Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



**Звіт**

З лабораторної роботи №1

Варіант№8

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Дослідження базових конструкцій мови JAVA»

Виконав: ст. гр. КІ-304

Кісіль В. І.

Прийняв:

Олексів М.В.

Львів 2023

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

**Теоретичні відомості**

**Зубчаті масиви**

Завдяки тому, що багатовимірні масиви у Java реалізуються як множина одновимірних масивів, стає можливим реалізувати багатовимірні масиви з різною кількістю елементів у межах виміру. Синтаксис оголошення зубчатого масиву нічим не відрізняється від синтаксису оголошення звичайного багатовимірного масиву. Різниця є лише у способі ініціалізації, де використовується виділення пам’яті під різну кількість елементів у межах виміру.

Синтаксис оголошення та ініціалізації зубчатого масиву:

тип[][] змінна = new тип[N][];

змінна[0] = new тип[розмір\_виміру\_20];

змінна[1] = new тип[розмір\_виміру\_21];

...

змінна[N-1] = new тип[розмір\_виміру\_2N-1];

**Особливості використання масивів**

Розмір масиву зберігається у властивості length.

Копіювання масивів не можна здійснити звичайним присвоюванням однієї змінної- масиву іншій. У цьому випадку обидві змінні-масиви посилатимуться на одну і ту саму область пам’яті, тобто фізично на один і той самий масив. Для коректного копіювання масивів слід скористатися методом copyOf класу Arrays. Цей метод створює копію масиву, що переданий через перший параметр методу, у пам’яті та повертає посилання на нього. Кількість елементів масиву, що підлягають копіюванню, передається через другий параметр методу. Приклад копіювання масиву:

int [] copiedArray = Arrays.copyOf(originalArray, originalArray.length);

Для перевірки масивів на ідентичність використовується метод equals класу Arrays. Цей метод приймає через параметри 2 масиви простих типів, та повертає true, якщо вони ідентичні, та false, якщо ні. Приклад порівняння масивів на ідентичність:

boolean compared = Arrays.equals(array1, array2);

**Завдання:**

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.

1. Автоматично згенерувати документацію до розробленої програми.
2. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

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

5. 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.io.\*;

import java.util.\*;

/\*\*

\* Клас Lab1 реалізує приклад програми до лабораторної роботи №1

\*/

public class Lab1KisilKI\_304

{

/\*\*

\* Статичний метод main є точкою входу в програму

\*

\* @param args

\* @throws FileNotFoundException

\*

\*/

public static void main(String[] args) throws FileNotFoundException

{

int nRows;

char[][] arr;

String filler;

Scanner in = new Scanner(System.in);

File dataFile = new File("MyFile.txt");

PrintWriter fout = new PrintWriter(dataFile);

System.out.print("Enter the size of the square matrix:");

nRows = in.nextInt();

in.nextLine();

/\* створення рядків матриці \*/

arr = new char[nRows][];

for(int i = 0; i < nRows/2; i++)

{

arr[i]= new char[i+1];

}

int c=nRows/2;

/\* створення стовпців матриці для парних і непарних рядків \*/

if (nRows%2==0) {

for(int i = nRows/2; i<nRows;i++)

{

arr[i]=new char[c];

c--;

}

}

else

{

for(int i = nRows/2; i<nRows;i++)

{

arr[i]=new char[c+1];

c--;

}

}

System.out.print("\nEnter a placeholder character: ");

filler = in.nextLine();

exit:

/\* заповнення матриці \*/

for (int i=0;i<nRows;i++)

{

for(int k = 0; k<nRows-arr[i].length;k++) {

System.out.print(" ");

}

for (int j=0; j<arr[i].length;j++)

{if(filler.length() == 1)

{

arr[i][j] = (char) filler.codePointAt(0);

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

fout.print(arr[i][j] + " ");

}

else if (filler.length() == 0)

{

System.out.print("\nNo placeholder character entered");

break exit;

}

else

{

System.out.print("\nToo many placeholder characters");

break exit;

}}

System.out.print("\n");

fout.print("\n");

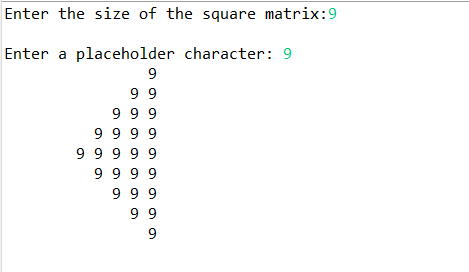
}

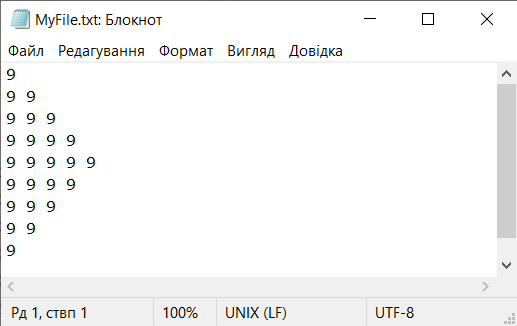
fout.flush();

fout.close();

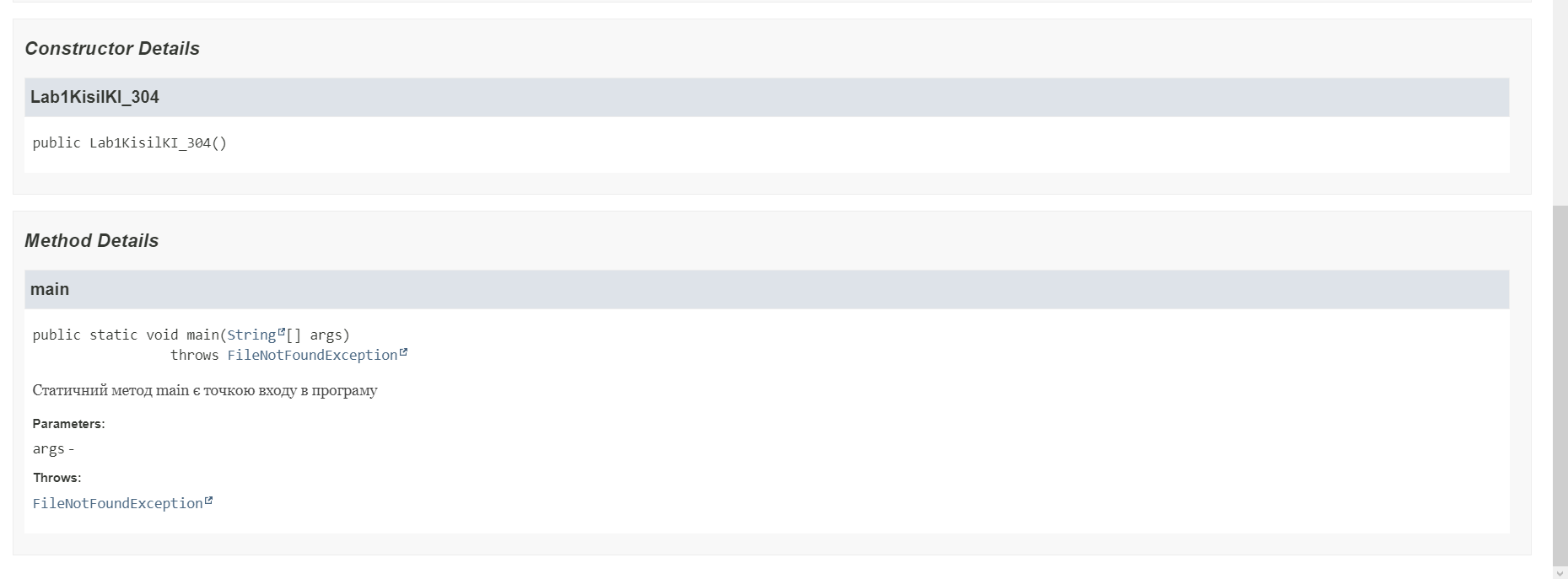
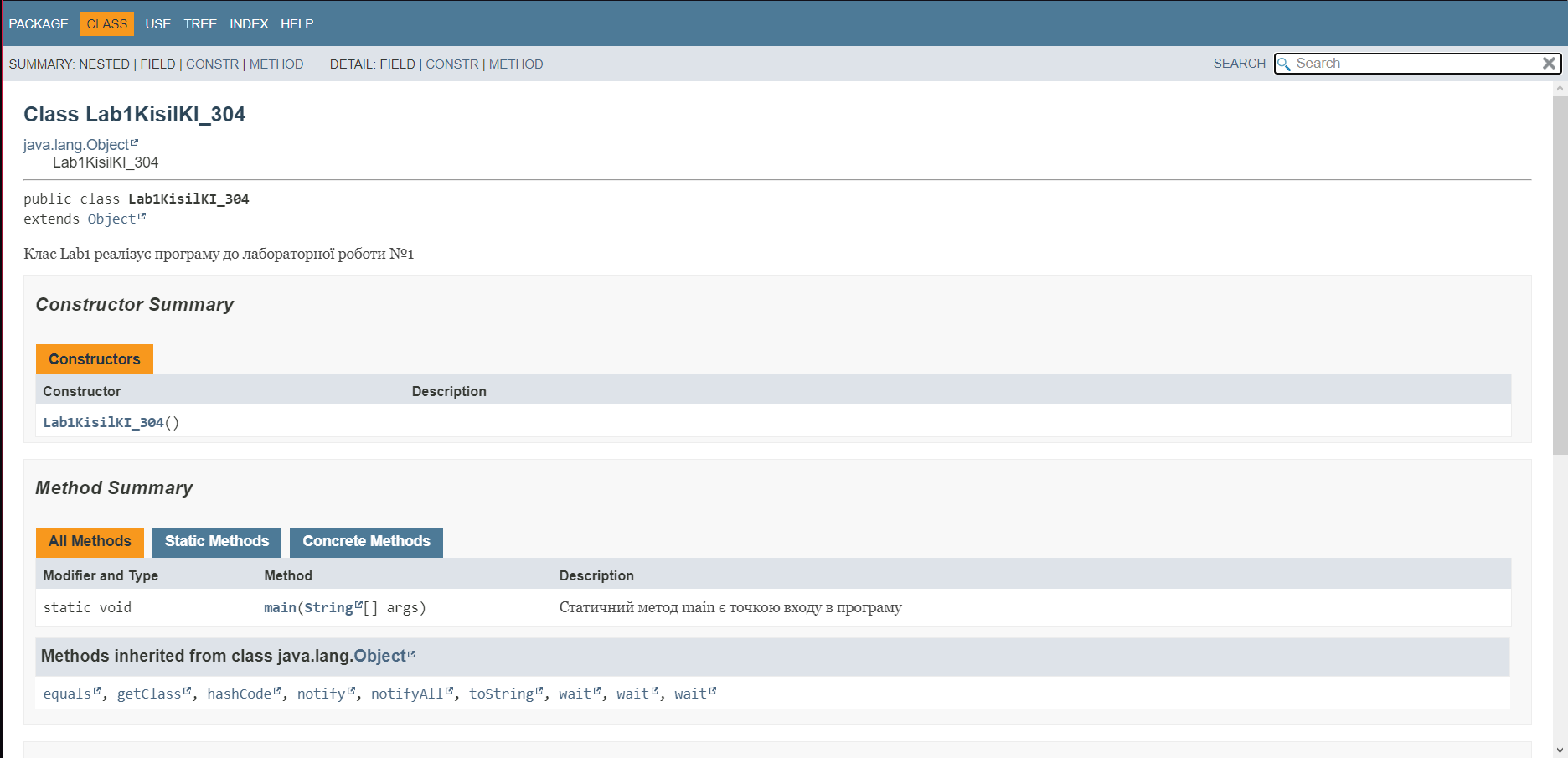
}

}**Результат роботи програми:**

****

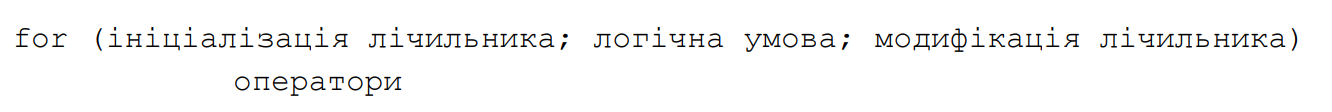


**Фрагмент згенерованої документації**

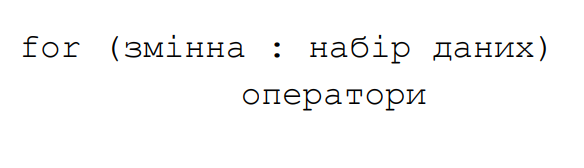
****

**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.