Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



**Звіт**

З лабораторної роботи №2

Варіант – 16

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Дослідження базових конструкцій мови JAVA»

Виконав: ст. гр. КІ-35

Павлик С. В.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2022

**Мета роботи:** Ознайомитися з базовими конструкціями мови Java та оволодіти навиками написання й автоматичного документування простих консольних програм мовою Java.

**Завдання(Варіант 16)**

1. Написати та налагодити програму на мові Java згідно варіанту. Програма має задовольняти наступним вимогам: програма має розміщуватися в загальнодоступному класі Lab2ПрізвищеГрупа; програма має генерувати зубчатий масив, який міститиме лише заштриховані області квадратної матриці згідно варіанту; розмір квадратної матриці і символ-заповнювач масиву вводяться з клавіатури; при не введені або введенні кількох символів-заповнювачів відбувається коректне переривання роботи програми; сформований масив вивести на екран і у текстовий файл; програма має володіти коментарями, які дозволять автоматично згенерувати документацію до розробленої програми.

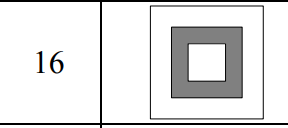
2. Автоматично згенерувати документацію до розробленої програми.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.

4. Дати відповідь на контрольні запитання:

1. Як автоматично згенерувати документацію?
2. В чому різниця між різними варіантами оператора for?

**Варіант завдання:**

****

**Код програми:**

**import java.util.\*;**

**import java.io.\*;**

**/\*\***

**\* Клас Lab\_02 реалізує приклад програми до лабораторної роботи 2**

**\***

**\* @author Pavlyk Serhii**

**\* @version 1.1**

**\* @since version 1.0**

**\***

**\*/**

**public class Lab\_02 {**

**/\*\***

**\* Статичний метод main є точкою входу в програму**

**\***

**\* @param args**

**\***

**\*/**

**public static void main (String [] args)throws IOException**

**{**

**int size = 0;**

**System.out.print("Enter size\t");**

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

**BufferedWriter writer = new BufferedWriter((new FileWriter("MyFile.txt")));**

**while (true)**

**{**

**try**

**{**

**size = scanner\_in.nextInt(); //вводимо розмір матриці з клавіатури**

**if (size <= 3)//перевіряємо чи наша матриця є більшої розмірності ніж 3**

**{**

**System.out.print("The size of the matrix is less than 3!\nRepeat the input\t\t");//якщо ні то виводимо відповідний текст і просимо повторити ввід**

**size = scanner\_in.nextInt();//вводимо розмір матриці з клавіатури**

**scanner\_in.nextLine();**

**}**

**break;**

**}**

**catch (Exception e)**

**{**

**System.out.println("Incorrect input of int. Try again. ");**

**System.out.print("Enter size\t");**

**scanner\_in.nextLine();**

**}**

**}**

**scanner\_in.nextLine();**

**String filler;**

**System.out.print("Enter fill character\t");**

**filler = scanner\_in.nextLine();//вводимо символ заповнювач з клавіатури)**

**while (true)**

**{**

**if (filler.length() != 1)**

**{**

**System.out.println("fill character must be one character!");**

**System.out.print("Enter fill character\t");**

**filler = scanner\_in.nextLine();**

**}**

**else break;**

**}**

**char [][]arr = new char[size][];**

**for (int i = 0; i < size; i++)**

**{**

**if(i==0 ||i==size-1)**

**{**

**System.out.println();**

**writer.write("\n");**

**continue;**

**}**

**arr[1] = new char[size-1];**

**arr[size-2]=new char[size-1];**

**arr[i] = new char[size-1];**

**for (int j = 1; j < size-1; j++)**

**{**

**System.out.print("\t");**

**writer.write("\t");**

**arr[1][j] = (char)filler.codePointAt(0);//заповнюємо верхню горизонтальну частину заштрихованої області**

**arr[i][1] = (char)filler.codePointAt(0);//заповнюємо ліву вертикальну частину заштрихованої області**

**arr[size-2][j] = (char)filler.codePointAt(0);//заповнюємо нижню горизонтальну частину заштрихованої області**

**arr[i][size-2] = (char)filler.codePointAt(0);**

**System.out.print(arr[i][j]);**

**writer.write( arr[i][j]);**

**}**

**System.out.println();**

**writer.write("\n");**

**}**

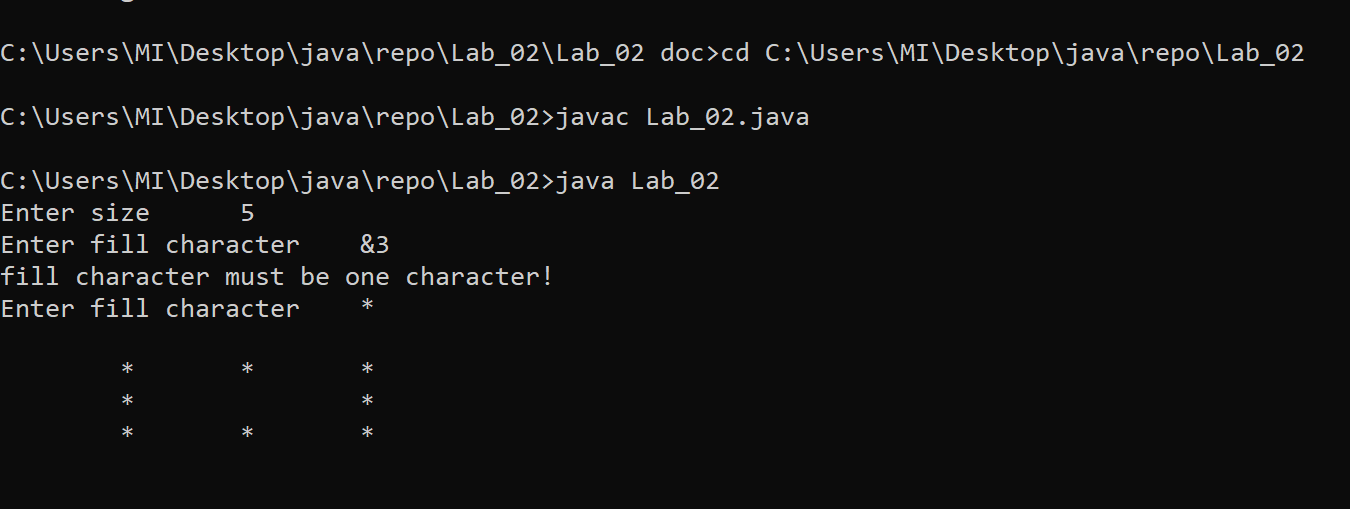
**scanner\_in.close();**

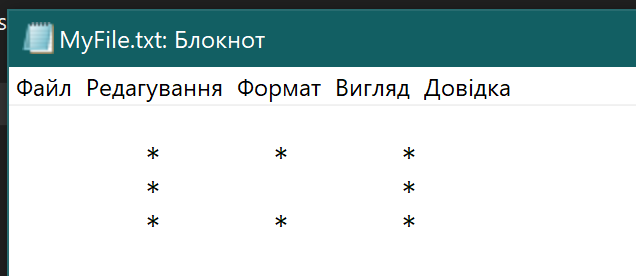
**writer.flush();**

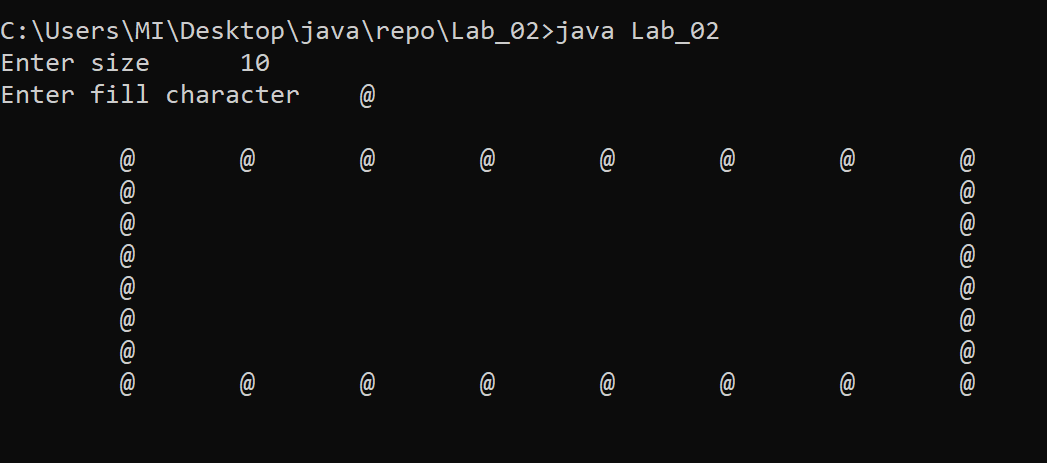
**writer.close();**

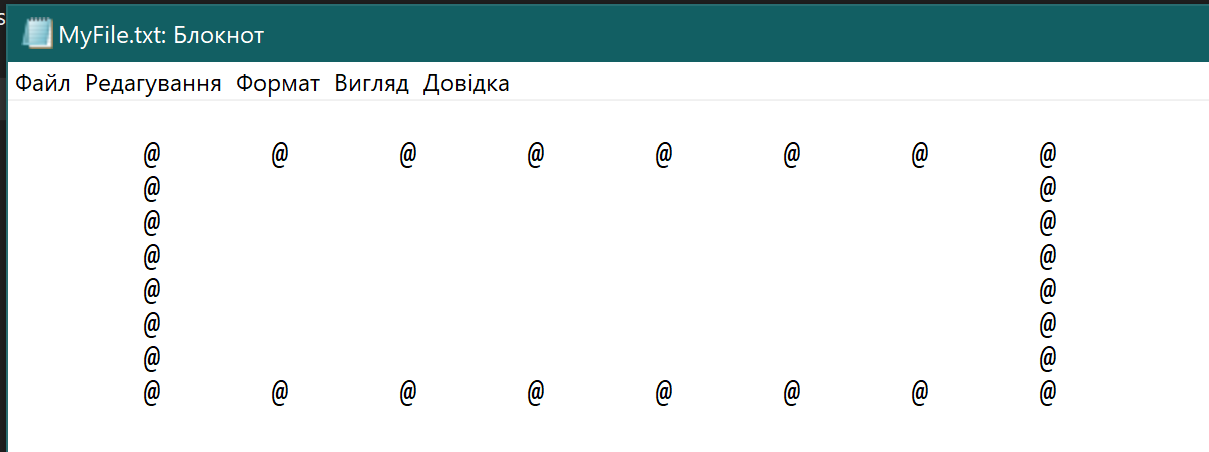
**}**

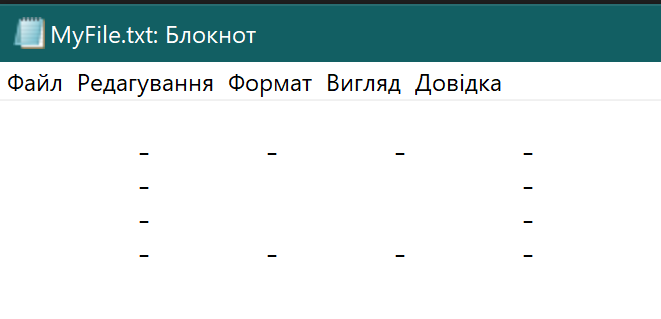
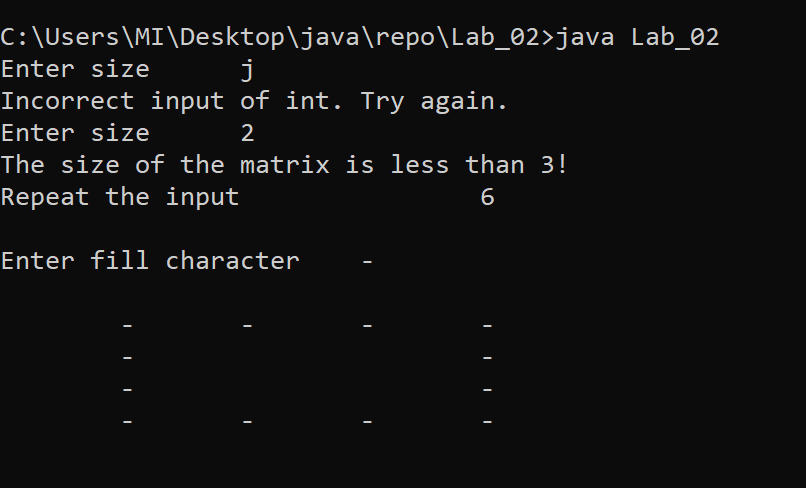
**}**

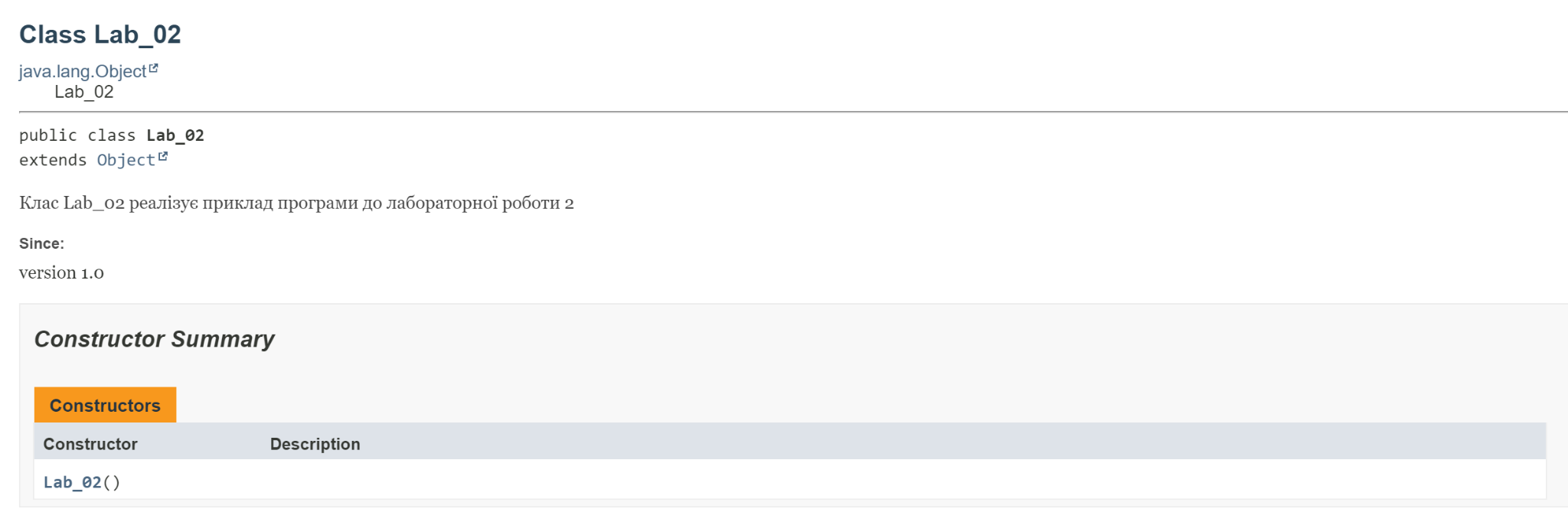
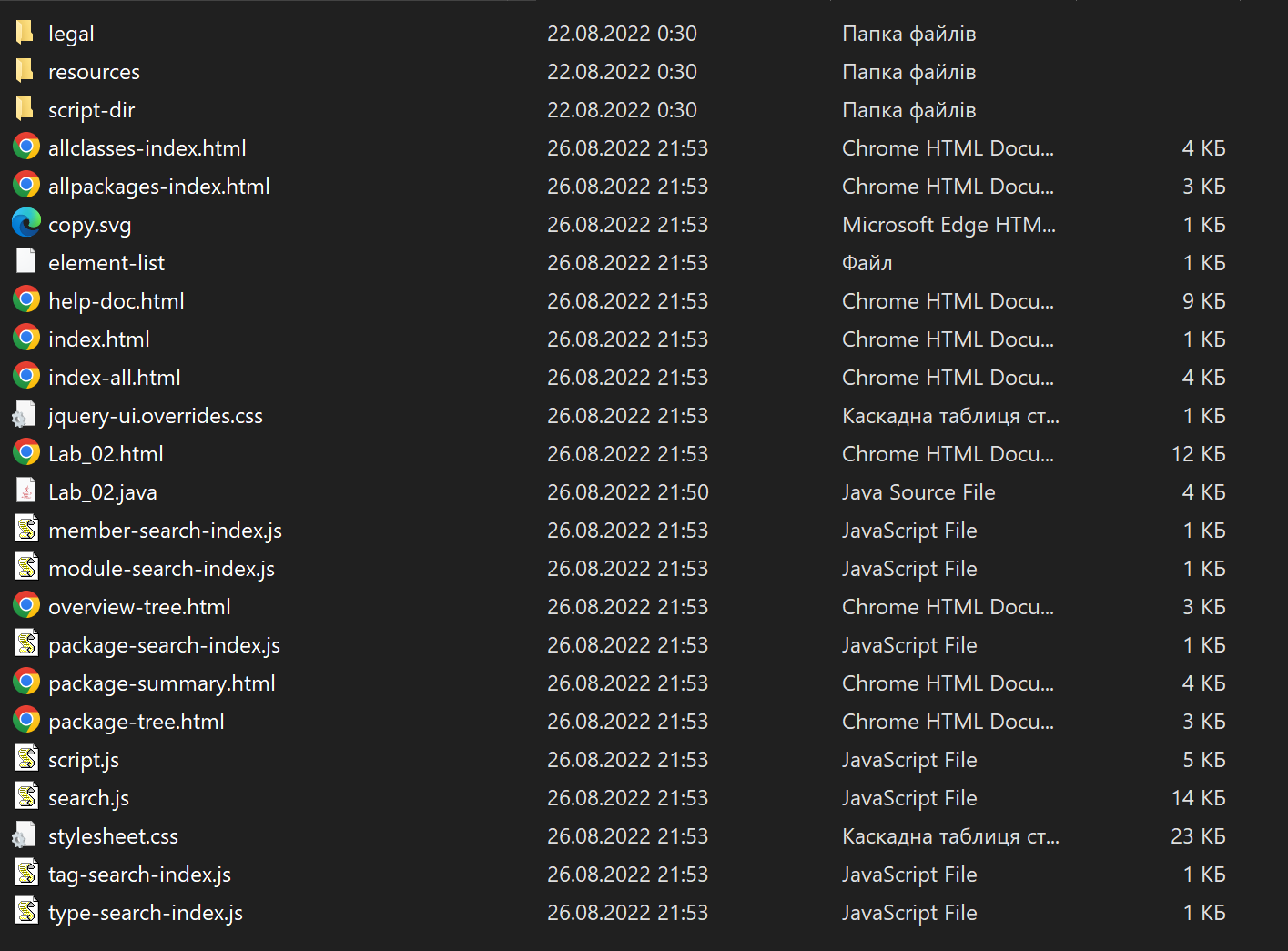
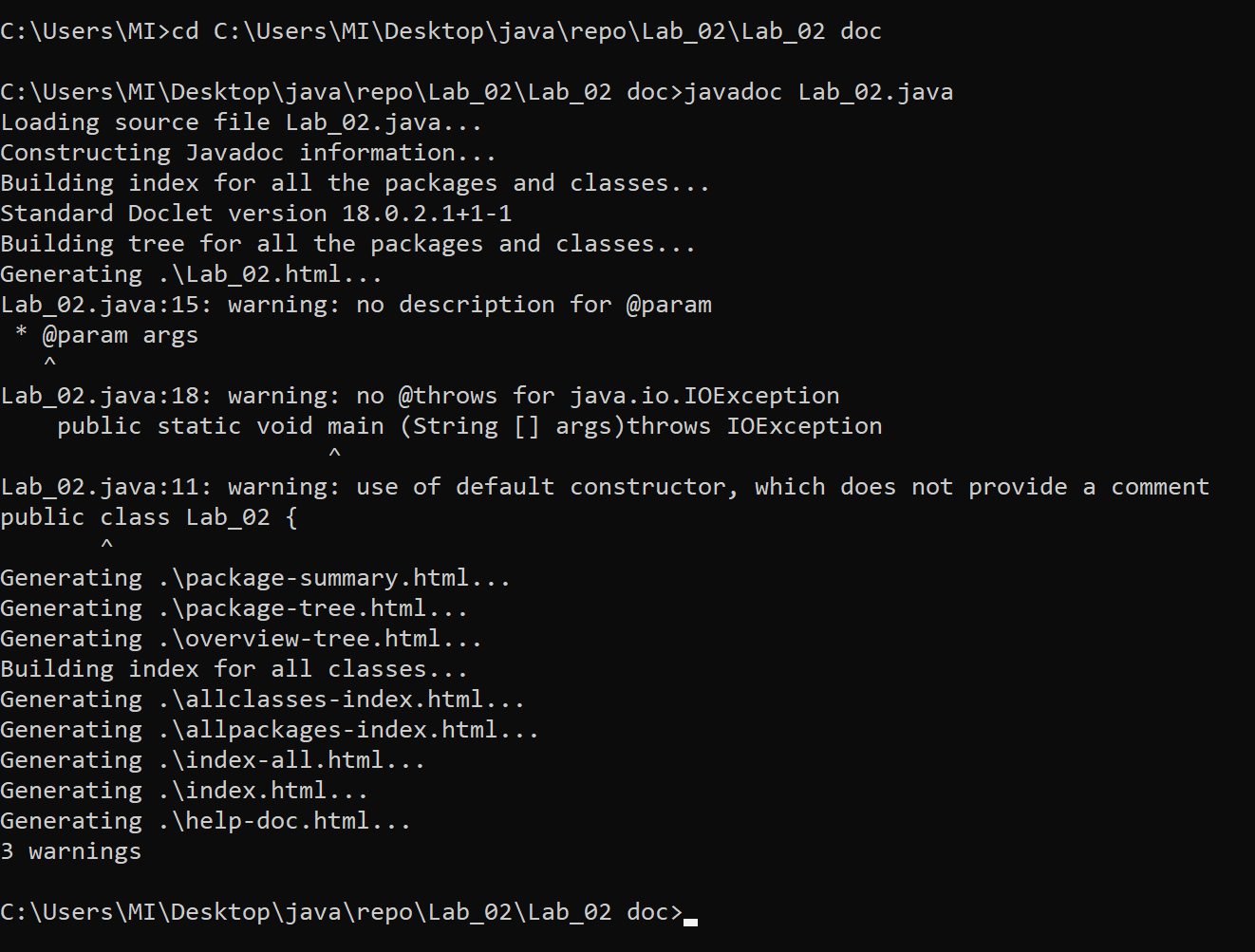
**Результата роботи програми:**

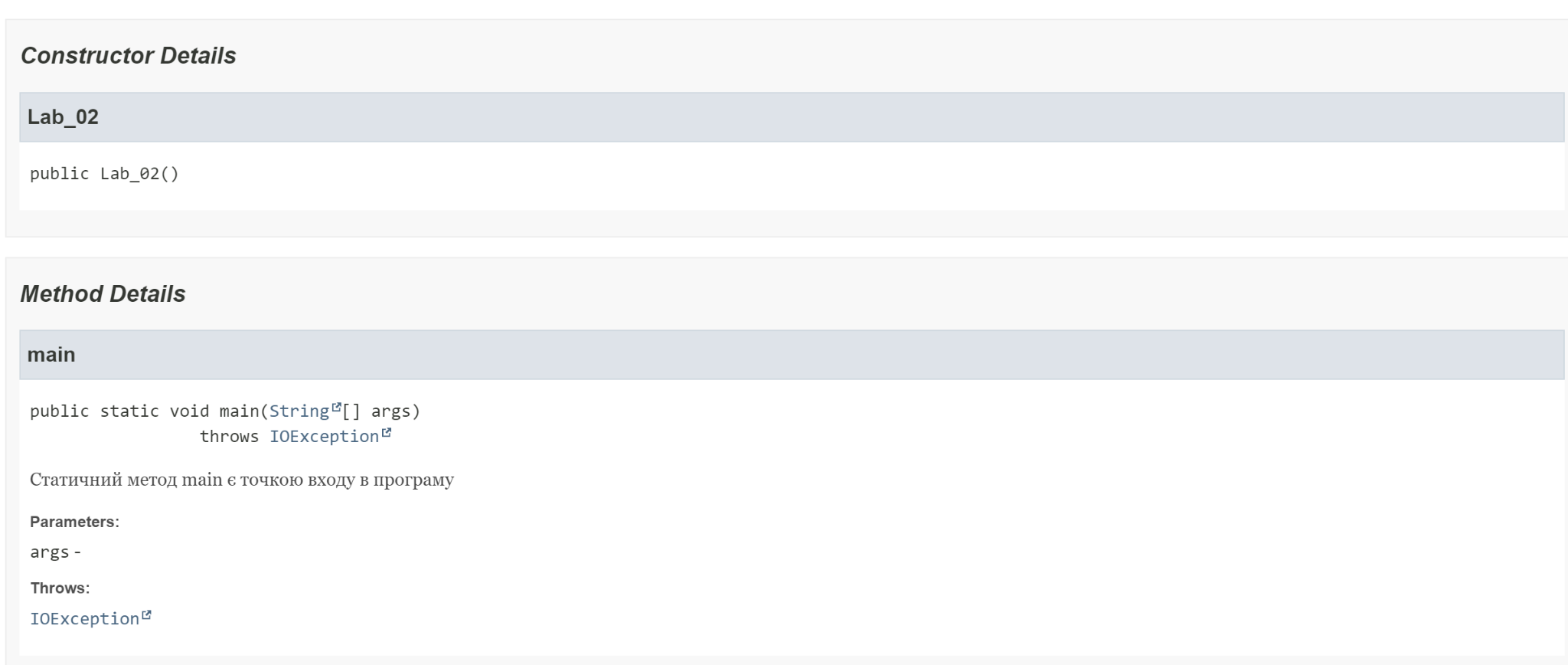






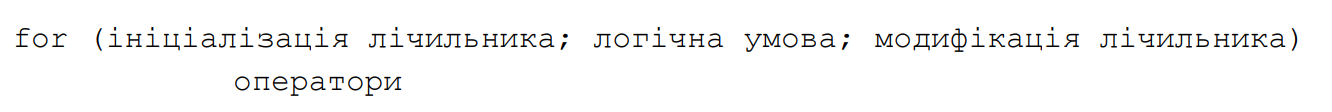
**Фрагмент згенерованої документації**



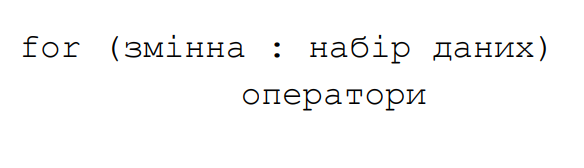


**Відповіді на контрольні запитання**

1. При автоматичній генерації документації використовується утиліта javadoc, яка аналізує вміст між /\*\* і \*/ та на його базі генерує документацію у форматі \*.html. Коментарі між /\*\* і \*/ прийнято починати з описового тексту, за яким слідують дескриптори. Використання дескрипторів полегшує як автоматичну генерацію документації, так і розуміння коду, до якого відноситься коментар.
2. Синтаксис оператора for в стилі С/С++ має такий вигляд:



Оператор циклу for з синтаксисом foreach дозволяє послідовно перебирати всі елементи набору даних без застосування лічильника. Таким набором даних може бути будь-який клас, що реалізує інтерфейс Iterable, або масив. Оператор циклу for з синтаксисом foreach має наступний вигляд:



**Висновок:** Я ознайомився з базовими конструкціями мови Java та оволодів навиками написання й автоматичного документування простих консольних програм мовою Java.