

Міністерство освіти і науки України
Національний університет «Львівська політехніка»



Звіт

до лабораторної роботи №2

З дисципліни: «Кросплатформенні засоби програмування»

На тему: «ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ JAVA»

Виконав:

Студент групи КІ-34

Романів В. А.

Прийняв:

Іванов Ю. С.

Львів 2022

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

Виконання роботи

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 введенні 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.

Завдання: вивести зубчатий масив, у якому заповнені лише заштриховані області відповідно до варіанту

ВАРІАНТИ ЗАВДАНЬ

№		№		№		№	
1		6		11		16	
2		7		12		17	
3		8		13		18	
4		9		14		19	
5		10		15		20	

Код програми:

```
import java.io.*;
import java.util.*;

/**
 * Клас Lab2 реалізує приклад програми до лабораторної роботи №2
 *
 * @author Romaniv Vitalii
 * @version 1.0
 * @since version 1.0
 */

public class Lab2RomanivKI34 {

    /**
     * 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("Введіть розмір квадратної матриці: ");
        nRows = in.nextInt();
        in.nextLine();
        arr = new char[nRows][];

        /**
         * Перевірка на довжину сторони квадрата
         * 6 10 14 18 22 26 30 ... повертає 1
         * 5 7 9 11 13 15 17 ... повертає 1
         * 8 12 16 20 24 28 32 ... повертає 0
         */
        int type = (nRows / 2) % 2;

        switch (type)
        {
            case 0:
                int first_layer = nRows / 4;
                int three_layer = nRows - first_layer - 1;
                for (int i = 0; i < nRows; ++i)
                {
```

```

        if(i < first_layer)
            arr[i] = new char[nRows];
        else if(i == first_layer)
            arr[i] = new char[first_layer * 2];
        else if(i > first_layer && i < three_leyer)
            arr[i] = new char[nRows - 2];
        else if(i == three_leyer)
            arr[i] = new char [first_layer * 2];
        else if(i > three_leyer && i < nRows)
            arr[i] = new char [nRows];
    }
    System.out.print("\nВведіть СИМВОЛ-заповнювач: ");
    filler = in.nextLine();
    in.close();

    if(filler.length() == 0)
    {
        System.out.print("\nНе введено СИМВОЛ заповнювач");
        break;
    }else if(filler.length() > 1) {
        System.out.print("\nЗабагато СИМВОЛІВ заповнювачів");
        break;
    }

    for(int i = 0; i < nRows; ++i){
        for(int j =0; j < arr[i].length; ++j){
            arr[i][j] = (char)filler.charAt(0);
        }
    }
    for(int i = 0; i < nRows; ++i){
        for(int j = 0; j < arr[i].length; ++j){
            if(i < first_layer) {
                System.out.print(arr[i][j] + " ");
                fout.print(arr[i][j] + " ");
            }
            else if(i == first_layer || i == three_leyer){
                if(j == (arr[i].length / 2)){
                    int length_s = (nRows - (arr[i].length)) * 2;
                    boolean run = true;
                    while (run)
                    {
                        if(length_s == 0)
                            break;
                        System.out.print(" ");
                        fout.print(" ");
                        --length_s;
                    }
                }
                System.out.print(arr[i][j] + " ");
                fout.print(arr[i][j] + " ");
            }
            else if(i > first_layer && i < three_leyer)
            {
                if(j == first_layer || j == (nRows - first_layer
- 2)) {
                    System.out.print(" ");
                    fout.print(" ");

```

```

        }
        System.out.print(arr[i][j] + " ");
        fout.print(arr[i][j] + " ");
    }
    else if(i > three_leyer && i < nRows) {
        System.out.print(arr[i][j] + " ");
        fout.print(arr[i][j] + " ");
    }
}
System.out.print("\n");
fout.print("\n");
}
fout.flush();
fout.close();
break;

case 1:
    int yellow_length = (nRows / 2) - 1;
    int size_yell = nRows / 4; // границя першого слою
    int seredina = nRows / 2 - 1; // кількість жовтих квадратиків
між якими пусто другий слой

    int tritina = nRows / 3; //
    int next_chastina = nRows - tritina; // третій слой границя
ЖОВТИХ

    int yellow_square_in_the_chenter = nRows - 2; // центральні
ЖОВТІ ШТУКИ

    for (int i = 0; i < nRows; ++i)
    {
        if (i < size_yell)
            arr[i] = new char[nRows];
        else if (i == size_yell)
            arr[i] = new char[seredina];
        else if (i > size_yell && i < next_chastina)
            arr[i] = new char[yellow_square_in_the_chenter];
        else if (i == next_chastina)
            arr[i] = new char[seredina];
        else if (i > next_chastina && i < nRows)
            arr[i] = new char[nRows];
    }
    System.out.print("\nВведіть СИМВОЛ-заповнювач: ");
    filler = in.nextLine();
    in.close();

    if(filler.length() == 0)
    {
        System.out.print("\nНе введено СИМВОЛ заповнювач");
        break;
    }else if(filler.length() > 1) {
        System.out.print("\nЗабагато СИМВОЛІВ заповнювачів");
        break;
    }

    for(int i = 0; i < nRows; ++i){
        for(int j =0; j < arr[i].length; ++j){
            arr[i][j] = (char)filler.charAt(0);

```

```

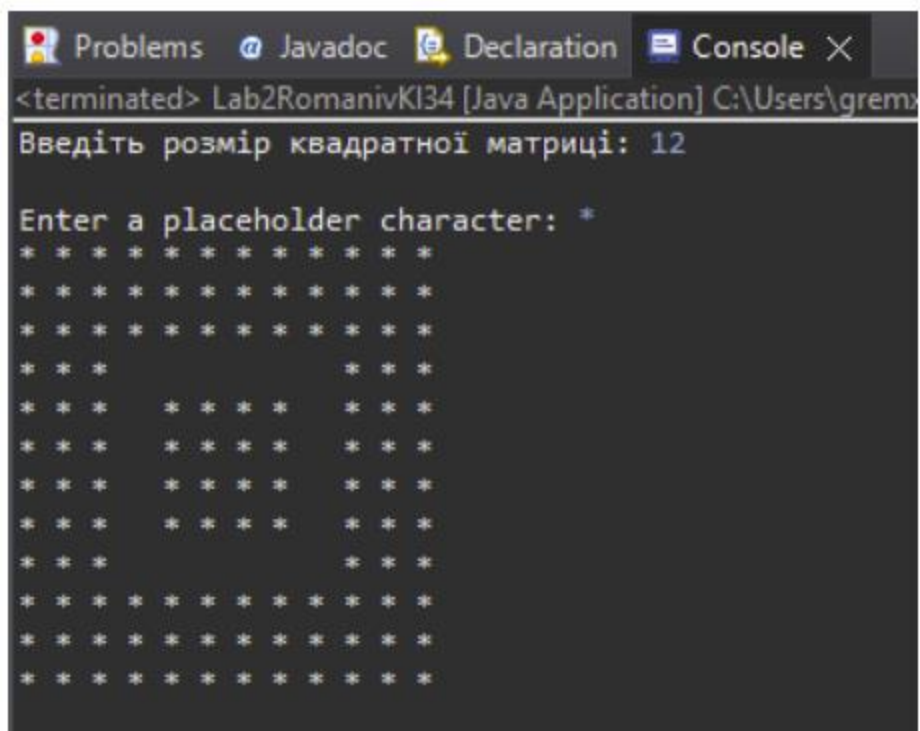
    }
}

for(int i = 0; i < nRows; ++i){
    for(int j =0; j < arr[i].length; ++j){
        if(i < size_yell) {
            System.out.print(arr[i][j] + " ");
            fout.print(arr[i][j] + " ");
        }
        else if(i == size_yell || i == next_chastina){
            if(j == (arr[i].length / 2 )){
                int length_s = (nRows - seredina) * 2;
                boolean run = true;
                while (run)
                {
                    if(length_s == 0)
                        break;
                    System.out.print(" ");
                    fout.print(" ");
                    --length_s;
                }
                System.out.print(arr[i][j] + " ");
                fout.print(arr[i][j] + " ");
            }
            else if((i > size_yell && i < next_chastina)){
                int inducator = seredina / 2;
                if (j == inducator || j == (nRows - inducator -
2)) {
                    System.out.print(" ");
                    fout.print(" ");
                }
                System.out.print(arr[i][j] + " ");
                fout.print(arr[i][j] + " ");
            }
            else if (i > next_chastina && i < nRows) {
                System.out.print(arr[i][j] + " ");
                fout.print(arr[i][j] + " ");
            }
        }
        System.out.print("\n");
        fout.print("\n");
    }
    fout.flush();
    fout.close();
    break;
default:
    break;
}

}
}

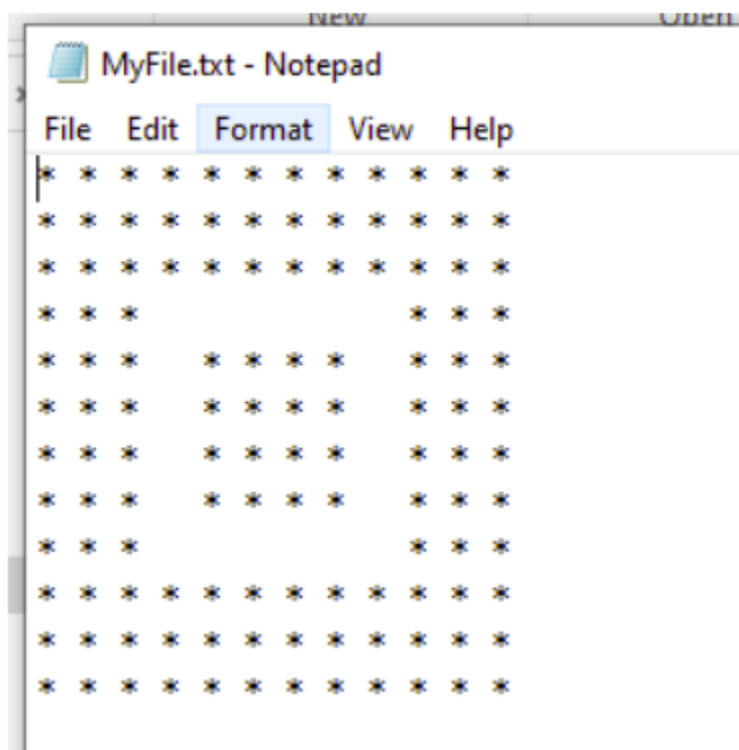
```

Результат роботи програми:



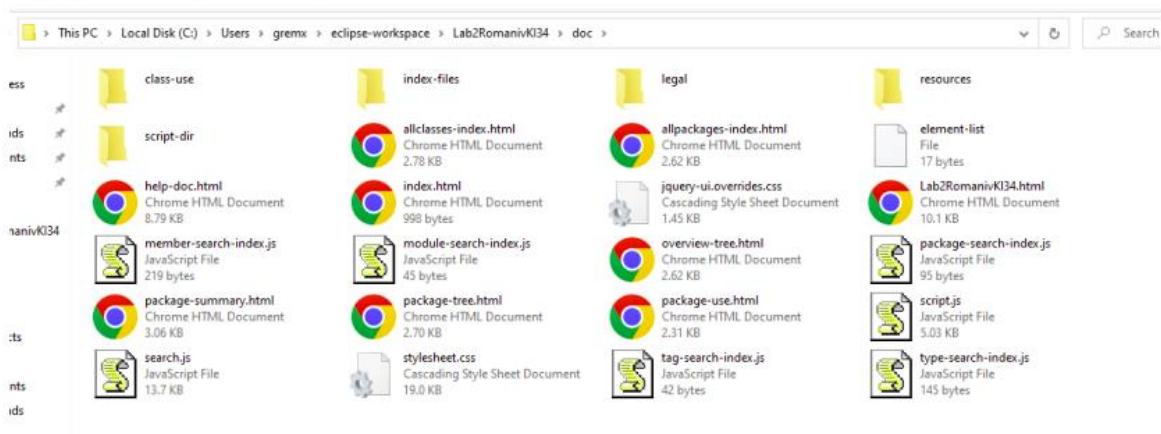
```
<terminated> Lab2RomanivKI34 [Java Application] C:\Users\gremx
Введіть розмір квадратної матриці: 12
Enter a placeholder character: *
* * * * *
* * * * *
* * * * *
* * *           * * *
* * *   * * * *   * * *
* * *   * * * *   * * *
* * *   * * * *   * * *
* * *   * * * *   * * *
* * *           * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Створено текстовий файл:



```
MyFile.txt - Notepad
File Edit Format View Help
* * * * *
* * * * *
* * * * *
* * *           * * *
* * *   * * * *   * * *
* * *   * * * *   * * *
* * *   * * * *   * * *
* * *   * * * *   * * *
* * *           * * *
* * * * *
* * * * *
* * * * *
```

Сформована java-документація:



Відповіді на контрольні запитання

1. `/** text @ ... */`
2. До методів застосовуються коментарі такі ж, як і до класів(зап. 1)
3. Для генерування документації по пакету слід ввести в консолі ОС Windows: `javadoc -d каталог_doc ім'я_пакету` Опція `-d каталог_doc` задає каталог, де слід розмістити згенеровану документація до пакету.
4. Double, int, char, float, byte, short, Boolean, long.
5. `int array[] = new int[size]; double array[][] = new double[size][size];`
6. цикл for та for в стилі foreach, перевірка умови оператор if else if, оператори переривання break та continue.
7. for (ініціалізація лічильника; логічна умова; модифікація лічильника) оператори)

Робота оператора циклу for в стилі C/C++ починається з виконання операторів поля ініціалізації лічильника, після чого відбувається перевірка логічної умови, виконання операторів тіла циклу та модифікація лічильника. Після першої ітерації, поки логічний вираз є істинним, циклічно послідовно виконуються лише операції перевірки умови, тіла циклу та модифікації лічильника. Область видимості змінних, що оголошені в полі ініціалізації лічильника та час їх життя обмежені тілом циклу for. 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:

for (змінна : набір даних)

оператори

При опрацюванні циклу змінній послідовно присвоюється кожен елемент набору даних (наприклад, елемент масиву) після чого виконується оператор.

8. Для введення інформації з консолі необхідно створити об'єкт класу Scanner і зв'язати його з стандартним потоком вводу System.in.

9. Популярним механізмом виводу на консоль є використання методу print об'єкту out з пакету System, який виводить переданий через параметр текстовий рядок на екран.

10. Для введення інформації з файлу необхідно підключити пакет java.io та створити об'єкт класу Scanner з об'єкту File: `Scanner fin = new Scanner(File("MyFile.txt"));`

Висновок: написано програму виведення зубчатого масиву, освоєно створення документації у тому числі атоматичної.