Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



**Звіт**

з лабораторної роботи №2

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Дослідження базових конструкцій мови Java»

Виконав: ст.гр. КІ-34

Турчин П.Р.

Прийняв:

викл. каф. 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:**

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

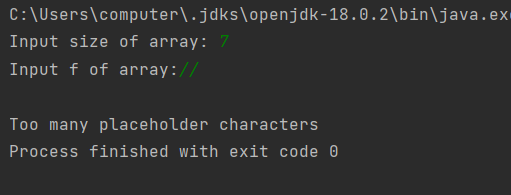
* програма має розміщуватися в загальнодоступному класі Lab2ПрізвищеГрупа;
* програма має генерувати зубчатий масив, який міститиме лише заштриховані області квадратної матриці згідно варіанту;
* розмір квадратної матриці і символ-заповнювач масиву вводяться з клавіатури;
* при не введенні або введенні 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;

1. Автоматично згенерувати документацію до розробленої програми.
2. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
3. Дати відповідь на контрольні запитання.

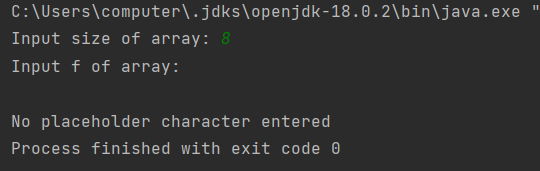
**Лістинг програми:**

import java.io.\*;  
import java.util.\*;  
*/\*\*  
 \* Клас Lab2TurchynKI34 реалізує програму до лабораторної роботи №2  
 \*  
 \** ***@author*** *Pavlo Turchyn  
 \** ***@version*** *1.0  
 \** ***@since*** *version 1.0  
 \*  
 \*/*public class Lab2TurchynKI34 {  
 */\*\*  
 \* Статичний метод main є точкою входу в програму  
 \*  
 \** ***@param*** *args function parameter  
 \** ***@throws*** *FileNotFoundException throw about non-existent file  
 \*  
 \*/* public static void main(String[] args) throws IOException {  
 int arrsize;  
 String[][] arr;  
 String filler;  
  
 File myFile = new File("D:/Lab2TurchynKI34.txt");  
 Scanner myScanFile = new Scanner(myFile);  
 Scanner myScanSys = new Scanner(System.*in*);  
 FileWriter myWriter = new FileWriter(myFile);  
  
 System.*out*.print("Input size of array: ");  
 arrsize = myScanSys.nextInt();  
 myScanSys.nextLine();  
  
  
  
  
 arr = new String[arrsize][arrsize];  
 for (int i = 0; i < arrsize; i++) {  
 for(int j = 0; j<arrsize; j++)  
 arr[i][j] = " ";  
 }  
 System.*out*.print("Input f of array:");  
 filler = myScanSys.nextLine();  
  
 if (filler.length() == 0) {  
 System.*out*.print("\nNo placeholder character entered");  
 System.*exit*(0);  
 }  
 if (filler.length() > 1) {  
 System.*out*.print("\nToo many placeholder characters");  
 System.*exit*(0);  
 }  
  
 exit:  
 if (filler.length() == 1) {  
 for (int i = 0; i < arrsize; i++) {  
 for(int k =0; k<i;k++) {  
 System.*out*.print(" ");  
 myWriter.write( " ");  
 }  
 for (int j = i; j < arrsize - i; j++) {  
  
 arr[i][j] = filler;  
 System.*out*.print(arr[i][j] + " ");  
 myWriter.write(arr[i][j] + " ");  
  
  
 }  
  
  
 System.*out*.println();  
 myWriter.write("\n");  
 }  
 System.*out*.print("\n");  
 myWriter.write("\n");  
 }  
  
 myWriter.flush();  
 myWriter.close();  
 }  
}

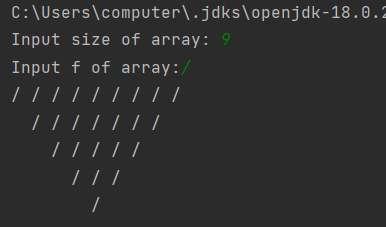
**Результат виконання програми:**



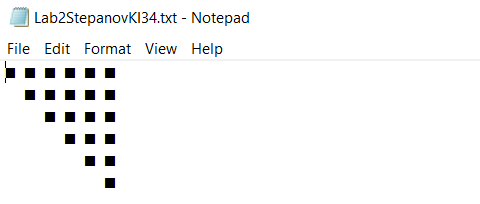
*Переривання програми при введенні кількох символ-заповнювачів*



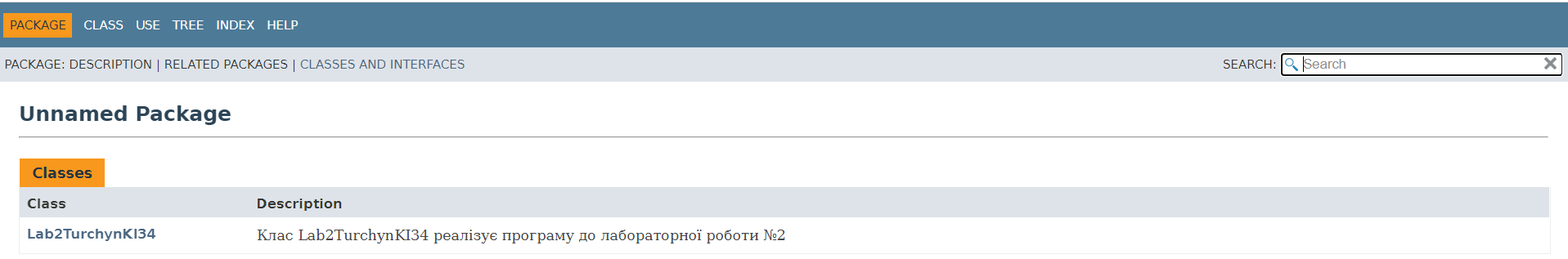
*Переривання програми при не введенні символ-заповнювача*

**

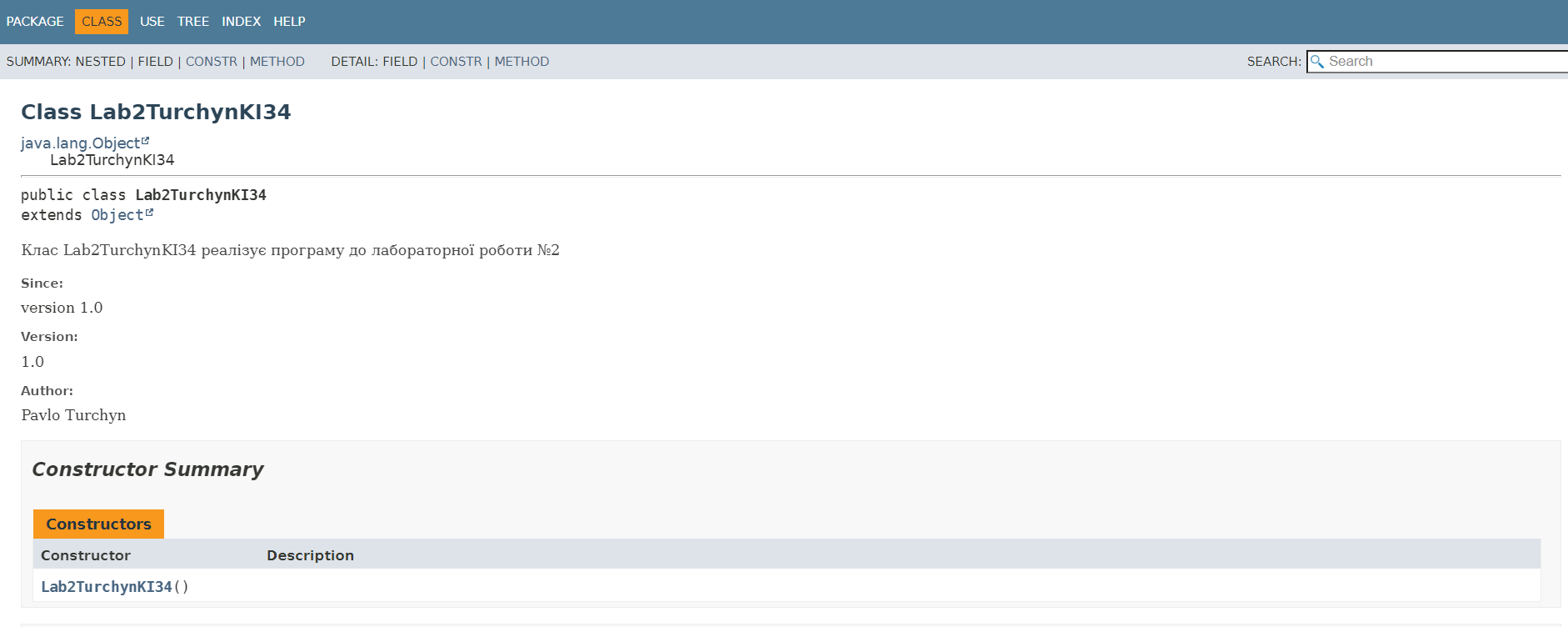
*Успішне виконання програми*



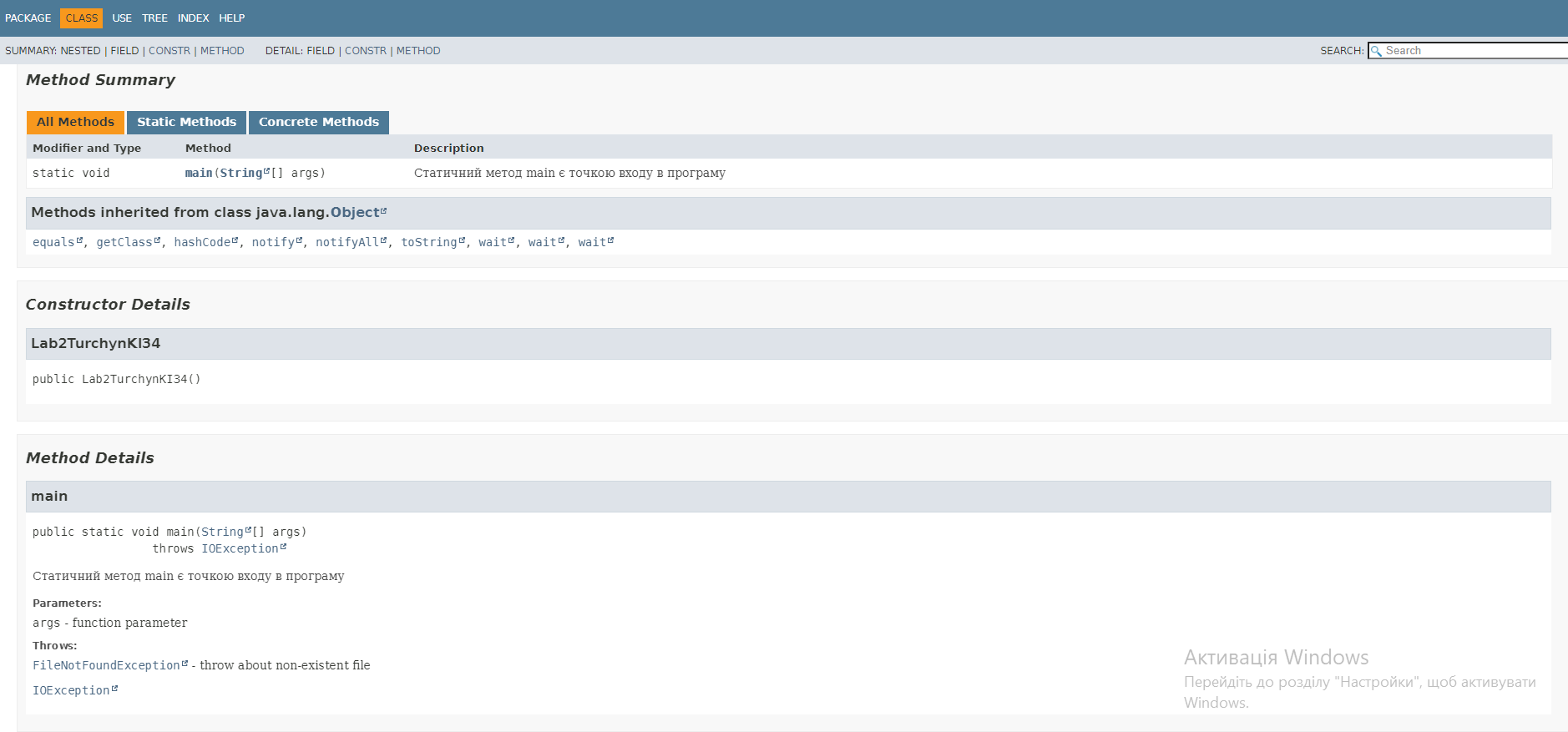
*Успішне виведення масиву у файл*

**

*Згенерована документація*

**

*Інформація про клас Lab2TurchynKI34*

**

*Інформація про клас Lab2TurchynKI34*

**Відповіді на контрольні запитання:**

1. **Які дескриптори використовуються при коментуванні класів?**

При коментуванні класів використовуються такі дескриптори:

* @author
* @deprecated
* @param
* @see
* @version та інші.

1. **Які дескриптори використовуються при коментуванні методів?**

Крім дескрипторів загального призначення для коментування методів використовуються дескриптори:

* @param
* @return
* @throw

1. **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.

1. **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.

Перелік: boolean, char, byte, short, int, long, float, double.

1. **Як оголосити змінну-масив?**

Синтаксиси оголошення та ініціалізації одновимірного масиву:

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

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

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

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

Синтаксиси оголошення та ініціалізації двовимірного масиву:

*тип[][] змінна = new тип[розмір\_виміру\_1][розмір\_виміру\_2];*

*тип[][] змінна = {{значення11, значення12,..., значення1N},*

*{{значення21, значення22,..., значення2N}*

*...*

*{значенняM1, значенняM2,..., значенняMN}};*

*тип змінна[][] = new тип[розмір\_виміру\_1][розмір\_виміру\_2];*

*тип змінна[][] = {{значення11, значення12,..., значення1N},*

*{{значення21, значення22,..., значення2N}*

*...*

*{значенняM1, значенняM2,..., значенняMN}};*

1. **Які керуючі конструкції підтримує Java?**

Основні конструкції мови Java багато в чому співпадають з аналогічними конструкціями мов С/С++. Такі оператори як switch, if-else, while, do-while –ідентичні аналогічним конструкціям у мовах С/С++. Оператор циклу for має деякі особливості. У Java цей оператор має 2 різновиди. До операторів переривання потоку виконання відносяться оператори break і continue.

1. **В чому різниця між різними варіантами оператора for?**

Робота оператора циклу for в стилі С/С++ 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.

1. **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. Зробивши це ми отримаємо доступ до методів класу Scanner, які призначені для введення даних простих типів і рядків.

1. **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. Пошук файлу відбувається у директорії з якої була запущена на виконання програма. Після відкривання файлу інформацію з нього можна читати використовуючи методи класу Scanner.

1. **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. Зробивши це ми отримаємо доступ до методів класу PrintWriter, які призначені для виведення даних простих типів і рядків.

**Висновок:**

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