

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра «Електронних обчислювальних машин»



Звіт
з лабораторної роботи № 1
з дисципліни: «Кросплатформенні засоби програмування»
на тему: «Дослідження базових конструкцій мови Java»

Виконав:

студент групи КІ-306

Чаус Б.В.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

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

Завдання (варіант № 24)

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 (рис. 1);

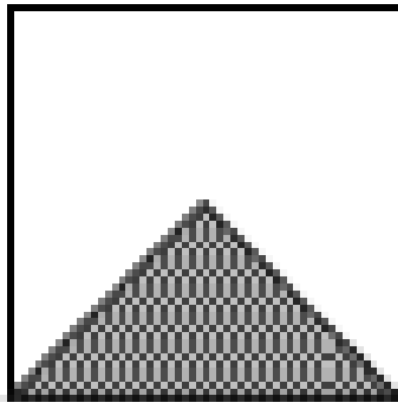


Рис. 1. Заштрихована область квадратної матриці.

- 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 відповіді на контрольні запитання:

- які дескриптори використовуються при коментуванні класів?
- які дескриптори використовуються при коментуванні методів?

- як автоматично згенерувати документацію?
- які прості типи даних підтримує java?
- як оголосити змінну-масив?
- які керуючі конструкції підтримує java?
- в чому різниця між різними варіантами оператора for?
- як здійснити ввід з консолі?
- як здійснити ввід з текстового файлу?
- як здійснити запис у текстовий файл?

Вихідний код програми

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

/**
 * The Lab1ChausKI306 class implements an example program for laboratory work #1.
 *
 * @author Chaus Bohdan
 * @version 1.0
 * @since version 1.0
 */
public class Lab1ChausKI306 {
    /**
     * Generates a jagged array with a specified size and symbol.
     *
     * @param size The size of the square matrix.
     * @param symbol The character used to fill the matrix.
     * @return The jagged array.
     */
    public static char[][] generateJaggedArr(int size, char symbol) {
        char[][] jaggedArr = new char[size][];

        for (int i = 0; i < size; i++) {
            jaggedArr[i] = new char[i + 1];
            Arrays.fill(jaggedArr[i], ' ');
        }

        for (int i = size / 2; i < size; i++) {
            int spaces = Math.abs(i - size + 1);
            for (int j = spaces; j < (size - spaces); j++) {
                jaggedArr[i][j] = symbol;
            }
        }

        return jaggedArr;
    }

    /**
     * The entry point of the program.
     *
     * @param args The command-line arguments (not used).
     * @throws FileNotFoundException Exception thrown if the file is not found.
     */
    public static void main(String[] args) throws FileNotFoundException {
```

```

int nRows;           // Number of rows in the matrix
char[][] arr;        // Jagged array
char filler = ' ';   // Character for filling the matrix
Scanner in = new Scanner(System.in); // Object for reading input

// Prompt the user for the size of the square matrix
System.out.print("Enter size of square matrix: ");
nRows = in.nextInt();
in.nextLine(); // Clear the buffer

// Prompt the user for the character to fill the matrix
System.out.print("Enter character to fill matrix: ");
filler = in.next().charAt(0);

if (filler != ' ') { // If a character is specified
    arr = generateJaggedArr(nRows, filler); // Generate the matrix
    printArray(arr); // Print the matrix to the screen
    writeToTextFile(arr); // Write the matrix to a file
}
else if (filler == ' ') {
    System.out.println("\nNo character entered placeholder");
}
else {
    System.out.println("\nToo many characters fillers");
}
}

/**
 * Prints the jagged array to the screen.
 *
 * @param arr The jagged array.
 */
public static void printArray(char[][] arr) {
    for (char[] row : arr) {
        for (char c : row) {
            System.out.print(c + " ");
        }
        System.out.println();
    }
}

/**
 * Writes the jagged array to a text file.
 *
 * @param arr The jagged array.
 */
public static void writeToTextFile(char[][] arr) {
    try (PrintWriter fout = new PrintWriter("MyFile.txt")) {
        for (char[] row : arr) {
            for (char c : row) {
                fout.print(c + " ");
            }
            fout.println();
        }
        System.out.println("\nData written to MyFile.txt");
    } catch (FileNotFoundException e) {
        System.out.println("Error: File not found");
    }
}
}

```

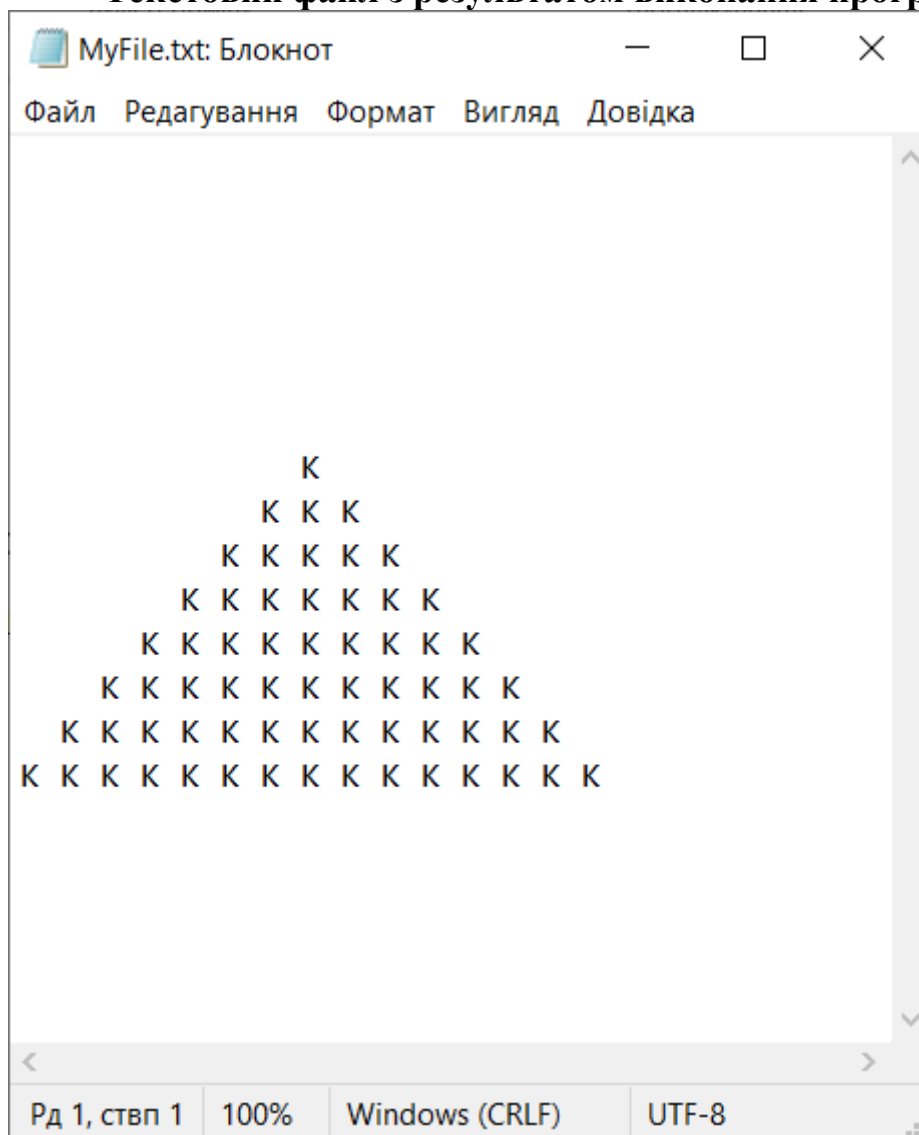
Результат виконання програми

Enter size of square matrix: 15
Enter character to fill matrix: K

```
      К
    К К К
  К К К К К
К К К К К К К
К К К К К К К К
К К К К К К К К К
К К К К К К К К К К
К К К К К К К К К К К
```

Data written to MyFile.txt

Текстовий файл з результатом виконання програми



Фрагмент згенерованої документації

PACKAGE

CLASS

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHODDETAIL: FIELD | CONSTR | METHOD

SEARCH

Class Lab1ChausKI306

java.lang.Object[®]
Lab1ChausKI306

public class Lab1ChausKI306
extends Object[®]

The Lab1ChausKI306 class implements an example program for laboratory work #1.

Since:
version 1.0

Constructor Summary

Constructors

Constructor	Description
Lab1ChausKI306()	

Method Summary

All MethodsStatic MethodsConcrete Methods

Modifier and Type	Method	Description
static char[][]	generateJaggedArr(int size, char symbol)	Generates a jagged array with a specified size and symbol.
static void	main(String[] args)	The entry point of the program.
static void	printArray(char[][] arr)	Prints the jagged array to the screen.
static void	writeToTextFile(char[][] arr)	Writes the jagged array to a text file.

Відповіді на контрольні запитання

1. Коментування класів: використовуються Javadoc-стиль та блочні коментарі.
2. Коментування методів: також використовуються Javadoc-стиль та блочні коментарі.
3. Генерація документації: за допомогою інструментів, таких як Javadoc, можна автоматично генерувати документацію з коментарів вихідного коду.
4. Прості типи даних в Java: int, double, boolean, char, byte, short, long, float.
5. Оголошення змінної-масиву: тип_даних[] назва_масиву; (наприклад, int[] numbers;)
6. Керуючі конструкції в Java: if-else, switch, for, while, do-while, break, continue, return.
7. Різниця варіантів оператора for:

for (ініціалізація; умова; інкремент) - базовий варіант

for (тип_даних назва : масив) - покращений варіант для ітерації по елементах масиву.

8. Ввід з консолі в Java: Використовуйте клас Scanner для зчитування введення користувача з консолі.
9. Ввід з текстового файлу в Java: Використовуйте клас FileReader та BufferedReader для зчитування даних з файлу.
10. Запис у текстовий файл в Java: Використовуйте клас FileWriter та BufferedWriter для запису даних у файл.

Висновок

Під час лабораторної роботи, я ознайомився з базовими конструкціями мови Java та оволодів навиками написання й автоматичного документування простих консольних програм мовою Java.