Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління

Зображення, що містить текст

Автоматично згенерований опис

**Звiт**

до лабораторної роботи № 1

з дисципліни

# “ Прикладне програмування”

Виконав: студент групи ОІ-11СП

**Богдан Заяць**

Прийняв:

**МельникР.В.**

Львів – 2024

**Завдання:**

1. Згідно з індивідуальним завданням напишіть програму (з використанням об’єктно-орієнтованого підходу), що відповідає наступним вимогам:

• Програма визначає і опрацьовує задані числа Фібоначчі (1, 1, 2, 3, 5, … ) або Люка (1, 3, 4, 7, 11, …). Номер числа та інші вхідні дані користувач має можливість передати через командний рядок, або в іншому випадку вводить з клавіатури.

• Створіть клас, який містить дані про номер та значення числа Фібоначчі або Люка, методи доступу до даних та інші методи, відповідно до індивідуального завдання.

• У головній функції оголосіть об’єкти або масив об’єктів створеного класу, які зберігають та опрацьовують введені дані.

• Програма виводить на екран вхідні дані і результати обчислень та перевірок із відповідни ми підказками.

• Скомпілюйте та запустіть програму через командний рядок та за допомогою обраного середовища розробки.

2. Продокументуйте програму з п. 1 з використанням JavaDoc і згенеруйте на основі нього документацію.

**Індивідуальне завдання**

Варіант 5

Визначити, які числа серед перших *N* чисел Фібоначчі є простими.

**Програма:**

**FibonacciNumber.java**

package edu.bogdan.zaiats.lab1;  
  
*/\*\*  
 \* stores value and index of a Fibonacci number  
 \*  
 \* @param value a real whole number  
 \* @param idx position in the sequence  
 \*\*/*public record FibonacciNumber(long value, int idx) {  
  
 */\*\*  
 \* gets a Fibonacci number from the index  
 \*  
 \* @param index Position of the num in the sequence  
 \* starts from 0  
 \* @return FibonacciNumber  
 \*\*/* public static FibonacciNumber fromIdx(int index) {  
 if (index <= 0) {  
 return new FibonacciNumber(0, 0);  
 }  
  
 long prev = 0, next = 1;  
 int idx = 1;  
 for (; idx < index; idx++) {  
 long numSum = next + prev;  
 prev = next;  
 next = numSum;  
 }  
  
 return new FibonacciNumber(next, idx);  
 }  
  
 */\*\*  
 \* <pre>  
 \* Makes next Fibonacci number if the previous\_number is it's previous number in the sequence  
 \* Otherwise return null  
 \* </pre>  
 \*  
 \* @param previous previous Fibonacci Number  
 \* @return FibonacciNumber  
 \*\*/* public FibonacciNumber next(FibonacciNumber previous) {  
 if (!(idx - previous.idx() == 1)) {  
 return null;  
 }  
 return new FibonacciNumber(previous.value() + value, idx + 1);  
 }  
}

**IsPrime.java**

package edu.bogdan.zaiats.lab1;  
  
*/\*\*  
 \* Tells if the number is prime or not  
 \*/*public final class IsPrime {  
 */\*\*  
 \* Tells if the number is prime or not  
 \*  
 \* @param num any whole positive number  
 \* @return boolean  
 \*/* public static boolean isPrime(long num) {  
 if (num < 2) {  
 return false;  
 }  
  
 for (int i = 2; i <= Math.*sqrt*(num); i++) {  
 if (num % i == 0) {  
 return false;  
 }  
 }  
 return true;  
 }  
}

**Main.java**

package edu.bogdan.zaiats.lab1;  
  
import java.util.ArrayList;  
import java.util.Scanner;  
  
import static edu.bogdan.zaiats.lab1.IsPrime.*isPrime*;  
  
*/\*\*  
 \* <pre>  
 \* ЛР 1  
 \* Виконав ст. гр. ОІ-11СП  
 \* Програма визначає й опрацьовує задані числа Фібоначчі (1, 1, 2, 3, 5, … ) або Люка (1, 3, 4, 7, 11, …).  
 \* Номер числа та інші вхідні дані користувач має можливість передати через командний рядок,  
 \* або в іншому випадку вводить з клавіатури.  
 \* ## індивідуальне завдання  
 \* Визначити, які числа серед перших N чисел Фібоначчі є простими.  
 \* Аргументи:  
 \* --from\_idx  
 \* генерує числа від заданої точки  
 \* </pre>  
 \* Ліміт - 69  
 \* @author Богдан Заяць  
 \* @version 1.0  
 \*\*/*public class Main {  
 private static final String *FLAG* = "--from\_idx";  
 private static final int *LIMIT* = 69;  
  
 */\*\*  
 \* args --from\_idx  
 \*/* public static void main(String[] args) {  
 var reader = new Scanner(System.*in*);  
 int idx = 0;  
  
 if (args.length > 1) {  
 System.*err*.println("Too many args");  
 return;  
 }  
  
 if (args.length == 1) {  
 var flag = args[0];  
 if (flag.equals(*FLAG*)) {  
 System.*out*.println("Enter the index of a Fibonacci number");  
 idx = reader.nextInt();

if (idx < 0){  
 System.*err*.println("negative index");  
 return;  
 }

} else {  
 System.*err*.println("incorrect flag");  
 return;  
 }  
 }  
  
 System.*out*.println("Enter the len of Fibonacci sequence numbers");  
 int seq\_len = reader.nextInt();  
  
 if (seq\_len <= 0) {  
 System.*err*.println("You get none of the Fibonacci numbers");  
 return;  
 }  
 if (seq\_len + idx > *LIMIT*) {  
 System.*err*.println("You want too much from my pc");  
 return;  
 }  
 var sequence = *getFibonacciSequence*(idx, seq\_len);  
 System.*out*.println("Fibonacci sequence");  
 sequence.forEach(i -> {  
 var is\_prime = *isPrime*(i.value());  
 System.*out*.printf("index %5d, value %5d, is prime: %b, \n", i.idx(), i.value(), is\_prime);  
 });  
 }  
  
 */\*\*  
 \* getFibonacciSequence  
 \*  
 \* @param seq\_len Length of the Fibonacci sequence  
 \* @return ArrayList*<*FibonacciNumber*>  
 *\*\*/* private static ArrayList<FibonacciNumber> getFibonacciSequence(int start\_idx, int seq\_len) {  
 var sequence = new ArrayList<FibonacciNumber>();  
 sequence.addFirst(FibonacciNumber.*fromIdx*(start\_idx));  
  
 if (seq\_len <= 1) {  
 return sequence;  
 }  
  
 sequence.add(FibonacciNumber.*fromIdx*(start\_idx + 1));  
  
 for (int i = 2; i < seq\_len; i++) {  
 var current = sequence.get(i - 1).next(sequence.get(i - 2));  
 sequence.add(current);  
 }  
 return sequence;  
 }  
}

**Lab1Test.java**

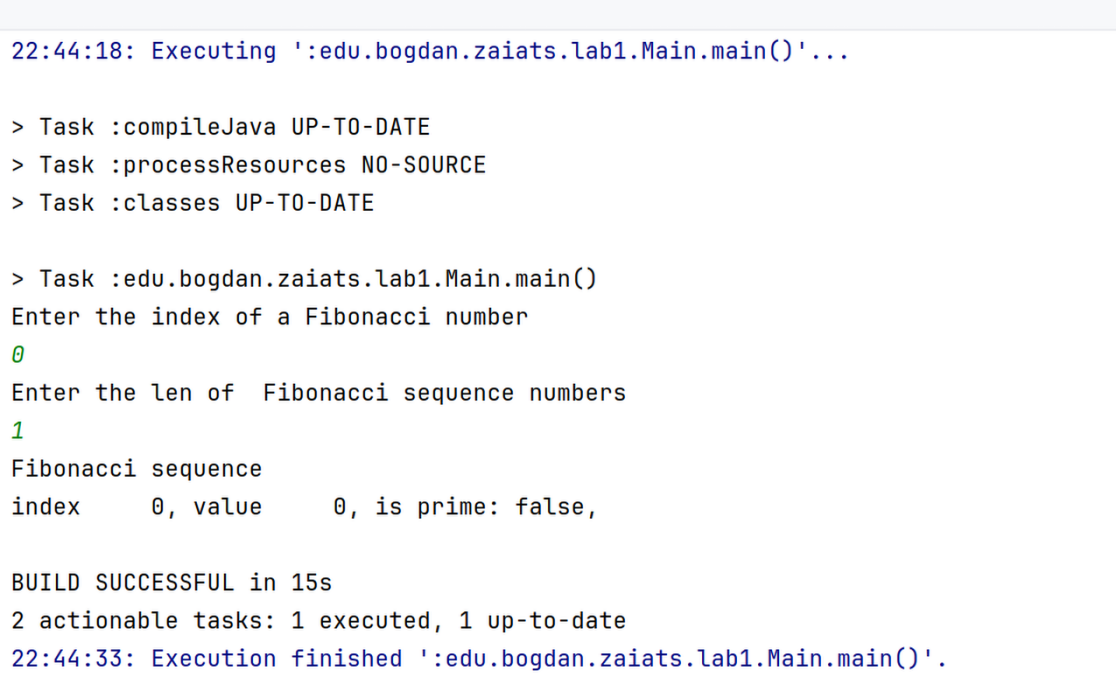
import edu.bogdan.zaiats.lab1.FibonacciNumber;  
import org.junit.jupiter.api.Test;  
  
import static edu.bogdan.zaiats.lab1.IsPrime.*isPrime*;  
import static org.junit.jupiter.api.Assertions.\*;  
  
  
public class Lab1Test {  
 @Test  
 public void isPrimeTest (){  
 *assertTrue*(*isPrime*(2));  
 *assertTrue*(*isPrime*(3));  
 *assertTrue*(*isPrime*(5));  
 *assertTrue*(*isPrime*(7));  
 *assertTrue*(*isPrime*(47));  
 }  
  
 @Test  
 public void isNotPrimeTest (){  
 *assertFalse*(*isPrime*(0));

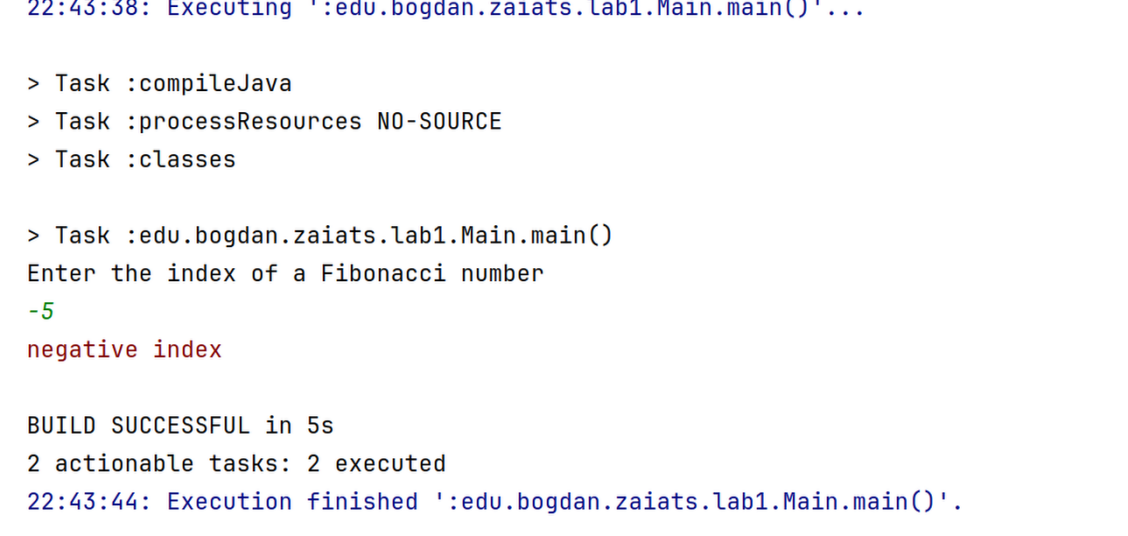
*assertFalse*(*isPrime*(1));  
 *assertFalse*(*isPrime*(4));  
 *assertFalse*(*isPrime*(9));  
 *assertFalse*(*isPrime*(81));  
 *assertFalse*(*isPrime*(49));  
 *assertFalse*(*isPrime*(69));  
 *assertFalse*(*isPrime*(42));  
 }  
  
 @Test  
 public void fromIndexFibonacciTest (){  
 *assertEquals*(new FibonacciNumber(0,0), FibonacciNumber.*fromIdx*(0));  
 *assertEquals*(new FibonacciNumber(1,1), FibonacciNumber.*fromIdx*(1));  
 *assertEquals*(new FibonacciNumber(1,2), FibonacciNumber.*fromIdx*(2));  
 *assertEquals*(new FibonacciNumber(2,3), FibonacciNumber.*fromIdx*(3));  
 *assertEquals*(new FibonacciNumber(3,4), FibonacciNumber.*fromIdx*(4));  
 }  
  
  
 @Test  
 public void nextFibonacciTest (){  
 var prev = new FibonacciNumber(1,2);  
 var curr = new FibonacciNumber(2,4);  
 *assertEquals*(new FibonacciNumber(4,3), curr.next(prev));  
 }  
}

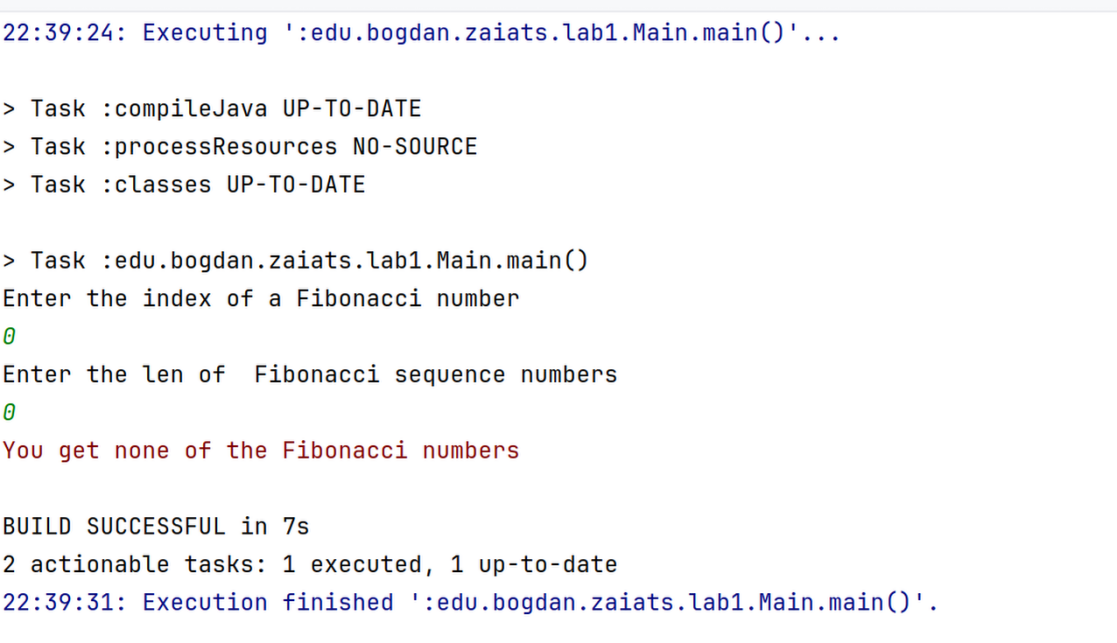
**Результати виконання:**

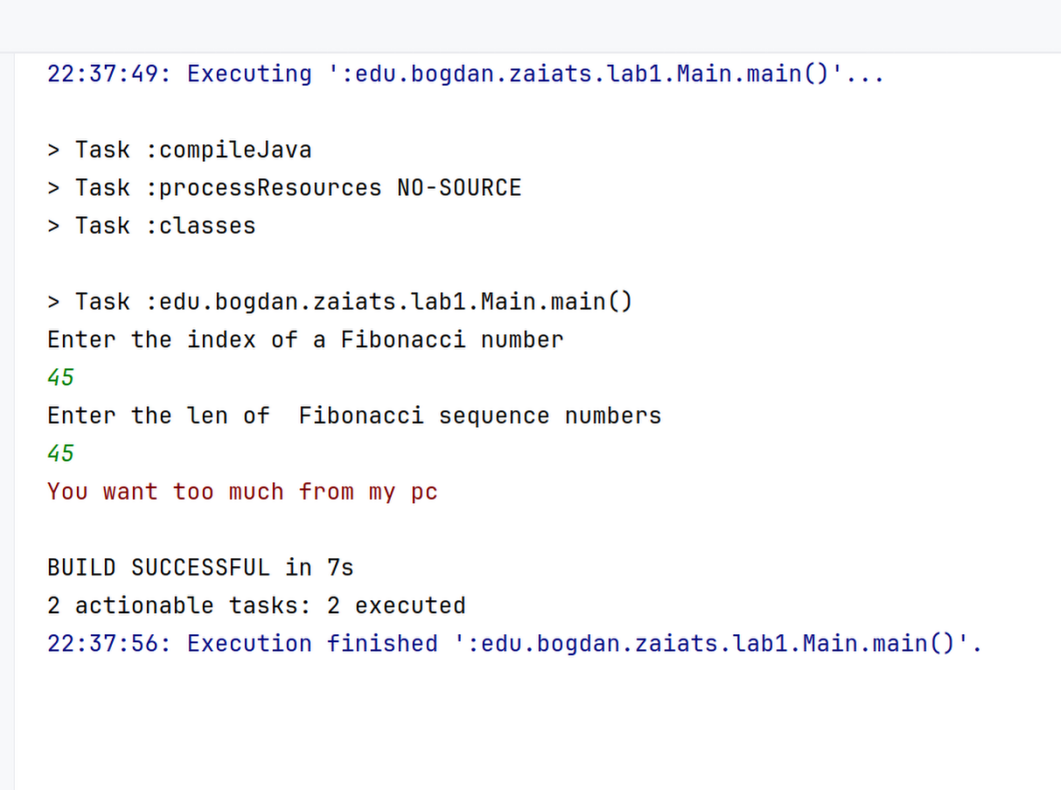
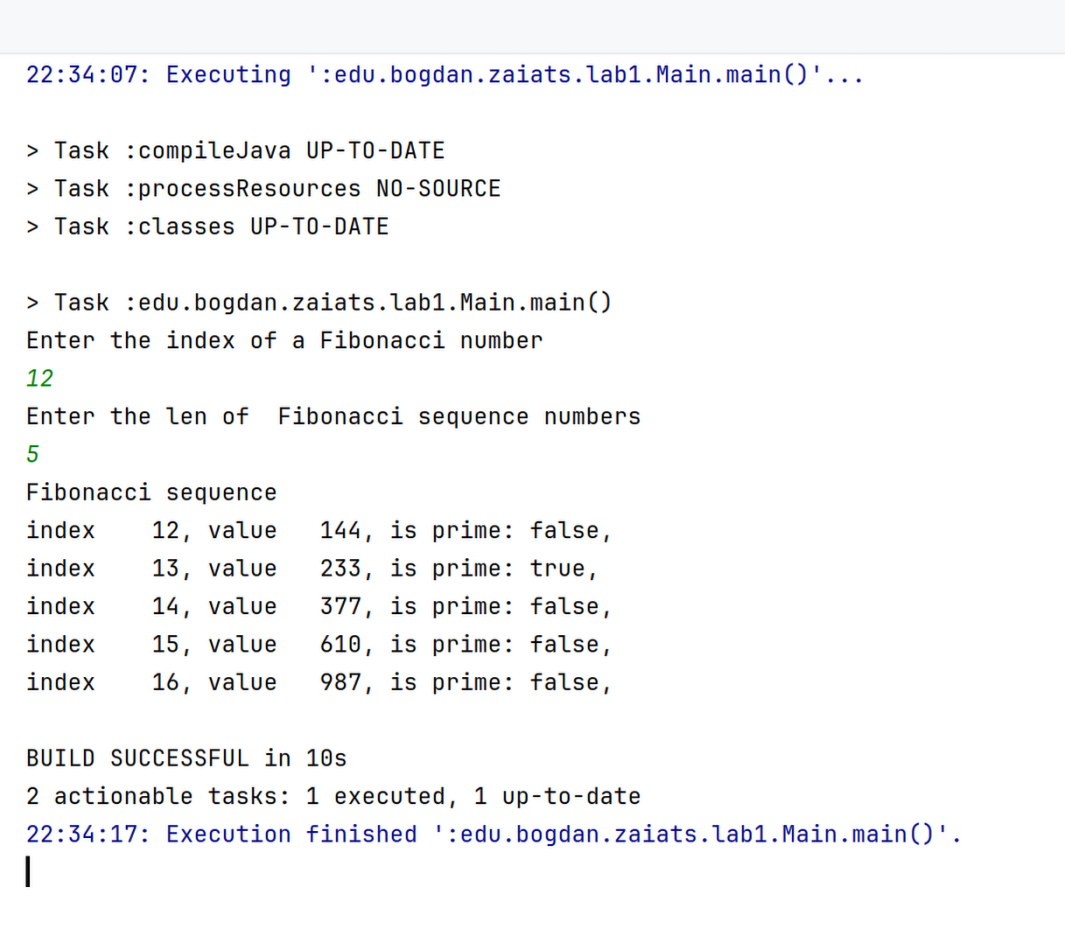
**З прапорцем**

**--from\_idx**

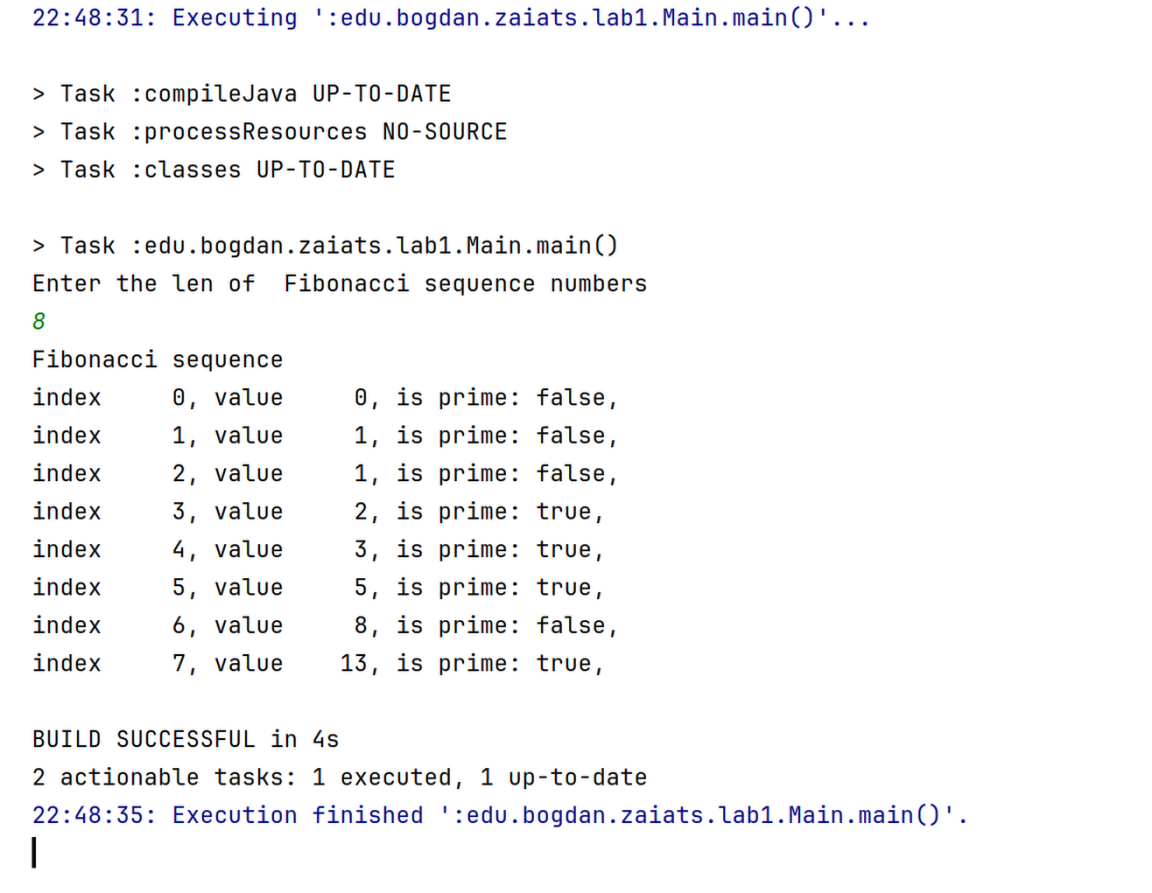
****

****

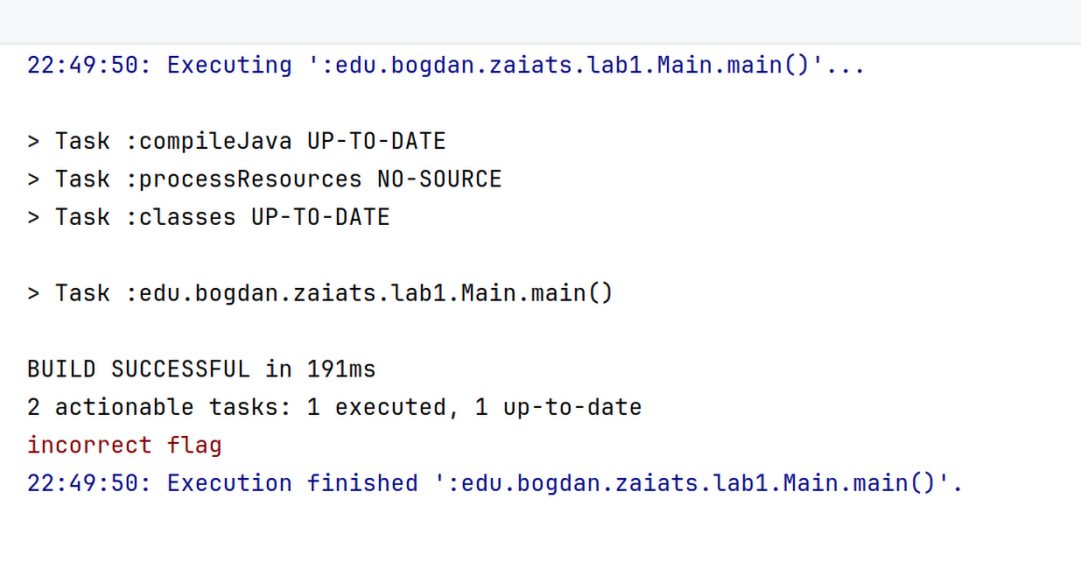
****

**ґ**

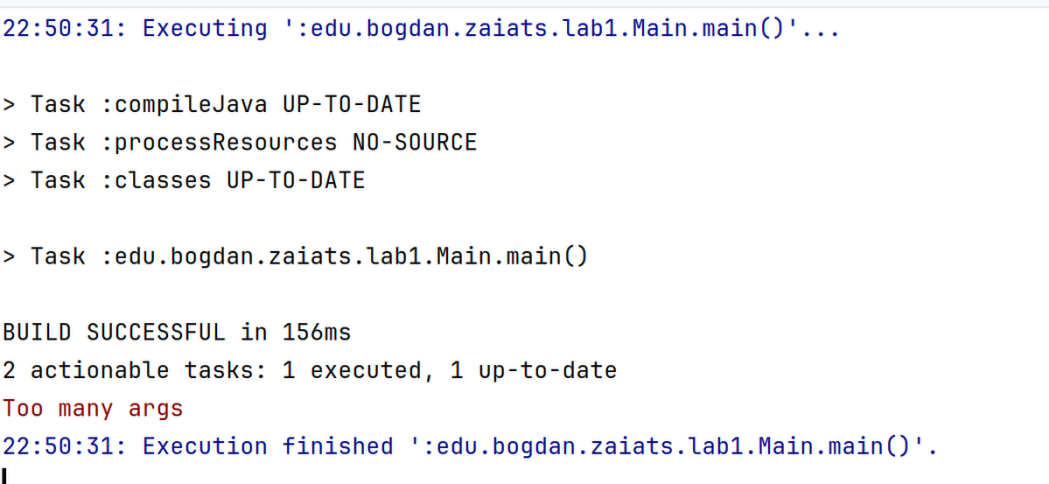
**Без прапорця**

****

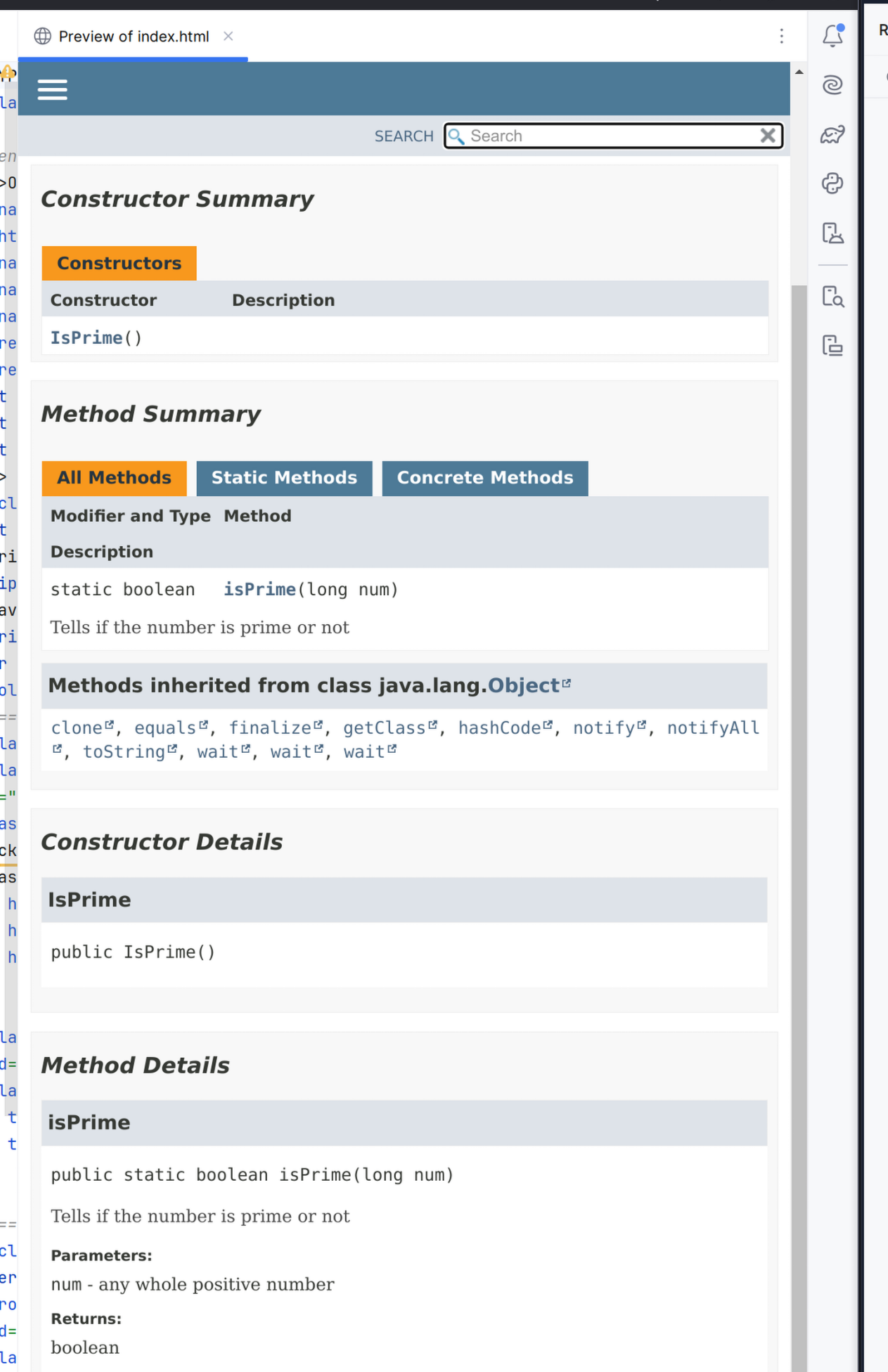
**Хибний прапорець**

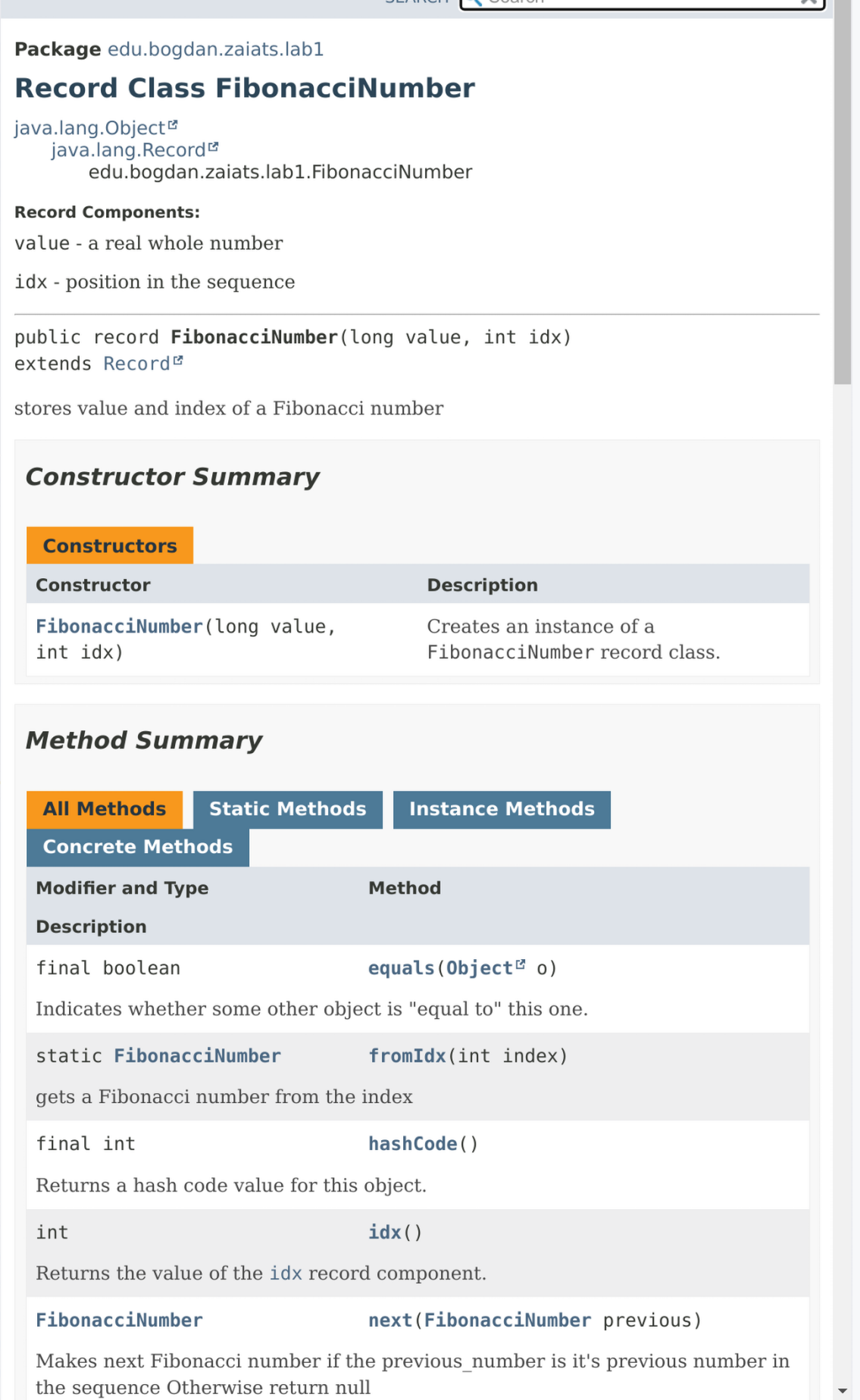
****

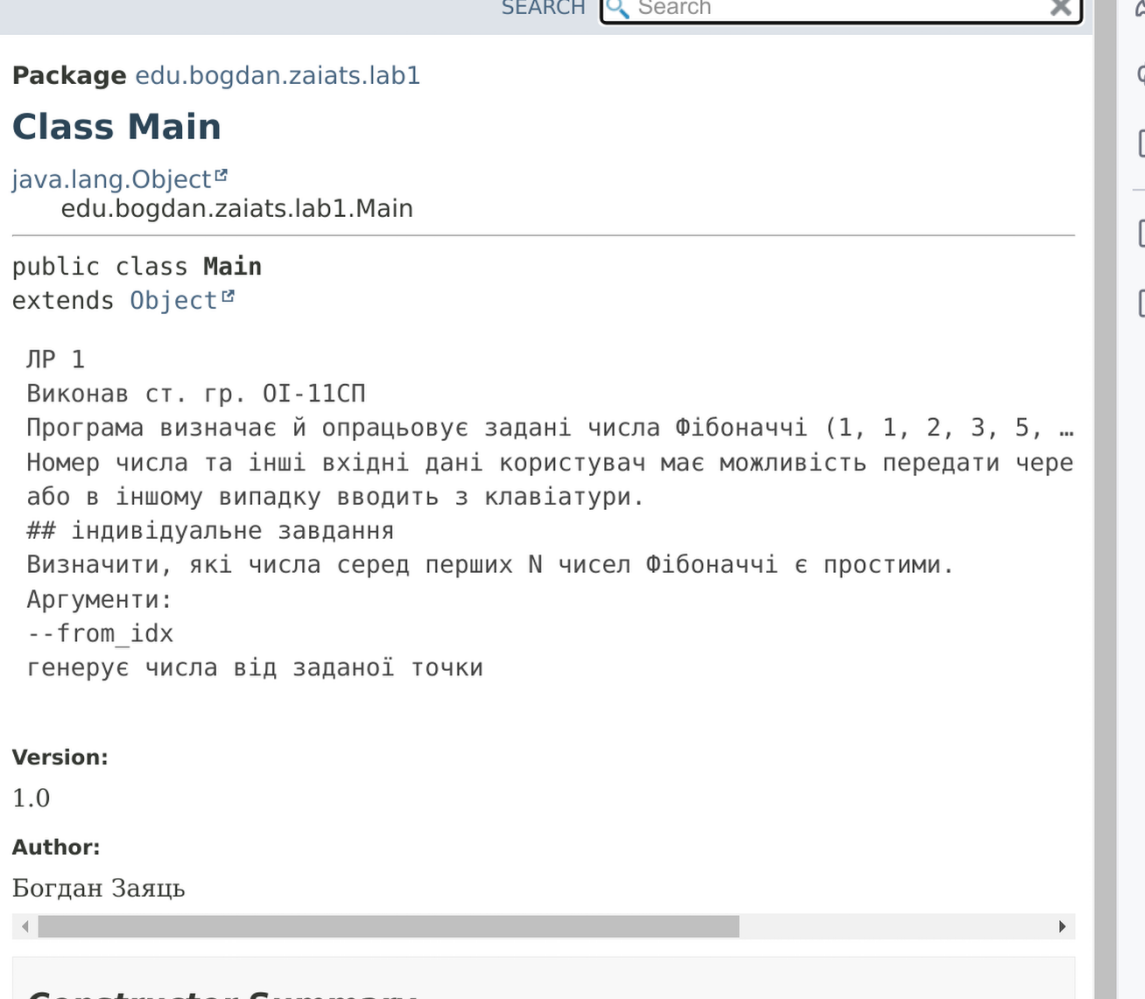
**Забагато прапорців**

****

**Документація**

****

****

****

**Висновок:**

В цій лабораторній роботі я вивчив основи Java.