

Міністерство освіти і науки України
Національний університет “Львівська політехніка”

Кафедра ЕОМ



Звіт

до лабораторної роботи №1

З дисципліни: «Кросплатформні засоби програмування»

На тему: «ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ JAVA»

Виконала: ст. гр. КІ-305

Романюк М. Р.

Прийняв: доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2023

Мета: 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 Lab1ПрізвищеГрупа;
 - 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. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.
5. Дати відповідь на контрольні запитання.

Мій варіант №2 (22):



Код програми:

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

public class Matrix {

    private static final int MIN_MATRIX_SIZE = 3;
    private static final String OUTPUT_FILE = "Result.txt";

    public static void main(String[] args) {
        try (Scanner scanner = new Scanner(System.in);
            BufferedWriter writer = new BufferedWriter(new
FileWriter(OUTPUT_FILE))) {

            int size = getInputSize(scanner);
            String fillCharacter = getInputFillCharacter(scanner);

            int rowWithExtraChar = getRowWithExtraCharInput(scanner, size);
            String extraChar = getExtraCharInput(scanner, rowWithExtraChar);

            char[][] matrix = generateMatrix(size, fillCharacter,
rowWithExtraChar, extraChar);
            displayMatrix(matrix);
            saveMatrixToFile(matrix, writer);

        } catch (IOException e) {
```

```

        System.err.println("An error occurred while writing to the file: " +
e.getMessage());
    }
}

private static int getRowWithExtraCharInput(Scanner scanner, int size) {
    int row = 0;
    while (true) {
        try {
            System.out.print("Enter row number: ");
            row = scanner.nextInt();
            if (row == 0 || row > size) {
                System.out.println("The row number must be between 1 and " +
size);
                continue;
            }
            break;
        } catch (InputMismatchException e) {
            System.out.println("Invalid input. Please enter an integer.");
            scanner.nextLine(); // Consume the invalid input
        }
    }
    return row;
}

private static String getExtraCharInput(Scanner scanner, int rowWithExtraChar)
{
    String extraharacter;

    while (true) {
        System.out.print(String.format("Enter character to fill %d row: ",
rowWithExtraChar));
        extraharacter = scanner.next();
        if (extraharacter.length() != 1) {
            System.out.println("A single character can be consumed.");
        } else {
            break;
        }
    }
    return extraharacter;
}

private static int getInputSize(Scanner scanner) {
    int size = 0;
    while (true) {
        try {
            System.out.print("Enter size: ");
            size = scanner.nextInt();
            if (size <= MIN_MATRIX_SIZE) {
                System.out.println("The size of the matrix must be greater
than " + MIN_MATRIX_SIZE);
                continue;
            }
            break;
        } catch (InputMismatchException e) {
            System.out.println("Invalid input. Please enter an integer.");
            scanner.nextLine(); // Consume the invalid input
        }
    }
    return size;
}

private static String getInputFillCharacter(Scanner scanner) {
    String fillCharacter;

    while (true) {
        System.out.print("Enter fill character: ");
        fillCharacter = scanner.next();
        if (fillCharacter.length() != 1) {

```

```

        System.out.println("Fill character must be a single character.");
    } else {
        break;
    }
}
return fillCharacter;
}

private static char[][] generateMatrix(int size, String fillCharacter, int
rowWithExtraChar, String extraChar) {
    char[][] matrix = new char[size][];
    int ceil = (int) Math.ceil(size / 2.0);
    for (int i = 0; i < ceil; i++) {
        matrix[i] = new char[i + 1];
        Arrays.fill(matrix[i], i == rowWithExtraChar - 1 ? extraChar.charAt(0)
: fillCharacter.charAt(0));
        if (size % 2 != 0) {
            int floor = (int) Math.floor(size / 2.0);
            matrix[i + floor] = new char[i];
            Arrays.fill(matrix[i + floor], i + floor == rowWithExtraChar - 1 ?
extraChar.charAt(0) : fillCharacter.charAt(0));
        } else {
            matrix[i + ceil] = new char[i + 1];
            Arrays.fill(matrix[i + ceil], i + ceil == rowWithExtraChar - 1 ?
extraChar.charAt(0) : fillCharacter.charAt(0));
        }
    }
    return matrix;
}

private static void displayMatrix(char[][] matrix) {
    for (int i = 0; i < matrix.length; i++) {
        if (i >= (int) Math.ceil(matrix.length / 2.0)) {
            for (int k = (int) Math.ceil(matrix.length / 2.0); k != 0; k--) {
                System.out.print("\t");
            }
        }
        for (int j = 0; j < matrix[i].length; j++) {
            System.out.print(matrix[i][j]);
            System.out.print("\t");
        }
        System.out.println("");
    }
}

private static void saveMatrixToFile(char[][] matrix, BufferedWriter writer)
throws IOException {
    for (int i = 0; i < matrix.length; i++) {
        if (i >= (int) Math.ceil(matrix.length / 2.0)) {
            for (int k = (int) Math.ceil(matrix.length / 2.0); k != 0; k--) {
                writer.write("\t");
            }
        }
        for (int j = 0; j < matrix[i].length; j++) {
            writer.write(matrix[i][j]);
            writer.write("\t");
        }
        writer.newLine();
    }
    writer.flush();
    System.out.println("Matrix saved to " + OUTPUT_FILE);
}
}

```

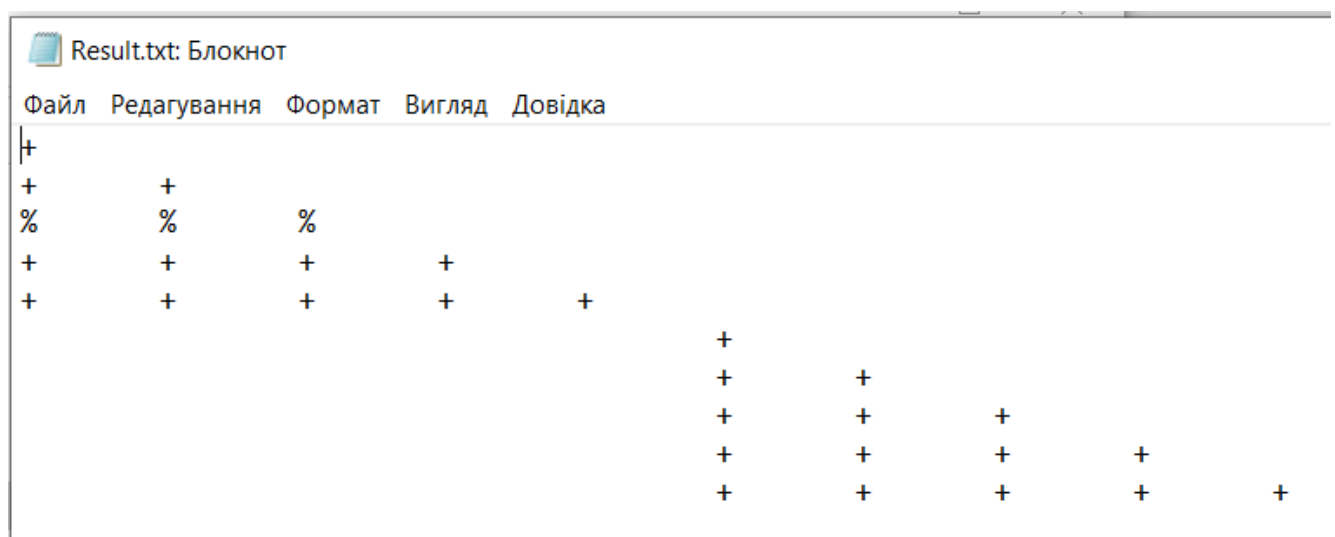
Результати роботи програми:

```
Enter size: 10
Enter fill character: +
Enter row number: 3
Enter character to fill 3 row: %























+
+   +
%   %   %
+   +   +   +
+   +   +   +   +

               +
               +   +
               +   +   +
               +   +   +   +
               +   +   +   +   +

Matrix saved to Result.txt
```



Згенерована документація:

	allclasses-index.html	26.09.2023 16:15	Chrome HTML Do...	4 КБ
	allpackages-index.html	26.09.2023 16:15	Chrome HTML Do...	3 КБ
	copy.svg	26.09.2023 16:15	Microsoft Edge HT...	1 КБ
	element-list	26.09.2023 16:15	Файл	1 КБ
	help-doc.html	26.09.2023 16:15	Chrome HTML Do...	9 КБ
	index.html	26.09.2023 16:15	Chrome HTML Do...	1 КБ
	LAB1.iml	26.09.2023 16:05	Файл IML	1 КБ
	Main.html	26.09.2023 16:15	Chrome HTML Do...	11 КБ
	member-search-index.js	26.09.2023 16:15	JavaScript File	1 КБ
	module-search-index.js	26.09.2023 16:15	JavaScript File	1 КБ
	overview-tree.html	26.09.2023 16:15	Chrome HTML Do...	3 КБ
	package-search-index.js	26.09.2023 16:15	JavaScript File	1 КБ
	package-summary.html	26.09.2023 16:15	Chrome HTML Do...	4 КБ
	package-tree.html	26.09.2023 16:15	Chrome HTML Do...	3 КБ
	Result.txt	26.09.2023 16:14	Текстовий докум...	0 КБ
	script.js	26.09.2023 16:15	JavaScript File	10 КБ
	search.html	26.09.2023 16:15	Chrome HTML Do...	4 КБ
	search.js	26.09.2023 16:15	JavaScript File	16 КБ
	search-page.js	26.09.2023 16:15	JavaScript File	11 КБ
	stylesheet.css	26.09.2023 16:15	Каскадна таблиц...	32 КБ
	tag-search-index.js	26.09.2023 16:15	JavaScript File	1 КБ
	type-search-index.js	26.09.2023 16:15	JavaScript File	1 КБ

PACKAGE

CLASS

TREE

INDEX

HELP

PACKAGE: DESCRIPTION | RELATED PACKAGES | CLASSES AND INTERFACES

Unnamed Package

Classes

Class	Description
Main	Головний клас програми для генерації та відображення матриці із заповненими символами.

```
public class Main
extends Object
```

Головний клас програми для генерації та відображення матриці із заповненими символами.

Constructor Summary

Constructors

Constructor	Description
Main()	

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
static void	main(String[] args)	Головний метод програми.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Details

Main

```
public Main()
```

Відповіді на контрольні запитання

1. Які дескриптори використовуються при коментуванні класів?

Відповідь: Для коментування класів в Java використовують дескриптори JavaDoc, такі як `/** ... */`.

2. Які дескриптори використовуються при коментуванні методів?

Відповідь: Для коментування методів в Java також використовують дескриптори JavaDoc, а саме `/** ... */`.

3. Як автоматично згенерувати документацію?

Відповідь: Документацію в Java можна автоматично згенерувати за допомогою інструменту JavaDoc, запускаючи його з командного рядка з вказанням шляху до вашого коду.

4. Які прості типи даних підтримує Java?

Відповідь: Java підтримує такі прості типи даних: `int`, `double`, `float`, `char`, `boolean`, `byte`, `short`, `long`.

5. Як оголосити змінну-масив?

Відповідь: Змінну-масив в Java оголошують, вказавши тип даних, за яким слід ім'я змінної та використовуючи квадратні дужки. Наприклад: `int[] myArray;`

6. Які керуючі конструкції підтримує Java?

Відповідь: Java підтримує такі керуючі конструкції: `if`, `else`, `switch`, `for`, `while`, `do-while`, `break`, `continue`, `return`.

7. В чому різниця між різними варіантами оператора `for`?

Відповідь: У Java існують три варіанти оператора `for`: `for`, `for-each`, і `for-each` для ітерації по колекціях. Різниця між ними полягає в способі ітерації та доступі до елементів.

8. Як здійснити ввід з консолі?

Відповідь: Для вводу з консолі в Java використовують об'єкт `Scanner`. Ви створюєте об'єкт `Scanner` для стандартного вводу, і потім використовуєте методи, такі як `nextInt()`, `nextDouble()`, тощо, для отримання введених даних.

9. Як здійснити ввід з текстового файлу?

Відповідь: Ввід з текстового файлу в Java здійснюється за допомогою класу `FileReader` або `BufferedReader`, які дозволяють читати дані з файлу рядок за рядком.

10. Як здійснити запис у текстовий файл?

Відповідь: Запис у текстовий файл в Java здійснюється за допомогою класу `FileWriter` або `BufferedWriter`, які дозволяють записувати дані у файл рядок за рядком.

Висновок: у ході виконання лабораторної роботи я ознайомилася з базовими конструкціями мови Java та оволоділа навичками написання й автоматичного документування простих консольних програм мовою Java.