

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**Інститут комп'ютерних технологій, автоматики та метрології**  
**кафедра “Електронних обчислювальних машин”**



**Звіт**  
**з лабораторної роботи №1**  
**дисципліни «Кросплатформні засоби програмування»**  
**на тему: «ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ JAVA»**  
**Варіант 26**

**Виконав:**  
студент групи КІ-306  
Тимків С. В.  
**Прийняв:**  
Олексів М. В.

# ЛАБОРАТОРНА РОБОТА №1

## ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ JAVA

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

### Теоретичний матеріал

Мова Java є строго типізованою. Це означає, що тип кожної змінної має бути оголошеним. Мова має 8 основних (простих) типів, які не є класами та однаково представляються на будь-якій машині, де виконується програма.

Тип	Розмір, байти	Діапазон значень	Приклад запису
boolean	1	true, false	true
char	2	\u0000...\uFFFF	\u0041 або 'A'
byte	1	-128...127	15
short	2	-32768...32767	15
int	4	$-2^{31} \dots 2^{31} - 1$	15
long	8	$-2^{63} \dots 2^{63} - 1$	15L
float	4	$\pm 3.4\text{E}+38$	15.0F
double	8	$\pm 1.79\text{E}+308$	15.0 або 15.0D

Синтаксис оголошення змінних:

тип назваЗмінної[=значення] {, назваЗмінної [= значення]};

Наприклад, int i; double x, y; boolean isZero = false;

Перед використанням змінну слід обов'язково ініціалізувати.

Масив – структура даних, що зберігає набір значень однакового типу. Пам'ять під масив виділяється у керованій кучі. При завершенні життєвого циклу масиву пам'ять, яку він займав, вивільняється збирачем сміття. Доступ до елементів масиву здійснюється за допомогою індексів. Індксація масивів у Java починається з 0. Для створення масиву у Java необхідно оголосити змінну-масив та ініціалізувати її. При створенні за допомогою оператора new масиву чисел всі його елементи ініціалізуються нулями (масиви типу boolean ініціалізуються значеннями false, масиви об'єктів ініціалізуються значеннями null). Після створення масиву змінити його розмір неможливо.

Розмір масиву зберігається у властивості `length`. Копіювання масивів не можна здійснити звичайним присвоюванням однієї змінної масиву іншій. У цьому випадку обидві змінні-масиви посилатимуться на одну і ту саму область пам'яті, тобто фізично на один і той самий масив. Для коректного копіювання масивів слід скористатися методом `copyOf` класу `Arrays`. Цей метод створює копію масиву, що переданий через перший параметр методу, у пам'яті та повертає посилання на нього. Кількість елементів масиву, що підлягають копіюванню, передається через другий параметр методу.

### Завдання (Варіант №26)

1) Написати та налагодити програму на мові Java згідно варіанту. Програма має задовольняти наступним вимогам

- програма має розміщуватися в загальнодоступному класі `Lab1Tymkiv306`;
- програма має генерувати зубчатий масив, який міститиме лише заштриховані області квадратної матриці згідно варіанту;

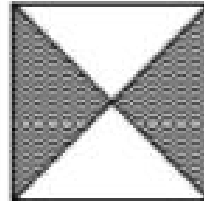


Рис. 1.1. Заштрихована область квадратної матриці.

- 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) 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.

GitHub Repository:

[https://github.com/SofiaTymkiv/CPPT\\_Tymkiv\\_SV\\_KI-36\\_2.git](https://github.com/SofiaTymkiv/CPPT_Tymkiv_SV_KI-36_2.git)

## Виконання завдання

```
package org.example;

import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

/**
 * Клас Lab1Tymkiv306 реалізує лабораторну роботу №1
 * */
public class Lab1Tymkiv306 {
    /**
     * Статичний метод main є точкою входу в програму
     *
     * @param args аргументи
     * */
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Ввід розміру матриці
        System.out.println("Введіть розмір матриці: ");
        int row = sc.nextInt();

        // Ввід символу заповнювача
        System.out.println("Введіть символ заповнювач: ");
        String symbol = sc.next();

        // Перевірка на валідність символу заповнювача
        if (symbol.length() != 1) {
            System.out.println("Введіть коректний символ заповнювач");
            return;
        }

        String[][] arr = createLengthOfEachSubArr(row);
        String fileName = "Lab1.txt";

        // Запуск бізнес логіки для генерації зубчастого масиву (виводу в консоль і
        запису в файл)
        try {
            printMatrix(arr, symbol, row, fileName);
        } catch (IOException e) {
            // Обробка помилки під час запису в файл
            throw new RuntimeException("Сталася помилка під час запису в файл: " +
e.getMessage());
        }
    }

    /**
     * Метод який генерує зубчастий масив (виводить в консоль і записує в файл)
     *
     * @param arr масив для заповнення
     * @param symbol символ заповнювач
     * @param row розмір масиву
     * @param file назва файлу
     * @throws IOException якщо сталась якась помилка при запису в файл
     * */
}
```

```

    public static void printMatrix(String[][] arr, String symbol, int row, String
file) throws IOException {
        System.out.println("Результат матриці: ");

        try (FileWriter writer = new FileWriter(file)) {
            // Логіка для формування зубчастого масиву за варіантом
            for (int i = 0; i < row; i++) {
                int indexJ = 0;

                for (int j = 0; j < row; j++) {
                    if (j <= i && j <= row - i - 1 && j <= row / 2) {
                        arr[i][indexJ] = symbol;

                        // Вивід в консоль і запис в файл
                        writer.write(arr[i][indexJ] + " ");
                        System.out.print(arr[i][indexJ] + " ");

                        indexJ++;
                    } else if (j >= i && j >= row - i - 1 && j >= row / 2) {
                        arr[i][indexJ] = symbol;

                        // Вивід в консоль і запис в файл
                        writer.write(arr[i][indexJ] + " ");
                        System.out.print(arr[i][indexJ] + " ");

                        indexJ++;
                    } else {
                        writer.write(" ");
                        System.out.print(" ");
                    }
                }
                System.out.println();
                writer.write("\n");
            }

            writer.flush();
        }
    }

    /**
     * Метод який знаходить для кожного під масива довжину
     *
     * @param row розмір масива
     * @return String[][]
     */
    public static String[][] createLengthOfEachSubArr(int row) {
        String[][] arr = new String[row][];

        for (int i = 0; i < row; i++) {
            int length = 0;

            for (int j = 0; j < row; j++) {
                if (j <= i && j <= row - i - 1 && j <= row / 2) {
                    length++;
                } else if (j >= i && j >= row - i - 1 && j >= row / 2) {
                    length++;
                }
            }

            arr[i] = new String[length];
        }
    }

```

```
}  
  
    return arr;  
}  
}
```

```
Введіть розмір матриці:  
7  
Введіть символ заповнювач:  
-  
Результат матриці:  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -
```

Рис 1.1 (Знімок консолі)

### Відповіді на контрольні питання

- 1) Які дескриптори використовуються при коментуванні класів? Дескриптори для коментування класів: `/** */` (JavaDoc коментарі).
- 2) Які дескриптори використовуються при коментуванні методів? Дескриптори для коментування методів: `/** */` (JavaDoc коментарі).
- 3) Як автоматично згенерувати документацію? Використовують команду `javadoc -d каталог_doc ім'я_паketу`, яка генерує документацію на основі JavaDoc коментарів.
- 4) Які прості типи даних підтримує Java? `byte`, `short`, `int`, `long`, `float`, `double`, `char`, `Boolean`
- 5) Як оголосити змінну-масив? `тип[] ім'яМасиву`
- 6) Які керуючі конструкції підтримує Java? `if`, `else`, `switch`, `for`, `while`, `do-while`, `break`, `continue`
- 7) В чому різниця між різними варіантами оператора `for`? Класичний `for`: використовується для ітерацій з відомою кількістю повторень. `enhanced for` (або `for-each`): зручний для ітерацій по масивах та колекціях.

- 8) Як здійснити ввід з консолі? `Scanner scanner = new Scanner(System.in);`
- 9) Як здійснити ввід з текстового файлу? `Scanner scanner = new Scanner(new File("filename.txt"));`
- 10) Як здійснити запис у текстовий файл? `FileWriter` або `PrintWrite`

**Висновок:** Отже, під час цієї лабораторної роботи я опанувала основні принципи програмування на Java, ознайомила з її синтаксисом та змогла створити свою першу програму, яка використовує масиви масивів (оскільки в Java немає багатовимірних масивів, як у C++, багатовимірні масиви реалізуються через масиви масивів) для побудови потрібного візерунку.