Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ



**Звіт**

з лабораторної роботи №1

з дисципліни: “Кросплатформні засоби програмування”

на тему: “Дослідження базових конструкцій мови Java”

Варіант 8

Виконав:

ст. гр. КІ-305

Шевчук М.О.

Прийняв:

Іванов Ю.С.

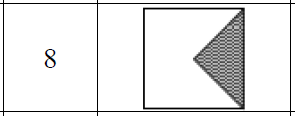
Львів – 2023

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

**Індивідуальне завдання:**

1. Написати та налагодити програму на мові Java згідно варіанту:

2. Автоматично згенерувати документацію до розробленої програми.



**Хід роботи:**

1. Запустив середовище IntelliJ IDEA та написав програму згідно індивідуального завдання:

import java.io.\*;

import java.util.Scanner;

/\*\*

\* Class Lab1ShevchukKI305 implements laboratory work #1

\*

\* @author Shevchouk Maxim

\* @version 1.0

\* @since version 1.0

\*

\*/

public class Lab1ShevchukKI305 {

/\*\*

\* The static main method is the entry point to the program

\*

\* @param args

\* @throws FileNotFoundException

\*

\*/

public static void main(String[] args) throws FileNotFoundException {

try (

Scanner scanner = new Scanner(System.in);// оголошення потоку вводу

FileOutputStream fileOutputStream = new FileOutputStream("Result.txt");// оголошення класу, що відповідає за поток запису байтів в файл

OutputStreamWriter outputStreamWriter = new OutputStreamWriter(fileOutputStream);// оголошення класу, що відповідає за зв'язок символьних і байтових потоків

BufferedWriter bufferedWriter = new BufferedWriter(outputStreamWriter)// оголошення буферу

) {

int n;

do {

System.out.println("Введіть розмір квадратної матриці: ");

n = scanner.nextInt();

} while (n % 2 == 0);

char[][] arr = new char[n][];

int number1 = 1;

int number2 = n / 2;

for (int i = 0; i < n; i++) {

if (i > n / 2) {

arr[i] = new char[number2];

number2--;

} else {

arr[i] = new char[number1];

number1++;

}

}

System.out.print("Введіть символ-заповнювач: ");

char symbol = scanner.next().charAt(0);

// Заповнення матриці символами

for (int i = 0; i < n; i++) {

for (int j = 0; j < arr[i].length; j++) {

arr[i][j] = symbol;

}

}

// Виведення матриці зі зсувом до правого краю

for (int i = 0; i < n; i++) {

for (int k = 0; k < n - arr[i].length; k++) {

System.out.print("\t");

bufferedWriter.write("\t");

}

for (int j = 0; j < arr[i].length; j++) {

System.out.print(arr[i][j] + "\t");

bufferedWriter.write(arr[i][j]+"\t");

}

System.out.println();

bufferedWriter.write("\n");

}

} catch (Exception e) {

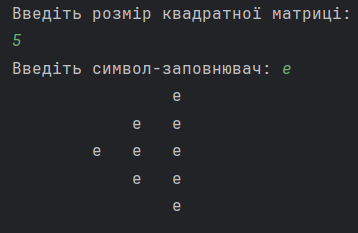
e.printStackTrace();

}

}

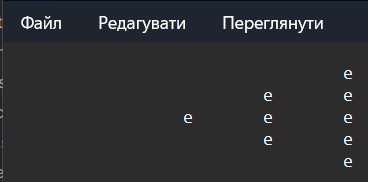
}

1. Запустив програму, використовуючи інтегроване середовище:



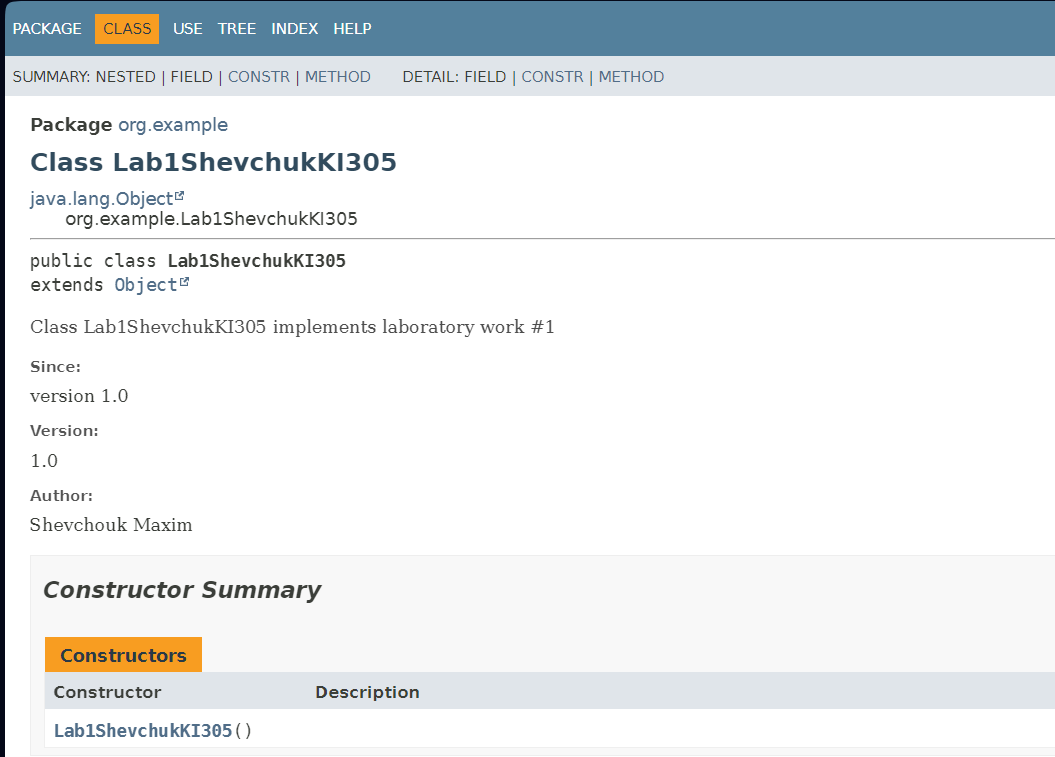
*Рис.1. Результат виконання програми*

1. Після виконання програми переглянув створений файл Result.txt:



*Рис.2. Результат виконання програми у текстовому файлі*

1. Згенерував документацію.



*Рис.3. Фрагмент згенерованої документації*

**Контрольні питання**

1. Які дескриптори використовуються при коментуванні класів?

При коментуванні класів в Java використовуються спеціальні дескриптори (теги) у коментарях, які надають інформацію про клас і його частини. Основні дескриптори для коментування класів включають наступні:

/\*\*

\* Цей клас представляє приклад класу в Java.

\* Він має деякі поля і методи для демонстрації.

\*

\* @author Ім'я автора

\* @version 1.0

\*/

public class ExampleClass {

// Поля і методи класу

}

Це JavaDoc коментарі, які використовуються для створення документації API. Вони мають бути розміщені перед класом і можуть містити таку інформацію, як опис класу, автор, версія, параметри конструкторів та методів, інформація про поведінку тощо.

2. Які дескриптори використовуються при коментуванні методів?

Коментарі до методів розташовуються безпосередньо перед методами, які вони описують. Крім дескрипторів загального призначення для коментування методів використовуються дескриптори:

• @param змінна опис. Цей дескриптор додає в опис методу розділ “parameters”. Опис цього елементу може складатися з кількох рядків та містити html-теги. Всі дескриптори @param, що відносяться до одного методу слід групувати разом.

• @return опис. Цей дескриптор додає в опис методу розділ “returns”. Опис цього елементу може складатися з кількох рядків та містити html-теги.

• @throws опис\_класу. Цей дескриптор додає в опис методу інформацію про класи об’єкти яких можуть генеруватися при виключних ситуаціях. Відомості про кожен клас слід описувати в окремому дескрипторі @throws.

3. Як автоматично згенерувати документацію?

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. Які прості типи даних підтримує Java?

Мова має 8 основних (простих) типів, які не є класами та однаково представляються на будь-якій машині, де виконується програма: boolean, char, byte, short, int, long, float, double.

5. Як оголосити змінну-масив?

тип\_даних[] ім'я\_змінної;

де тип\_даних - тип даних, який буде зберігатися в масиві, ім'я\_змінної – ім'я змінної.

Наприклад, int[] numbers;.

6. Які керуючі конструкції підтримує Java?

Умовні конструкції: if, else, else if, switch.

Циклічні конструкції: for, while, do…while.

Керування потоком виконання: break, continue.

Виключення: try, catch, finally.

7. В чому різниця між різними варіантами оператора for?

* for (ініціалізація; умова; інкремент): Традиційний for-цикл з ініціалізацією, умовою і кроком інкременту.
* for (тип\_даних ім'я\_змінної : колекція): Розширений for (for-each) для ітерації через елементи колекції (масиву, списку тощо).
* for (ініціалізація; умова; інкремент) { ... }: Звичайний for з блоком коду, в якому можна виконувати ітерації.  
  8. Як здійснити ввід з консолі?

9. 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"));

10. 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");

Зробивши це ми отримаємо доступ до методів класу PrintWriter, які призначені для виведення даних простих типів і рядків:  
print – виводить значення простих типів і рядків у текстовому вигляді;

write – призначений для виводу даних типу char і String у текстовий файл.

**Висновок:**

На даній лабораторній роботі ознайомився з базовими конструкціями мови Java та оволодів навиками написання й автоматичного документування простих консольних програм мовою Java.