

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»

Варіант 15

Виконав:  
Ст. групи КІ-34  
Ольховик О.С.

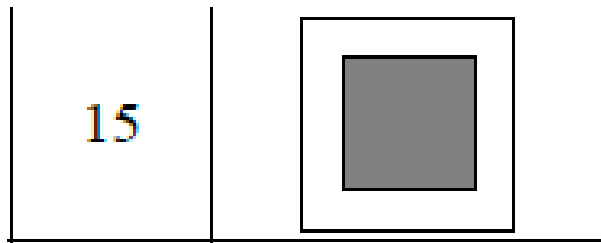
Прийняв:  
к.т.н., доцент  
Іванов Ю.С.

Львів 2022

**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 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.



### Код роботи

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Scanner;

public class lab20lkhovykKI34 {

    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);

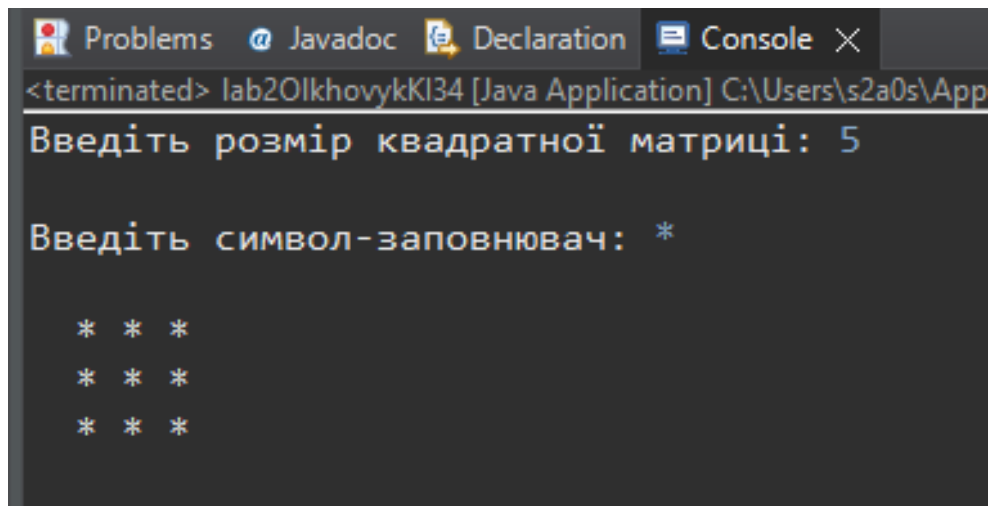
        System.out.print("Введіть розмір квадратної матриці: ");
        nRows = in.nextInt();
        filler = in.nextLine();
        arr = new char[nRows][];
        for (int i = 0; i < nRows; i++) {
            arr[i] = new char[nRows];
        }
    }
}
```

```

    }
    System.out.print("\nВведіть символ-заповнювач: ");
    filler = in.nextLine();
    exit: for (int i = 0; i < nRows; i++) {
        for (int j = 0; j < nRows; j++) {
            if (filler.length() == 1) {
                if ((i > 0 && i < nRows - 1) && (j > 0 && j < nRows - 1))
                    arr[i][j] = (char) filler.charAt(0);
                System.out.print(arr[i][j] + " ");
                fout.print(arr[i][j] + " ");
            } else if (filler.length() == 0) {
                System.out.print("\nНе введено символ заповнювач");
                break exit;
            } else {
                System.out.print("\nЗабагато символів заповнювачів");
                break exit;
            }
        }
        System.out.print("\n");
        fout.print("\n");
    }
    fout.flush();
    fout.close();
}
}

```

## Результат виконання програми



```

<terminated> lab2OlkhovykKI34 [Java Application] C:\Users\s2a0s\AppData\Local\Temp\lab2OlkhovykKI34\
Введіть розмір квадратної матриці: 5
Введіть символ-заповнювач: *
* * *
* * *
* * *

```

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