Міністерствоосвіти і науки

Національний університет „Львівська політехніка”

**Кафедра ЕОМ**

****

**Звіт про виконання лабораторної роботи №1**

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Дослідження базових конструкцій мови JAVA»

**Виконала:** ст. групи КІ-305

Циніцька Олена

**Прийняв:** доцент каф. ЕОМ

Іванов Ю. С.

**Львів – 2023**

**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.

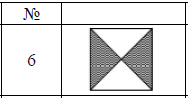
**2.** Автоматично згенерувати документацію до розробленої програми.

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

**4.** Дати відповідь на контрольні запитання:

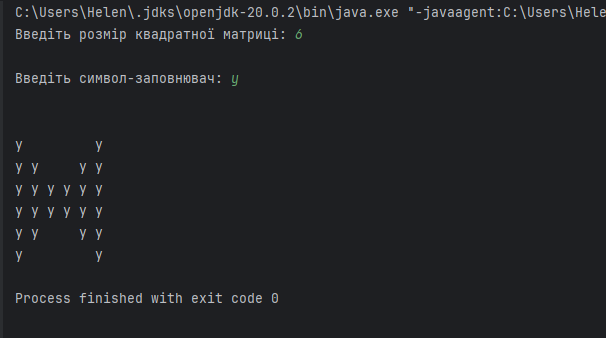
1. Як автоматично згенерувати документацію?
2. В чому різниця між різними варіантами оператора for?

Оскільки мій варціант за списком 26, то виконую варіант завдання №6:



**Код програми:**

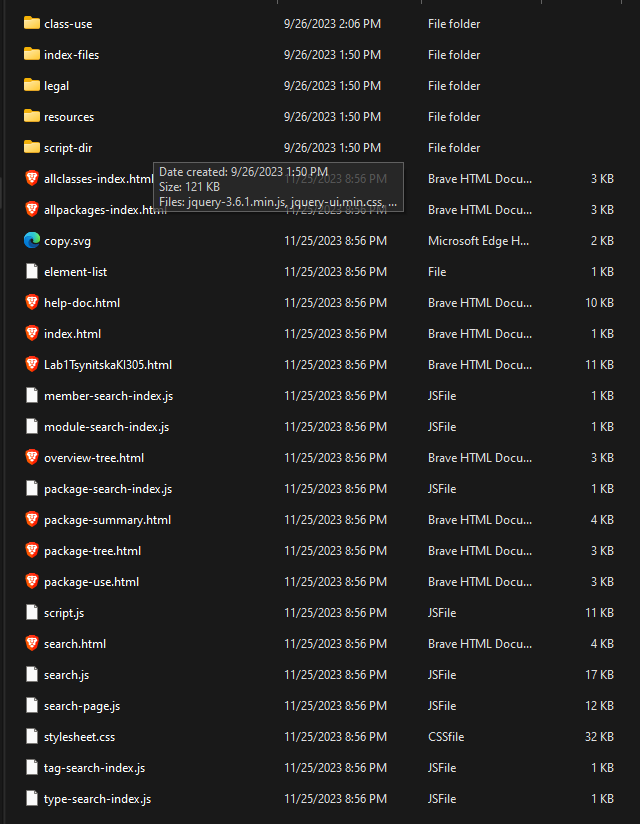
import java.io.\*;  
import java.util.\*;  
  
  
public class Lab1TsynitskaKI305{  
 */\*\*  
 \* The static main method is the entry point into the program  
 \*  
 \* @param args  
 \* @throws FileNotFoundException  
 \*  
 \*/* public static void main(String[] args) throws FileNotFoundException  
 {  
 String filler;  
 Scanner in = new Scanner(System.*in*);  
 File dataFile = new File("MyFile.txt");  
 PrintWriter fout = new PrintWriter(dataFile);  
  
 System.*out*.print("Введіть розмір квадратної матриці: ");  
 char[][] arr = new char[in.nextInt()][];  
 in.nextLine();  
  
 System.*out*.print("\nВведіть символ-заповнювач: ");  
 filler = in.nextLine();  
  
  
 if(filler.length() > 1)  
 {  
 System.*out*.print("\nВи ввели більше ніж один символ!\nЗавершення програми.");  
 fout.print("\nВи ввели більше ніж один символ!\nЗавершення програми.");  
 *closeProgram*(fout);  
 }  
 if(filler.isEmpty())  
 {  
 System.*out*.print("\nВи не ввели символ-заповнювач!\nЗавершення програми.");  
 fout.print("\nВи не ввели символ-заповнювач!\nЗавершення програми.");  
 *closeProgram*(fout);  
 }  
  
 System.*out*.print("\n");  
 fout.print("\n");  
  
 int num;  
 for(int i = 0; i < arr.length; i++)  
 {  
 arr[i] = new char[((i + 1)\*2 < arr.length) ? ((i + 1)\*2) :  
 ((i + 1)\*2 > arr.length + 2) ? ((arr.length - i)\*2) : arr.length];  
 }  
 System.*out*.print("\n");  
  
 for(int i = 0; i < arr.length; i++)  
 {  
 for(int j = 0; j < arr.length; j++)  
 {  
 num = 0;  
 if( (arr[i].length/ 2 > j) || (arr.length / 2 == i) ||  
 ((arr[i].length/ 2 + j >= arr.length)))  
 {  
 if (j < arr[i].length / 2) {  
 arr[i][j] = (char) filler.codePointAt(0);  
 }  
 System.*out*.print(arr[i][num] + " ");  
 fout.print(arr[i][num] + " ");  
 num++;  
 }  
 else  
 {  
 System.*out*.print(" ");  
 fout.print(" ");  
 }  
 }  
 System.*out*.print("\n");  
 fout.print("\n");  
 }  
 fout.flush();  
 fout.close();  
 }  
  
 private static void closeProgram(PrintWriter fout) {  
 fout.flush();  
 fout.close();  
 System.*exit*(0);  
 }  
}

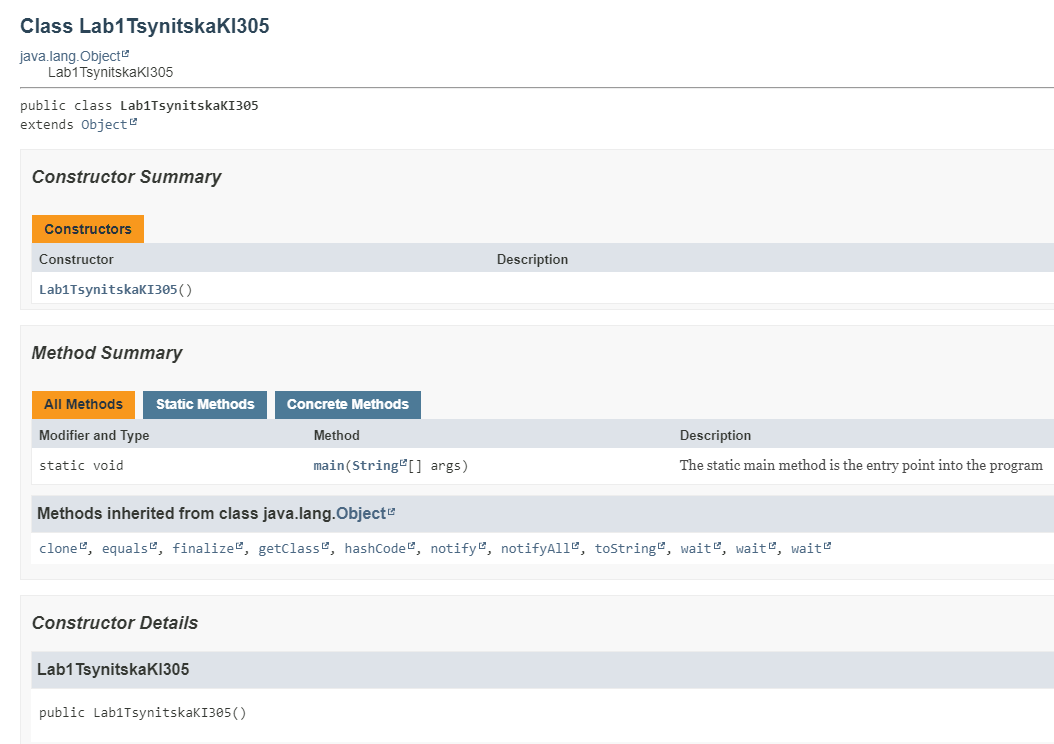
**Результат виконання програми:** 

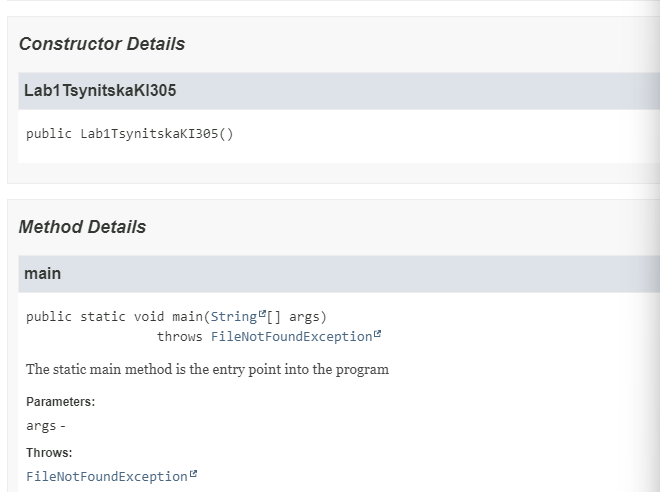
**Згенерований MyFile.txt з матрицею:**



Папка javadoc:

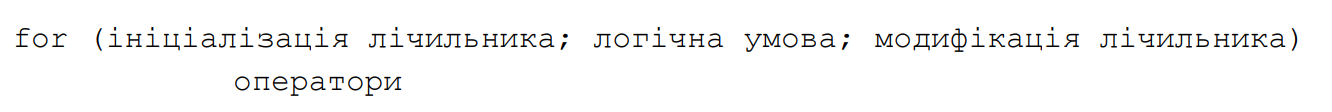


**Згенерована документація: **

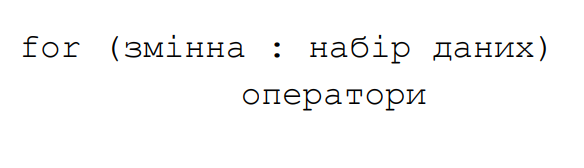


**Відповіді на контрольні запитання:**

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:**

Ознайомилась з базовими конструкціями мови Java та оволоділа навичками написання й автоматичного документування простих консольних програм мовою Java.