

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



Звіт  
з лабораторної роботи № 2  
з дисципліни: «Кросплатформенні засоби програмування»  
на тему: «Дослідження базових конструкцій мови Java»

**Виконав:**

студент групи КІ-34

Козлюк Д.С.

**Прийняв:**

доцент кафедри ЕОМ

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 (варіант №8)**

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);

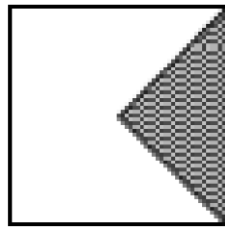


Рис. 1. Заштрихована область квадратної матриці.

- 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?

## Текст програми

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

/**
 * Class Lab2KozliukKI34 that realize tasks from laboratory work 2
 (variant 8)
 *
 * @author Dmytro Kozliuk KI-34
 * @version 1.0
 * @since version 1.0
 */
public class Lab2KozliukKI34
{
    /**
     * The static main method is the entry point to the programs
     *
     * @param args
     * @throws FileNotFoundException
     */
    public static void main(String[] args) throws FileNotFoundException
    {
        int nRows;
        char[][] arr;
        String filler;
        Scanner in = new Scanner(System.in);
        File dataFile = new File("Lab2_Kozliuk.txt");
        PrintWriter fout = new PrintWriter(dataFile);
        System.out.print("Введіть розмір квадратної матриці: ");
        nRows = in.nextInt();
        in.nextLine();
        arr = new char[nRows][];
```

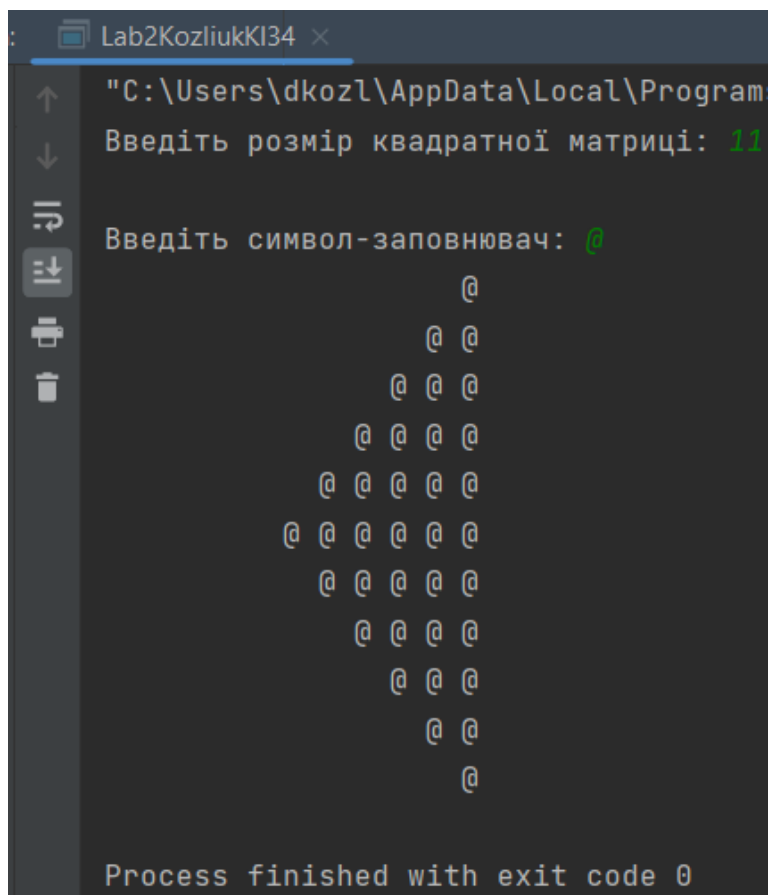
```

//create jagged array
for(int i = 0; i < nRows ; ++i)
    if(i >= nRows/2)
        arr[i] = new char[nRows - i];
    else
        arr[i] = new char[i + 1];

System.out.print("\nВведіть символ-заповнювач: ");
filler = in.nextLine();
exit:
for(int i = 0; i < arr.length; i++)
{
    for(int j = 0; j < arr.length; j++)
    {
        if(filler.length() == 1)
        {
            if (j >= i && j >= arr.length - i - 1)
            {
                arr[i][nRows - j - 1] = (char)
filler.codePointAt(0);
                System.out.print(arr[i][nRows - j - 1] + " ");
                fout.print(arr[i][nRows - j - 1] + " ");
            }
            else
            {
                fout.print(" ");
                System.out.print(" ");
            }
        }
        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();
}

```

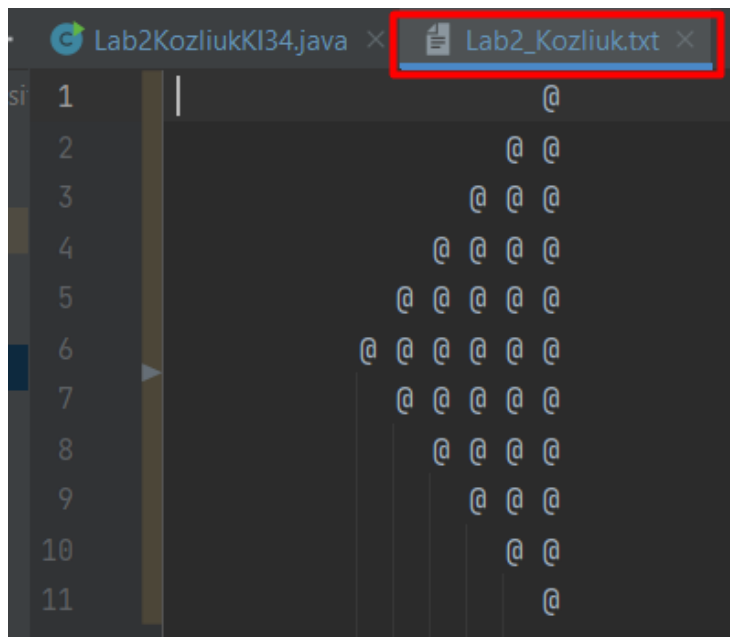
## Результат виконання програми



```
Lab2KozliukKI34 x
"C:\Users\dkozl\AppData\Local\Program
Введіть розмір квадратної матриці: 11
Введіть символ-заповнювач: @
      @
     @@
    @@@
   @@@@
  @@@@@
 @@@@@@
@@@@@@@
 @@@@@@
  @@@@@
   @@@@
    @@@
     @@
      @

Process finished with exit code 0
```

Рис.1.Результат виведення сформованого масиву на екран



```
Lab2KozliukKI34.java x Lab2_Kozliuk.txt x
1      @
2     @@
3    @@@
4   @@@@
5  @@@@@
6 @@@@@@
7 @@@@@@
8 @@@@@@
9  @@@@@
10   @@@@
11    @
```

Рис.2.Результат і виведення сформованого масиву у текстовий файл

## Фрагмент згенерованої документації

PACKAGE

CLASS

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

SEARCH

### Class Lab2KozliukKI34

java.lang.Object<sup>Ⓜ</sup>  
Lab2KozliukKI34

```
public class Lab2KozliukKI34
    extends ObjectⓂ
```

Class Lab2KozliukKI34 that realize tasks from laboratory work 2 (variant 8)

Since:  
version 1.0

#### Constructor Summary

Constructors

Constructor	Description
Lab2KozliukKI34()	

#### Method Summary

All MethodsStatic MethodsConcrete Methods

Modifier and Type	Method	Description
static void	main(String <sup>Ⓜ</sup> [] args)	The static main method is the entry point to the programs

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>

Рис.3. Фрагмент згенерованої документації

## Відповіді на контрольні запитання

1. Які дескриптори використовуються при коментуванні класів?

**@author** *Опис*

**@deprecated** *Опис*

**@param** *Назва параметру Опис*

**@see** *Тема*

**@see** *Пакет. Клас#Член Опис*

**@version** *Версія*

## 2. Які дескриптори використовуються при коментуванні методів?

**@param** змінна опис

**@return** опис

**@throws** опис\_класу

## 3. Як автоматично згенерувати документацію?

Для автоматичної генерації документації використовується утиліта `javadoc`, яка аналізує вміст між `/**` і `*/` та на його базі генерує документацію у форматі `*.html`.

## 4. Які прості типи даних підтримує Java?

- Boolean
- Char
- Byte
- Short
- Int
- Long
- Float
- double

## 5. Як оголосити змінну-масив?

тип[] змінна = new тип[кількість\_елементів\_масиву];

тип[] змінна = {значення1, значення2,..., значенняN};

тип змінна[] = new тип[кількість\_елементів\_масиву];

тип змінна[] = {значення1, значення2,..., значенняN};

## 6. Які керуючі конструкції підтримує Java?

1. Умовний оператор (if)
2. Багатоваріантний вибір (switch)
3. Оператори циклу
  - 3.1. Цикл з передумовою (while)
  - 3.2. Цикл с постумовою (do while)
  - 3.3. Цикл for()
  - 3.4. Цикл for each
4. Оператори переходу: break, continue, return.

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

### 1. Цикл for()

Синтаксис:

```
for ( <ініціалізація>; <умова>; <зміна> )
```

```
<оператор>
```

Вираз <ініціалізація> виконується один раз перед першим витком циклу. Перед кожним витком перевіряється <умова>, і після кожного витка циклу виконується вираз <зміна>. Як і в if() вираз «умова» може бути тільки логічним (типу boolean).

### 2. Цикл for each

Синтаксис:

```
for (<оголошення змінної> : <масив або колекція> )
```

```
<оператор>
```



Це дуже зручна модифікація циклу `for`, яка дозволяє пройти по всіх елементах масиву або колекції без використання лічильника.

#### 7. Як здійснити ввід з консолі?

В Java існує кілька способів реалізації форматowanego введення. Найбільш простий з них – використовувати клас `java.util.Scanner`. Цей клас знаходиться в пакеті `java.util`, тому його необхідно підключити на початку програми:

```
import java.util.Scanner; // Підключаємо клас на початку програми
```

При створенні об'єкт класа `Scanner` зв'язується з потоком введення, файлом чи рядком:

```
Scanner in = new Scanner(System.in); // Стандартний потік введення
```

```
Scanner in1 = new Scanner( "123 + 125" ); // Рядок
```

```
Scanner in2 = new Scanner( new File("a.txt") ); // Файл в директорії/ проекту
```

#### 8. Як здійснити ввід з текстового файлу?

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"));
```

#### 9. Як здійснити запис у текстовий файл?

Для виведення інформації у текстовому вигляді у файл треба підключити пакет `java.io` та створити об'єкт класу `PrintWriter` в конструкторі якого необхідно вказати назву файлу, що відкривається на запис, наприклад:

```
PrintWriter fout = new PrintWriter ("MyFile.txt");
```

## **Висновок**

Під час виконання даної лабораторної роботи я ознайомився з базовими конструкціями мови ви Java та оволодів навиками написання й автоматичного документування простих консольних програм мовою Java. Навчився створювати зубчасті масиви, працювати із керованими конструкціями, вводити дані із клавіатури та робити запис у текстовий файл.