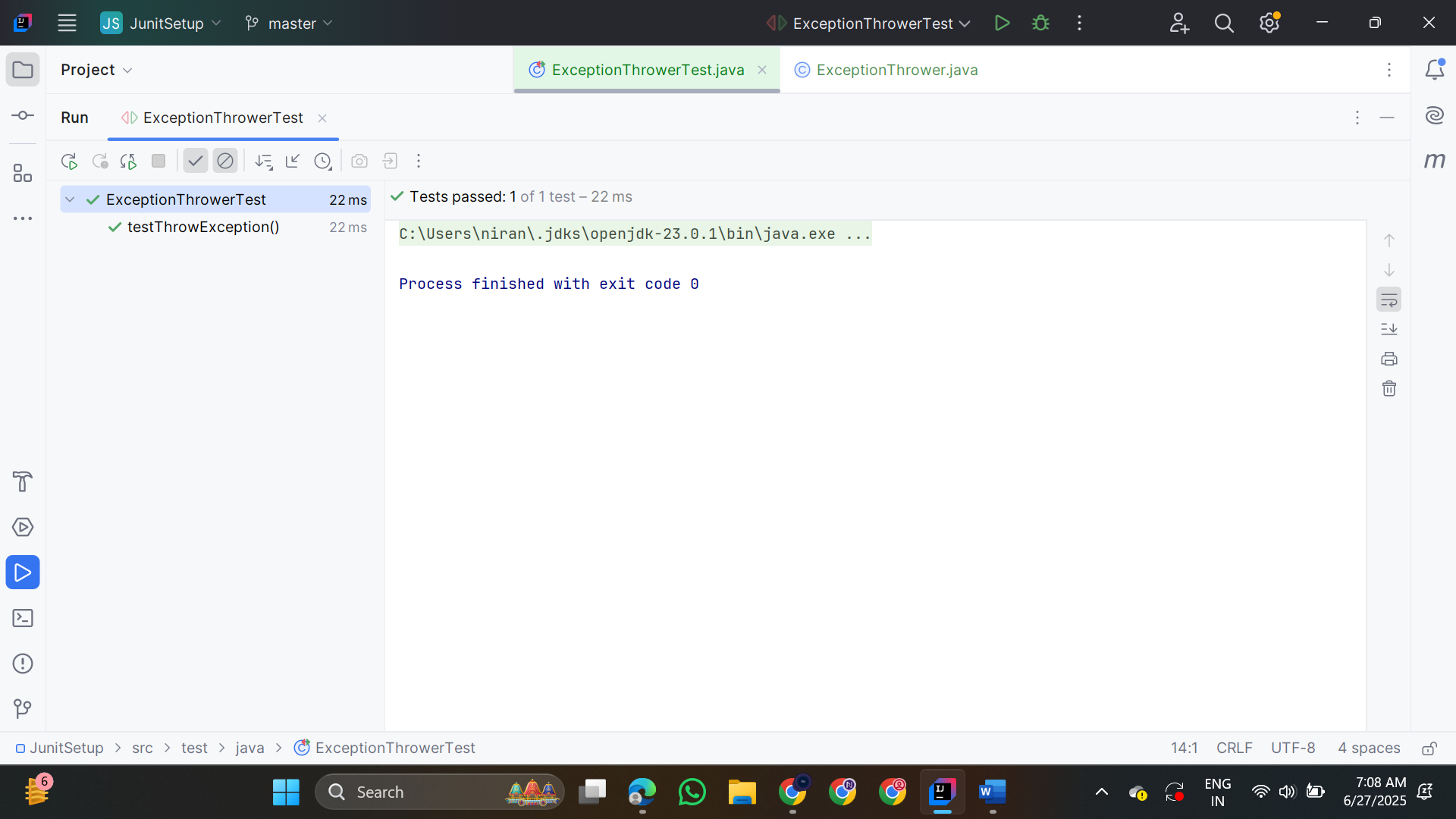
ExceptionThrower thrower = new ExceptionThrower();

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

}

}





**5.Timeout and Performance testing**

**PerformanceTester.java**

public class PerformanceTester {

public void performTask() {

try {

Thread.sleep(500);

} catch (InterruptedException e) {

Thread.currentThread().interrupt();

}

}

}

**PerformanceTesterTest.java**

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.Timeout;

import java.util.concurrent.TimeUnit;

public class PerformanceTesterTest {

@Test

@Timeout(value = 1, unit = TimeUnit.SECONDS)

void testPerformTaskCompletesInTime()

{

PerformanceTester tester = new PerformanceTester();

tester.performTask();

}

}

