Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ЕОМ

Звіт

до лабораторної роботи № 1

з дисципліни: «Кросплатформні засоби програмування»

на тему: «Дослідження базових конструкцій мови Java»

Варіант №13

Виконав:
ст. гр. КІ-203
Панченко Д. В.

Прийняв:
доцент кафедри ЕОМ
Іванов Ю. С.

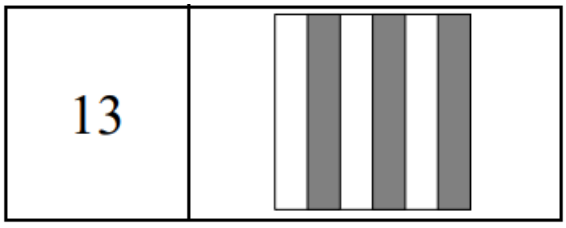
Львів 2023

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 Lab1ПрізвищеГрупа;
    - 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 введені 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.

Варіант завдання:



Вихідний код програми:

```
import java.io.*;
import java.util.*;
/**
* Class Lab1_Panchenko_KI_306 implements the program for laboratory work #1
*/
public class Lab1PanchenkoKI_306
{
/**
* The static main method is the entry point to the program
*
* @param args is arguments
* @throws FileNotFoundException is exception
*
*/

public static void main(String[] args) throws FileNotFoundException
{
int nRows;
char[][] arr;
String filler;
Scanner in = new Scanner(System.in);
File dataFile = new File("Lab-1.txt");
PrintWriter fout = new PrintWriter(dataFile);
System.out.print("Введіть розмір квадратної матриці: ");
nRows = in.nextInt();
in.nextLine();
arr = new char[nRows][];
for(int i = 0; i < nRows; i++)
{
arr[i]= new char[nRows/2];
}
System.out.print("\nВведіть символ-заповнювач: ");
filler = in.nextLine();
exit:
for(int i = 0; i < nRows; i++)
{
for(int j = 0; j < nRows/2; j++)
{
if(filler.length() == 1)
{
arr[i][j] = (char) filler.codePointAt(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;
}
}
if(nRows-nRows/2-nRows/2 == 1)
{
System.out.print("  " );
fout.print("  " );
}
System.out.print("\n");
fout.print("\n");
}
```

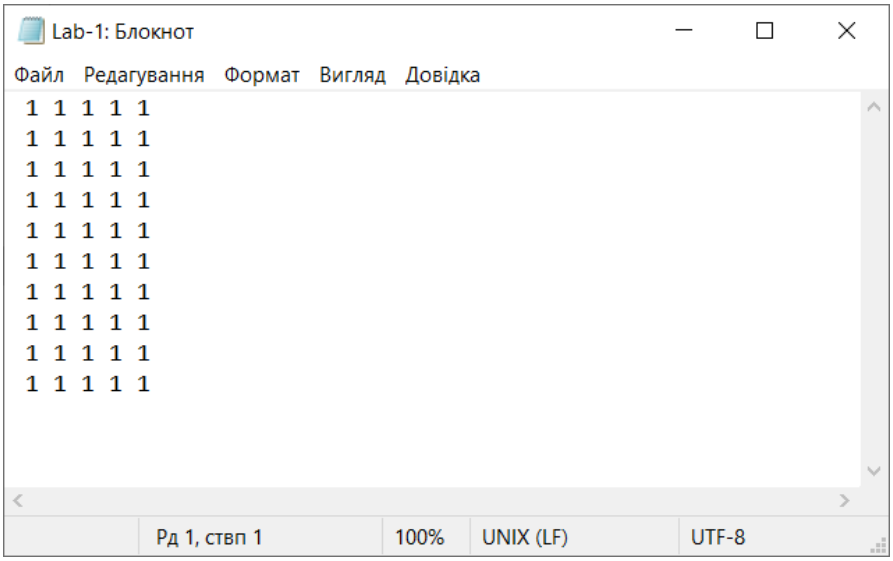
```
fout.flush();
fout.close();
}
}
```

Результат виконання програми:

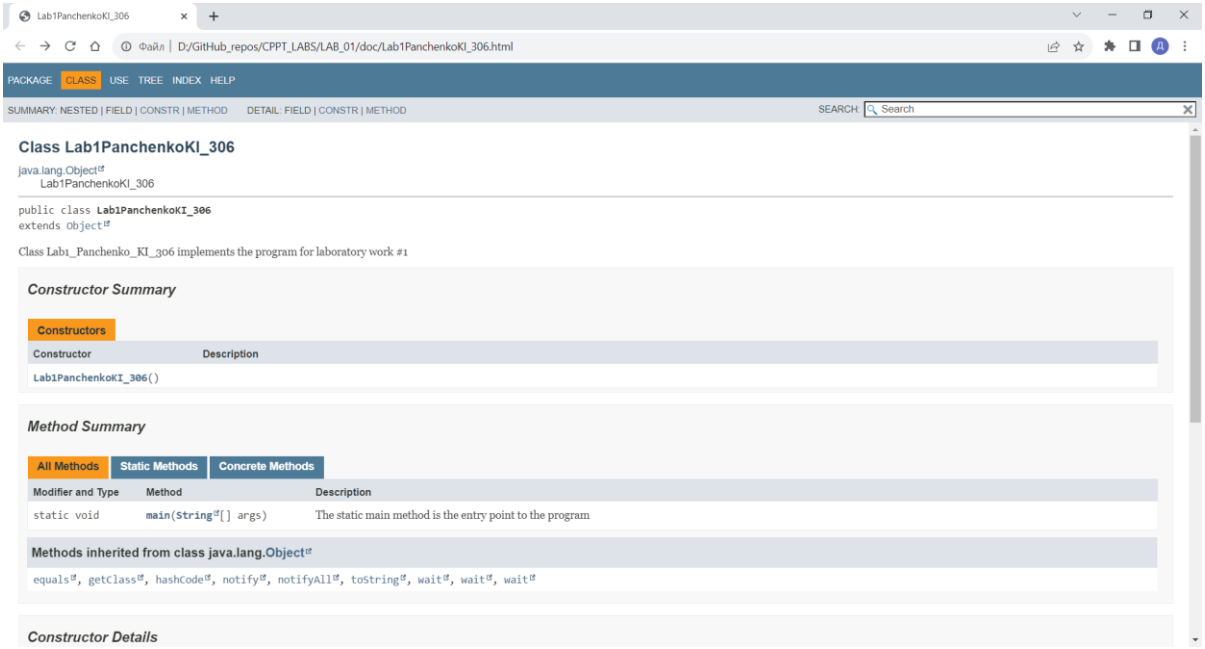
```
Введіть розмір квадратної матриці: 10

Введіть символ-заповнювач: 1
 1 1 1 1 1
 1 1 1 1 1
 1 1 1 1 1
 1 1 1 1 1
 1 1 1 1 1
 1 1 1 1 1
 1 1 1 1 1
 1 1 1 1 1
 1 1 1 1 1
 1 1 1 1 1
```

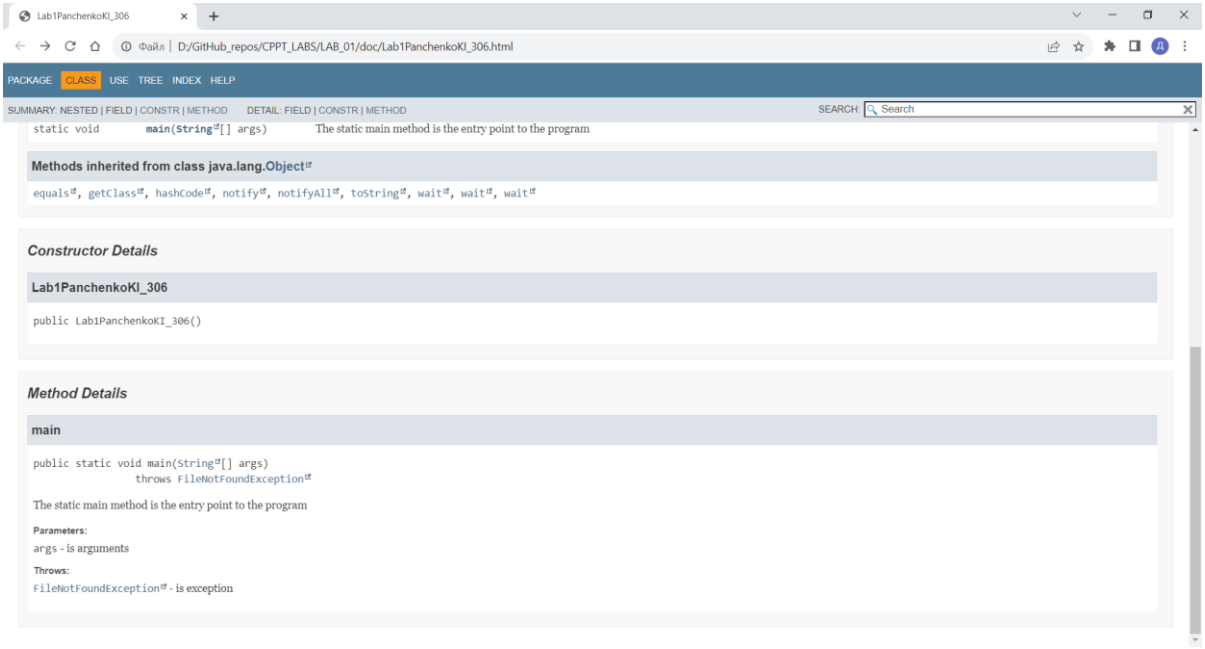
```
Введіть розмір квадратної матриці: 20

Введіть символ-заповнювач: 11

Забагато символів заповнювачів
```

```
Введіть розмір квадратної матриці: 20

Введіть символ-заповнювач:

Не введено символ заповнювач
```



Фрагмент згенерованої документації:





Відповіді на контрольні запитання:

1. Які дескриптори використовуються при коментуванні класів?

Коментарі до класу мають бути розміщені після директив import безпосередньо перед визначенням класу. Найчастіше цей коментар має вигляд одного або кількох коротких речень:
/**
Об’єкт класу <code>Person</code> описує особу.
Особа має властивості: ім’я, прізвище та стать.
*/

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 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	$-2^{31}...2^{31}-1$	15
long	8	$-2^{63}...2^{63}-1$	15L
float	4	±3.4E+38	15.0F
double	8	±1.79E+308	15.0 або 15.0D

5. Як оголосити змінну-масив?

Синтаксиси оголошення неініціалізованого одновимірного масиву:
тип[] змінна;
тип змінна[];
Синтаксиси оголошення неініціалізованого двовимірного масиву:
тип[][] змінна;
тип змінна[][];

6. Які керуючі конструкції підтримує Java?

Основні конструкції мови Java багато в чому співпадають з аналогічними конструкціями мов C/C++. Такі оператори як switch, if-else, while, do-while –ідентичні аналогічним конструкціям у мовах C/C++. Оператор циклу for має деякі особливості. У Java цей оператор має 2 різновиди:
• конструкція в стилі C/C++ з полем ініціалізації, логічною умовою та кроком;
• конструкція з синтаксисом foreach.

7. В чому різниця між різними варіантами оператора for?

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.

8. Як здійснити ввід з консолі?

Для введення інформації з консолі необхідно створити об’єкт класу Scanner і зв’язати його з стандартним потоком вводу System.in, наприклад:
Scanner in = new Scanner(System.in);

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");

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