

**Міністерство освіти і науки України**  
**Національний університет «Львівська політехніка»**  
**Кафедра ЕОМ**



**Звіт**  
**до лабораторної роботи № 2**  
**з дисципліни: «Кросплатформні засоби програмування»**  
**«Дослідження базових конструкцій мови Java»**  
**Варіант №22**

Виконав:  
ст.гр. КІ-34  
Трач В.І.  
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 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.

Варіант : 22(2)



## Теоретичний матеріал:

Автоматичне документування При автоматичній генерації документації використовується утиліта `javadoc`, яка аналізує вміст між `/ i /` та на його базі генерує документацію у форматі `.html`. Коментарі між `/ i /` 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 `/ i */` допускається розташування `html`-тегів, включаючи рисунки. Для автоматичної генерації документації між `/ i /` 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.*;
import java.util.Scanner;
import java.util.Arrays;
import java.io.IOException;

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

public class Lab2Trach34 {
    /**
     * Статичний метод main є точкою входу в програму
     *
     * @param args
     * @throws FileNotFoundException
     */
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter size of matrix : ");
        int size = scan.nextInt();

        String s;
        System.out.print("Enter symbol-filler : ");
        s = scan.next();
        char symbol = s.charAt(0);

        if (s.length() != 1){
            System.out.println("Error! Symbol filler is longer/shorter than 1");
            System.exit(0);
        }
    }
}
```

```

    }

    char [][] arr1 = new char[size][size];

    for (int i = 0; i < arr1.length; i++) {
        for (int j = 0; j < arr1.length; j++) {
            if(arr1.length % 2 == 0) {
                if (i < arr1.length / 2) {
                    if (j <= i) {
                        arr1[i][j] = symbol;
                        arr1[arr1.length - j - 1][arr1.length - i - 1] = symbol;
                    }
                }
            }
            else {
                if (i <= arr1.length / 2) {
                    if (j <= i) {
                        arr1[i][j] = symbol;
                        arr1[arr1.length - j - 1][arr1.length - i - 1] = symbol;
                    }
                }
            }
        }
    }

    char [][] arr2 = new char [size][];

    for (int i = 0; i < size; i++){
        arr2[i] = new char[i + 1];
        if(i >= size / 2){
            arr2[i] = new char[i - size / 2 + 1];
        }
    }

    for (int i = 0; i < size; i++) {
        if (i >= size / 2) {
            for (int j = 0; j <= i - size / 2; j++) {
                arr2[i][j] = symbol;
            }
        } else {
            for (int j = 0; j <= i; j++) {
                arr2[i][j] = symbol;
            }
        }
    }

    try {
        File file = new File("Lab_2.txt");
        if(!file.exists()) {
            file.createNewFile();
        }
        PrintWriter pw = new PrintWriter(file);
        pw.println("Enter size of matrix : " + size);
        pw.println("Enter symbol-filler : " + symbol);
        pw.println("Ragged array : ");
        System.out.println("Ragged array :");

        for (int i = 0; i < size; i++) {
            if (i < size / 2) {
                for (int j = 0; j <= i; j++) {
                    pw.print(arr2[i][j] + "\t");
                    System.out.print(arr2[i][j] + "\t");
                }
                pw.print("\n");
                System.out.print("\n");
            } else {
                pw.print("\t".repeat(size/2));
                System.out.print("\t".repeat(size/2));
                for (int j = 0; j <= i - size / 2; j++) {
                    pw.print(arr2[i][j] + "\t");
                    System.out.print(arr2[i][j] + "\t");
                }
                pw.print("\n");
                System.out.print("\n");
            }
        }
        pw.close();
    } catch (IOException e){
        System.out.print("Error! : " + e);
    }

```

```
}  
}  
}
```

## Результат виконання програми:

Вивід в консоль :

```
Enter size of matrix : 13  
Enter symbol-filler : x  
Ragged array :  
x  
x  x  
x  x  x  
x  x  x  x  
x  x  x  x  x  
x  x  x  x  x  x  
  
      x  
      x  x  
      x  x  x  
      x  x  x  x  
      x  x  x  x  x  
      x  x  x  x  x  x  
      x  x  x  x  x  x  x
```

Вивід в файл :

```
Lab2Trach34.java x Lab_2.txt x
1 Enter size of matrix : 13
2 Enter symbol-filler : x
3 Ragged array :
4 x
5 x x
6 x x x
7 x x x x
8 x x x x x
9 x x x x x x
10
11 x x
12
13 x x x
14 x x x x x
15 x x x x x x
16 x x x x x x x
```

## Документація

**Class Lab2Trach34**  
java.lang.Object<sup>®</sup>  
Lab2Trach34

public class Lab2Trach34  
extends Object<sup>®</sup>

Клас Lab2 реалізує приклад програми до лабораторної роботи №2

Since:  
version 1.0

**Constructor Summary**

**Constructors**

Constructor	Description
Lab2Trach34()	

**Method Summary**

**All Methods** **Static Methods** **Concrete Methods**

Modifier and Type	Method	Description
static void	main(String <sup>®</sup> [] args)	Статичний метод main є точкою входу в програму

**Methods inherited from class java.lang.Object<sup>®</sup>**

clone<sup>®</sup>, equals<sup>®</sup>, finalize<sup>®</sup>, getClass<sup>®</sup>, hashCode<sup>®</sup>, notify<sup>®</sup>, notifyAll<sup>®</sup>, toString<sup>®</sup>, wait<sup>®</sup>, wait<sup>®</sup>, wait<sup>®</sup>

**Constructor Details**

**Lab2Trach34**

public Lab2Trach34()

**Method Details**

**main**

public static void main(String<sup>®</sup>[] args)

Статичний метод main є точкою входу в програму

Parameters:  
args -

Throws:  
FileNotFoundException<sup>®</sup>

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