**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



Звіт

до лабораторної роботи № 2

з дисципліни: «Кросплатформні засоби програмування»

«ОСНОВИ РОЗРОБКИ ПРОГРАМ МОВОЮ JAVA»

Варіант №3

Виконав:

ст.гр. КІ-34

Гугель Р.Р

Прийняв:

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:**

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.

**Теоретичний матеріал:**

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. Дескриптор, на відміну від решти коментарів, починається з символу @ за яким слідує ім’я дескриптора. Оскільки документація генерується у форматі \*.html, то між /\*\* і \*/ допускається розташування html-тегів, включаючи рисунки. Для автоматичної генерації документації між /\*\* і \*/ можна розмістити:

• коментарі до класу;

• коментарі до методів;

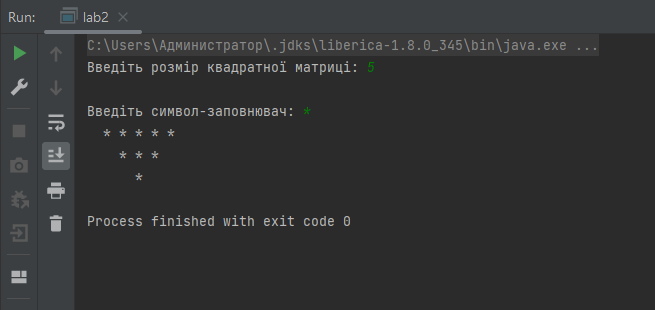
• коментарі до полів;

• загальні коментарі.

**Код програми :**

import java.io.\*;  
import java.util.\*;  
  
public class lab2 {  
  
 public static void main(String[] args) throws FileNotFoundException {  
 int nRows;  
 int SIZE;  
 int k = 0;  
 char[][] arr;  
 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: ");  
 SIZE = in.nextInt();  
 in.nextLine();  
 nRows = SIZE - (SIZE / 2);  
 arr = new char[nRows][];  
 for (int i = 0; i < nRows; i++) {  
 arr[i] = new char[SIZE - k];  
 k += 2;  
 }  
 k = 1;  
 int counter = 0;  
 boolean run = true;  
 System.*out*.print("\nВведіть символ-заповнювач: ");  
 filler = in.nextLine();  
  
 exit:  
 for (int i = 0; i < nRows; i++) {  
 while(run){  
 System.*out*.print(" ");  
 ++counter;  
 if(counter == k){  
 ++k;  
 counter = 0;  
 run = false;  
 }  
 }  
 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("\nНе введено символ заповнювач");  
 break exit;  
 } else {  
 System.*out*.print("\nЗабагато символів заповнювачів");  
 break exit;  
 }  
 }  
 System.*out*.print("\n");  
 fout.print("\n");  
 run = true;  
 }  
 fout.flush();  
 fout.close();  
  
 }  
}

**Результат виконання програми:**



**Висновок :** Я оволодів основами розробки програм мовою Java.