Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



Звіт

до **лабораторної роботи № 1**

з дисципліни

**Прикладне програмування**

на тему:

**“Основи Java”**

Виконала: студентка ОІ-25

Синичак Ю. М.

Прийняв: доц. кафедри АСУ

Рашкевич Ю.Ю.

Львів – 2024

**Тема:** Ознайомитись з мовою програмування Java та методом документації за допомогою JavaDoc

**Завдання:**

Визначити *N*-е число Фібоначчі для заданого *N*≤0.

**Вимоги:**

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 відповідними 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.

**Код програми:**

Main.java:

**package fibonacci*A*pp;**

**import java.util.Scanner;**

**/\*\***

**\* Main class that executes the program.**

**\* It prompts the user to enter a number N and calculates the N-th Fibonacci number.**

**\*/**

**public class Main {**

**/\*\***

**\* The main method where the program starts.**

**\* It asks the user to input a number and outputs the corresponding Fibonacci number.**

**\***

**\* @param args Command line arguments (not used).**

**\*/**

**public static void main(String[] args) {**

**int n = getUserInput();**

**// Create and calculate the Fibonacci sequence for the input value.**

**NumberSequence sequence = new Fibonacci(n);**

**System.out.println("N-th Fibonacci number: " + sequence.calculate());**

**}**

**/\*\***

**\* Method to read a valid integer input from the user.**

**\* Re-prompts the user until a valid integer is entered.**

**\***

**\* @return The valid integer input.**

**\*/**

**private static int getUserInput() {**

**Scanner scanner = new Scanner(System.in);**

**int n = -1;**

**while (true) {**

**System.out.print("Enter a non-negative integer N: ");**

**if (scanner.hasNextInt()) {**

**n = scanner.nextInt();**

**if (n <= 0) {**

**break; // Exit loop if valid non-negative integer**

**} else {**

**System.out.println("Please enter a negative integer.");**

**}**

**} else {**

**System.out.println("Invalid input. Please enter a number.");**

**scanner.next(); // Discard invalid input**

**}**

**}**

**scanner.close(); // Close the scanner to prevent resource leak**

**return n;**

**}**

**}**

NumberSequence.java:

package fibonacci*A*pp;

/\*\*

\* Abstract class representing a number sequence.

\* It serves as a base class for specific sequences like Fibonacci.

\*/

public abstract class NumberSequence {

protected int n;

/\*\*

\* Constructor that sets the number N for the sequence.

\*

\* @param n The position of the number in the sequence.

\*/

public NumberSequence(int n) {

this.n = n;

}

/\*\*

\* Abstract method to calculate the N-th number in the sequence.

\*

\* @return The N-th number in the sequence.

\*/

public abstract int calculate();

/\*\*

\* Getter method for the number N.

\*

\* @return The position N.

\*/

public int getN() {

return n;

}

}

Fibonacci.java:

package fibonacci*A*pp;

/\*\*

\* Class for calculating Fibonacci numbers.

\* It inherits from the NumberSequence class.

\*/

public class Fibonacci extends NumberSequence {

/\*\*

\* Constructor for Fibonacci sequence, sets the number N.

\*

\* @param n The position of the Fibonacci number to calculate.

\*/

public Fibonacci(int n) {

super(n);

}

/\*\*

\* Method to calculate the N-th Fibonacci number.

\* If N <= 0, returns 0. If N == 1, returns 1.

\* For N >= 2, it iterates to calculate the N-th Fibonacci number.

\*

\* @return The N-th Fibonacci number.

\*/

@Override

public int calculate() {

if (n == 0) return 0;

if (n == 1 || n == -1) return 1;

int a = 0, b = 1;

for (int i = -2; i >= n; i--) {

int tmp = a - b;

a = b;

b = tmp;

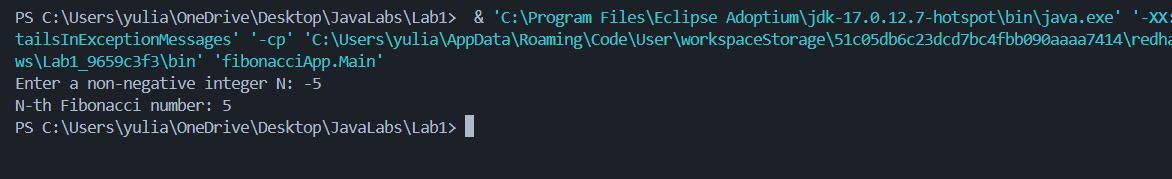
}

return b;

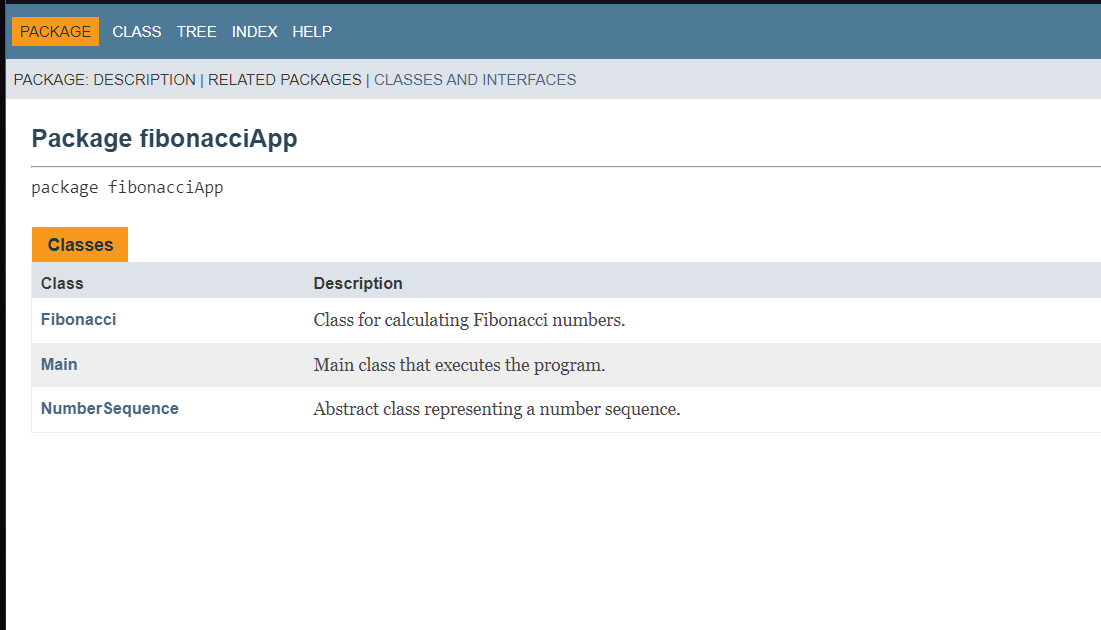
}

}

**Вивід програми:**



**Тепер вписуємо в термінал відповідну команду та отримуємо документацію:**



**Висновок:** я освоїла основи програмування на мові Java та навчилась використовувати JavaDoc.