**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



Звіт

до лабораторної роботи № 1

з дисципліни: «Кросплатформні засоби програмування»

«Дослідження базових конструкцій мови Java»

Варіант №16

Виконав:

ст.гр. КІ-34

Палій Н.С.

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.

1. Автоматично згенерувати документацію до розробленої програми.
2. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
3. Дати відповідь на контрольні запитання.

****

**Теоретичний матеріал:**

Автоматичне документування 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.

Коментарі до класу мають бути розміщені після директив import безпосередньо перед визначенням класу. Найчастіше цей коментар має вигляд одного або кількох коротких речень: /\*\* Об’єкт класу Person описує особу. Особа має властивості: ім’я, прізвище та стать./ Коментарі до методів розташовуються безпосередньо перед методами, які вони описують. Крім дескрипторів загального призначення для коментування методів використовуються дескриптори:

• @param змінна опис Цей дескриптор додає в опис методу розділ “parameters”. Опис цього елементу може складатися з кількох рядків та містити html-теги. Всі дескриптори @param, що відносяться до одного методу слід групувати разом.

• @return опис Цей дескриптор додає в опис методу розділ “returns”. Опис цього елементу може складатися з кількох рядків та містити html-теги.

• @throws опис\_класу

**Код програми :**

import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.Scanner;  
import java.util.Arrays;  
import java.io.PrintWriter;  
  
  
  
*/\*\*  
 \* Клас Lab2PaliiKI34 релізує лабораторну роботу №2  
 \*  
 \** ***@author*** *chemp  
 \** ***@version*** *1.0  
 \** ***@since*** *version 1.0  
 \*  
 \*/*public class Lab2PaliiKI34  
{  
 */\*\*  
 \* Статичний метод main є точкою входу в програму  
 \*  
 \** ***@param*** *args  
 \** ***@throws*** *FileNotFoundException  
 \*  
 \*/* public static void main(String[] args) throws FileNotFoundException {  
  
  
 File file = new File ("FileOut.txt");  
 try (PrintWriter pw = new PrintWriter(file)) {  
  
 System.out.println("Введіть розмір мтариці");  
 Scanner SizeOfArr = new Scanner(System.in);  
 int size = (SizeOfArr.nextInt()) - 2;  
  
  
 String symbol;  
 System.out.println("Введіть елемент заповнення масиву");  
 Scanner scanSymbol = new Scanner(System.in);  
 symbol = scanSymbol.nextLine();  
 char symbol2 = symbol.charAt(0);  
  
  
 if (symbol.length() != 1) { // шлях при некоректному вводі  
 System.out.println("Некоректний ввід");  
 System.exit(0);  
 }  
  
  
 char arr2[][] = new char[size][]; // створення зубчастої матриці  
 for (int i = 0; i < size; i++) {  
 if (i == 0 || i == (size - 1)) {  
 arr2[i] = new char[size];  
 } else {  
 arr2[i] = new char[2];  
 }  
 }  
  
  
 for (int i = 0; i < size; i++) { //запис зубчастої матриці  
 for (int j = 0; j < size; j++) {  
 if (i == 0 || i == (size - 1)) {  
 arr2[i][j] = symbol2;  
 } else if (j < 2) {  
 arr2[i][j] = symbol2;  
 }  
 }  
 }  
  
 for (char[] row : arr2) { // вивід в консоль  
 System.out.println(Arrays.toString(row));  
 }  
  
  
 for (char[] row : arr2) { // вивід в файл  
 pw.println(Arrays.toString(row));  
 }  
 pw.close();  
 }  
 }  
}

**Результат виконання програми:**

Вивід в консоль

Зображення, що містить текст

Автоматично згенерований опис

Вивід в файл

Зображення, що містить текст, екран

Автоматично згенерований опис

Робота програми при некоректному вводі

Зображення, що містить текст

Автоматично згенерований опис

ДокументаціяЗображення, що містить текст

Автоматично згенерований опис

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