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 - 7

Виконала:

ст. гр. КІ-305

Дзера А. Р.

Прийняв:

доц. каф. ЕОМ

Іванов Ю. С.

Львів – 2023

**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 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.

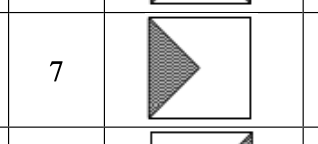
2. Автоматично згенерувати документацію до розробленої програми.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її

виконання та фрагменту згенерованої документації.

4. Дати відповідь на контрольні запитання.

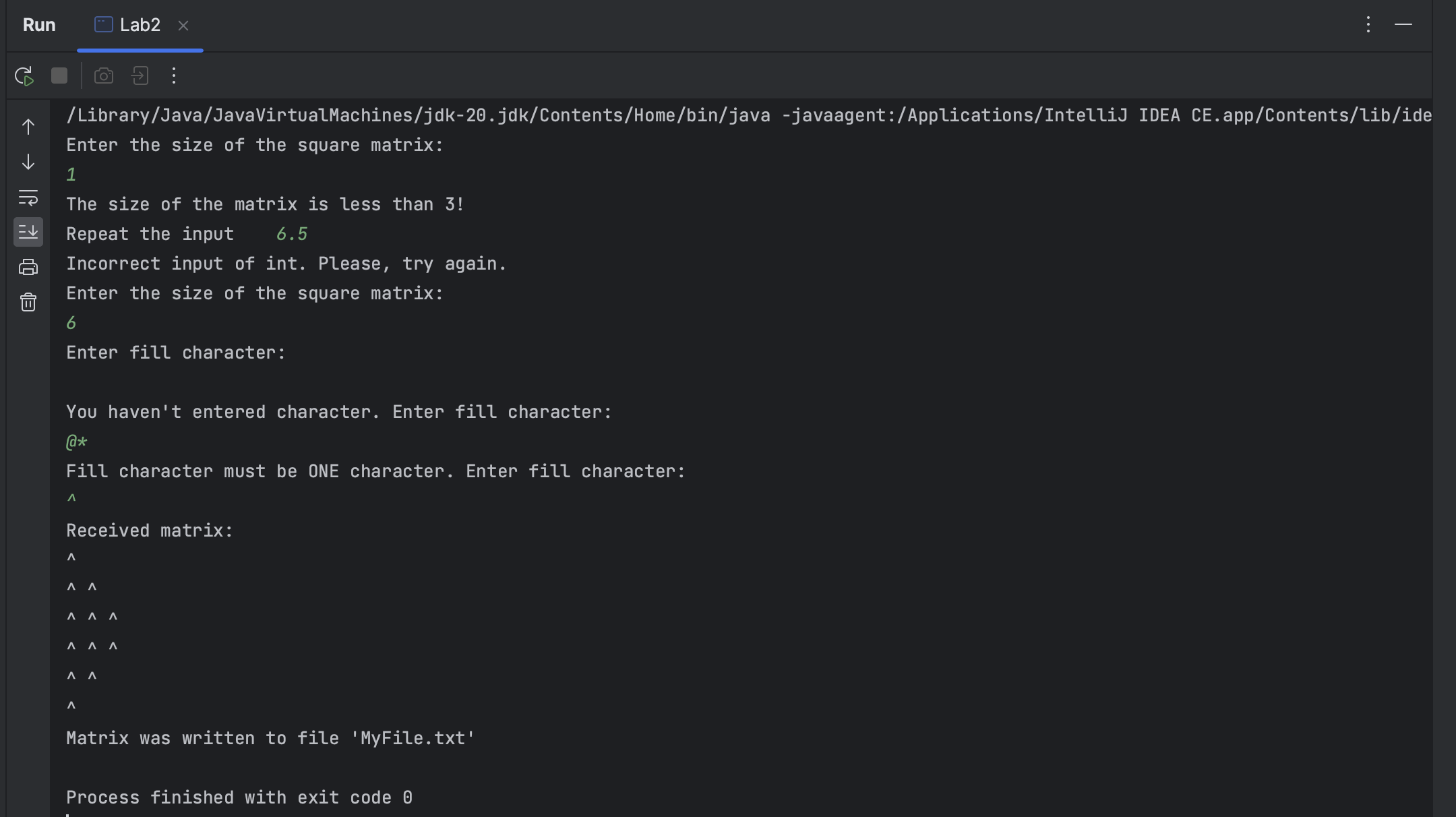
Варіант - 7



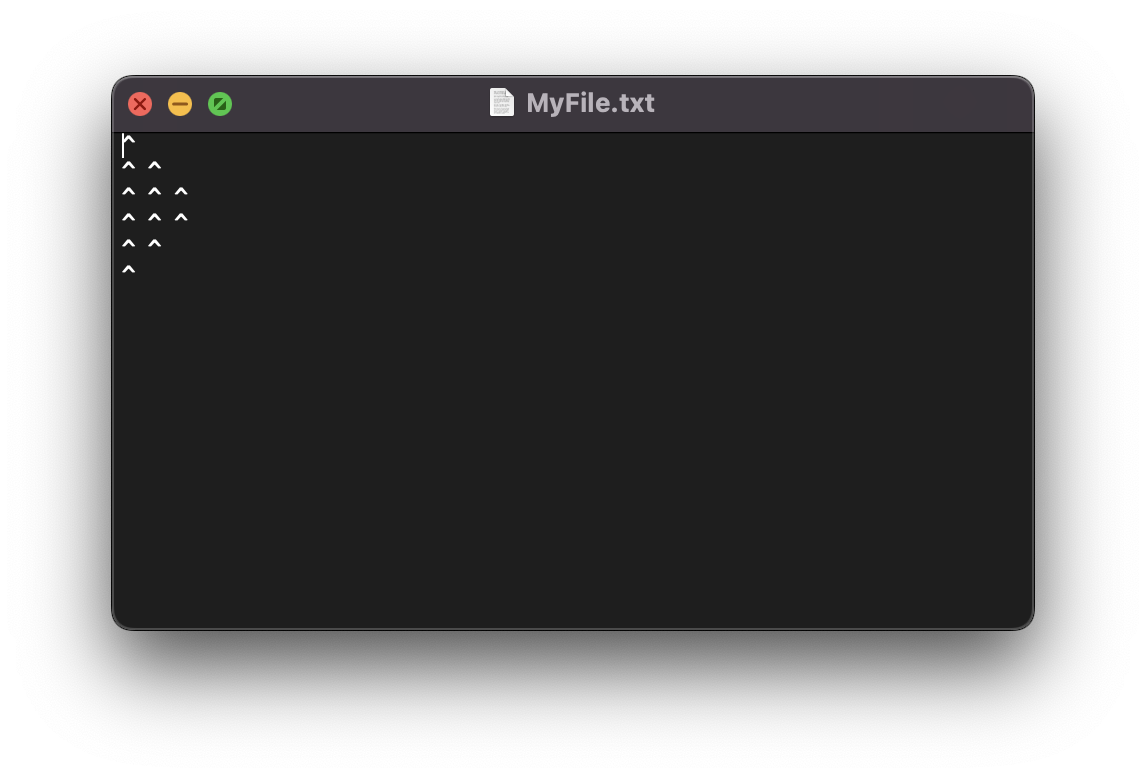
**Код програми**

*/\*\*  
 \* Пакет, що містить класи для лабораторних робіт.  
 \*/*package com.lab;  
  
import java.io.\*;  
import java.util.\*;  
import java.io.FileWriter;  
  
*/\*\*  
 \* Цей клас представляє приклад програми для лабораторної роботи №2.  
 \* ...  
 \** ***@author*** *Amina Dzera  
 \** ***@version*** *1.1  
 \** ***@since*** *1.0  
 \*/*public class Lab2 {  
  
 */\*\*  
 \* Статичний метод main є точкою входу в програму  
 \*  
 \** ***@param*** *args аргументи командного рядка (не використовуються)  
 \** ***@throws*** *IOException виникає, якщо виникають помилки вводу/виводу при роботі з файлом  
 \*/* public static void main(String[] args) throws IOException {  
 // код методу main  
 System.*out*.println("Enter the size of the square matrix: ");  
 Scanner scanIn = new Scanner(System.*in*);  
 //int size = scanIn.nextInt();  
 int size;  
  
 // введення розміру матриці та перевірка на коректність  
 while (true)  
 {  
 try  
 {  
 size = scanIn.nextInt(); //вводимо розмір матриці з клавіатури  
 if (size < 3) //перевіряємо чи наша матриця є більшої розмірності ніж 3  
 {  
 System.*out*.print("The size of the matrix is less than 3! Repeat the input\n"); //якщо ні то виводимо відповідний текст і просимо повторити ввід  
 size = scanIn.nextInt();  
 scanIn.nextLine();  
 }  
 break;  
 }  
 catch (Exception e)  
 {  
 System.*out*.println("Incorrect input of int. Please, try again. ");  
 System.*out*.println("Enter the size of the square matrix:\t");  
 scanIn.nextLine();  
 }  
 }  
 scanIn.nextLine();  
  
  
 String filler;  
 System.*out*.println("Enter fill character: ");  
 filler = scanIn.nextLine();  
  
 char[][] arr = new char[size][];  
 for (int i = 0; i < size; i++) {  
 arr[i] = new char[i + 1];  
 for (int j = 0; j < i + 1; j++) {  
 if (filler.length() == 1) {  
 arr[i][j] = filler.charAt(0);  
 } else if (filler.isEmpty()) {  
 System.*out*.println("You haven't entered character. Enter fill character: ");  
 filler = scanIn.nextLine();  
 break;  
 } else {  
 System.*out*.println("Fill character must be ONE character. Enter fill character: ");  
 filler = scanIn.nextLine();  
 break;  
 }  
 }  
  
 }  
  
 //створюємо масив з вказаним користувачем розміром і заповнюємо згідно варіанту  
 char[][] matrix = new char[size][];  
 for (int i = 0; i < size; i++) {  
 matrix[i] = new char[size]; // кожен рядок має довжину size  
 for (int j = 0; j < size; j++) {  
 if (i >= j && i + j <= size - 1) {  
 matrix[i][j] = filler.charAt(0);  
 } else {  
 matrix[i][j] = ' ';  
 }  
 }  
 }  
  
  
 // виведення отриманої матриці(масиву) на екран  
 for (int i = 0; i < matrix.length; i++) {  
 for (int j = 0; j < matrix[i].length; j++) {  
 System.*out*.print(matrix[i][j] + " ");  
 }  
 System.*out*.println(); // Перехід на новий рядок після кожного рядка матриці  
 }  
  
  
  
  
 // запис результату в файл 'MyFile.txt'  
 FileWriter fileWriter = new FileWriter("MyFile.txt");  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 fileWriter.write(matrix[i][j] + " ");  
 }  
 fileWriter.write("\n");  
 }  
  
 // закриття файлу та завершення роботи програми  
 scanIn.close();  
 fileWriter.flush();  
 fileWriter.close();  
 System.*out*.println("Matrix was written to file 'MyFile.txt'");  
  
 }  
}

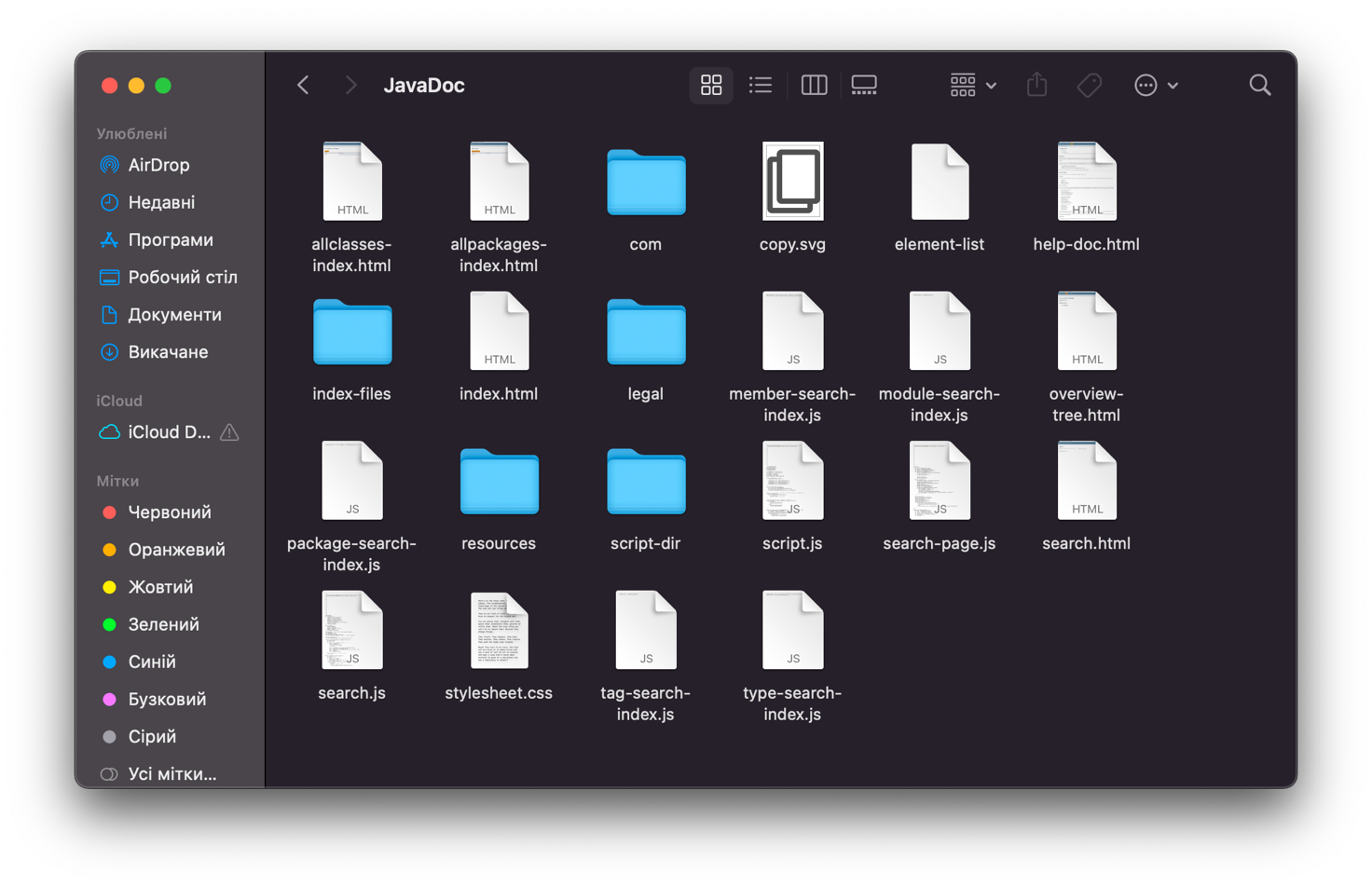
**Результати роботи програми**

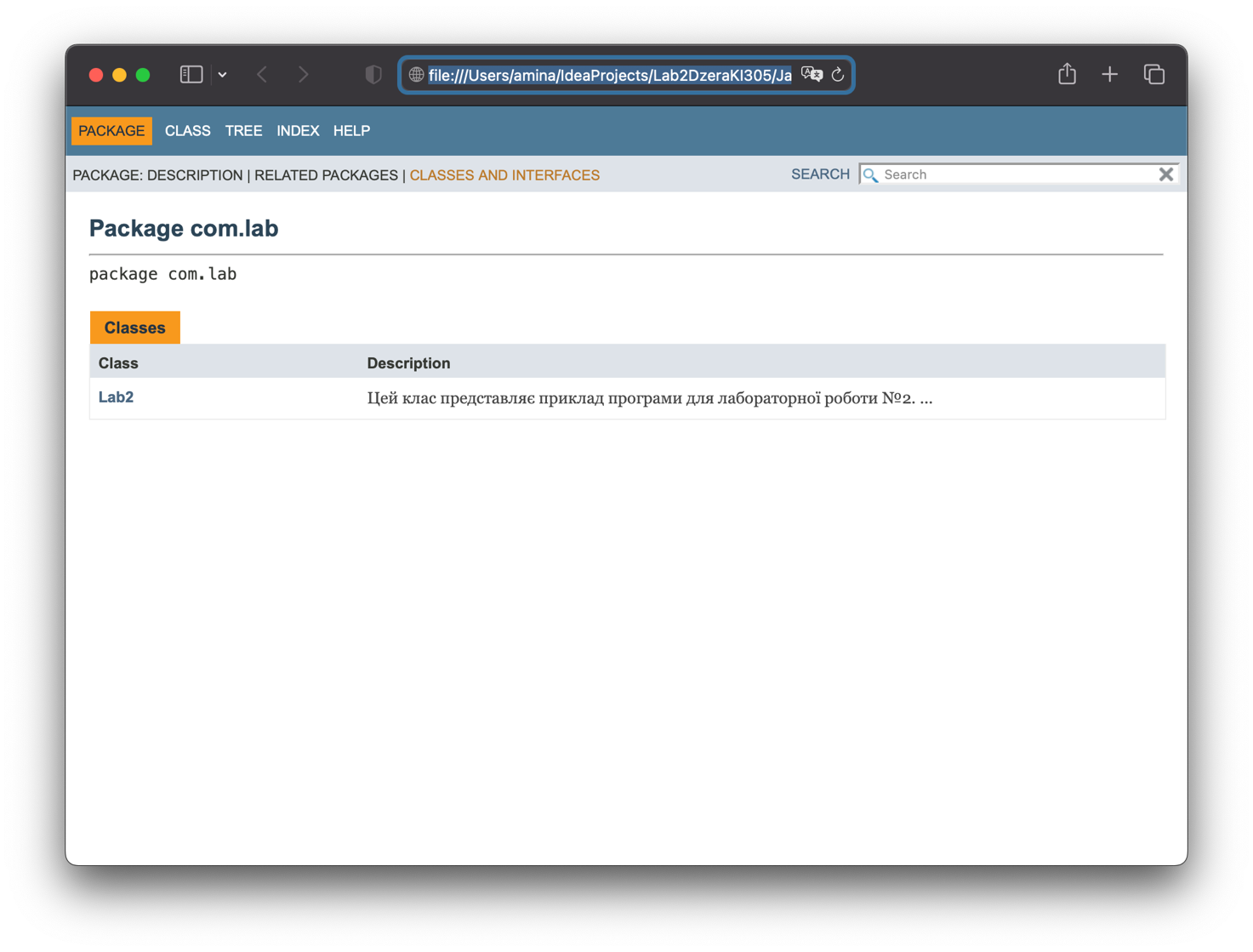
****

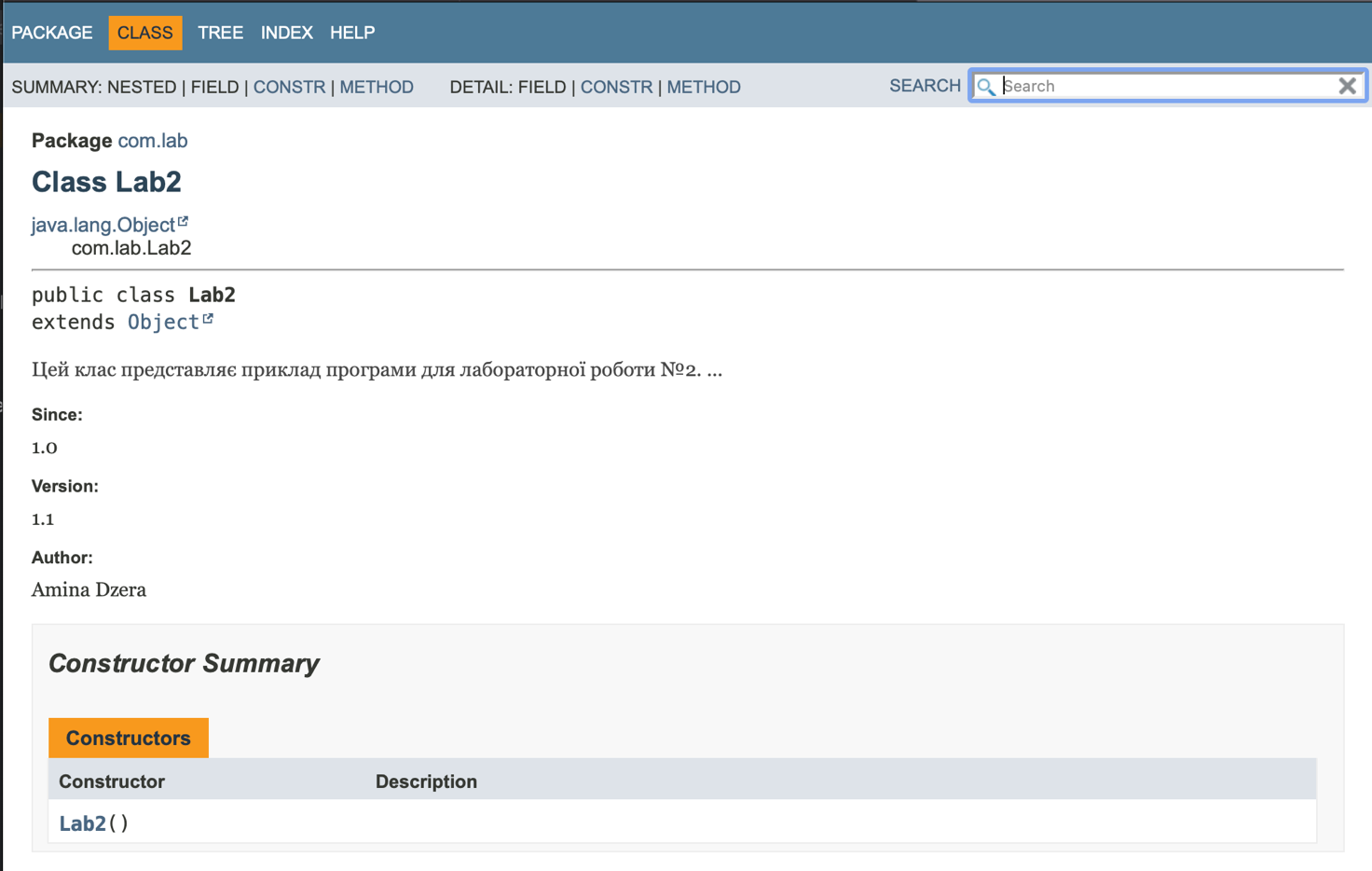
**Вміст файлy MyFile.txt**

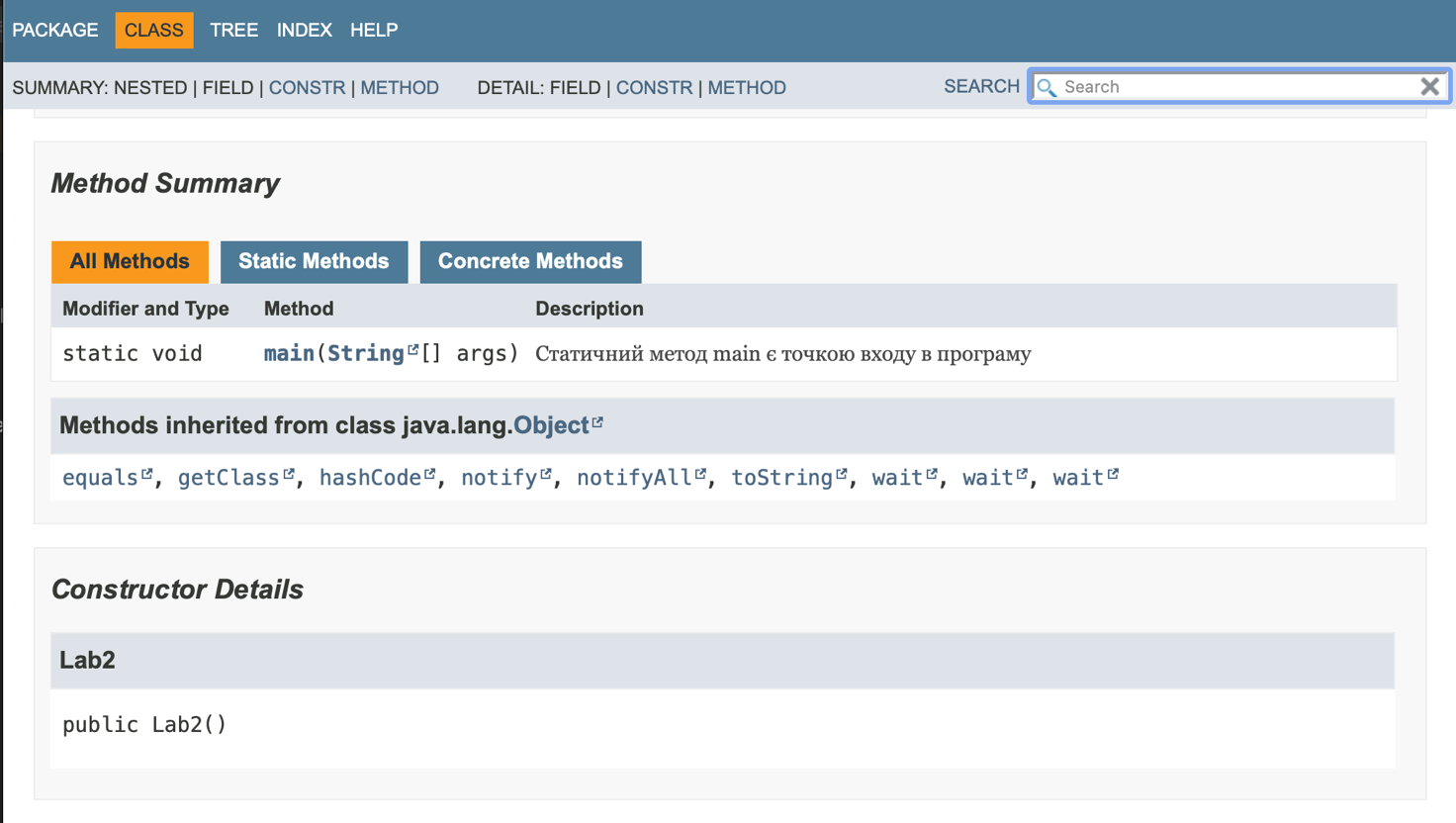
****

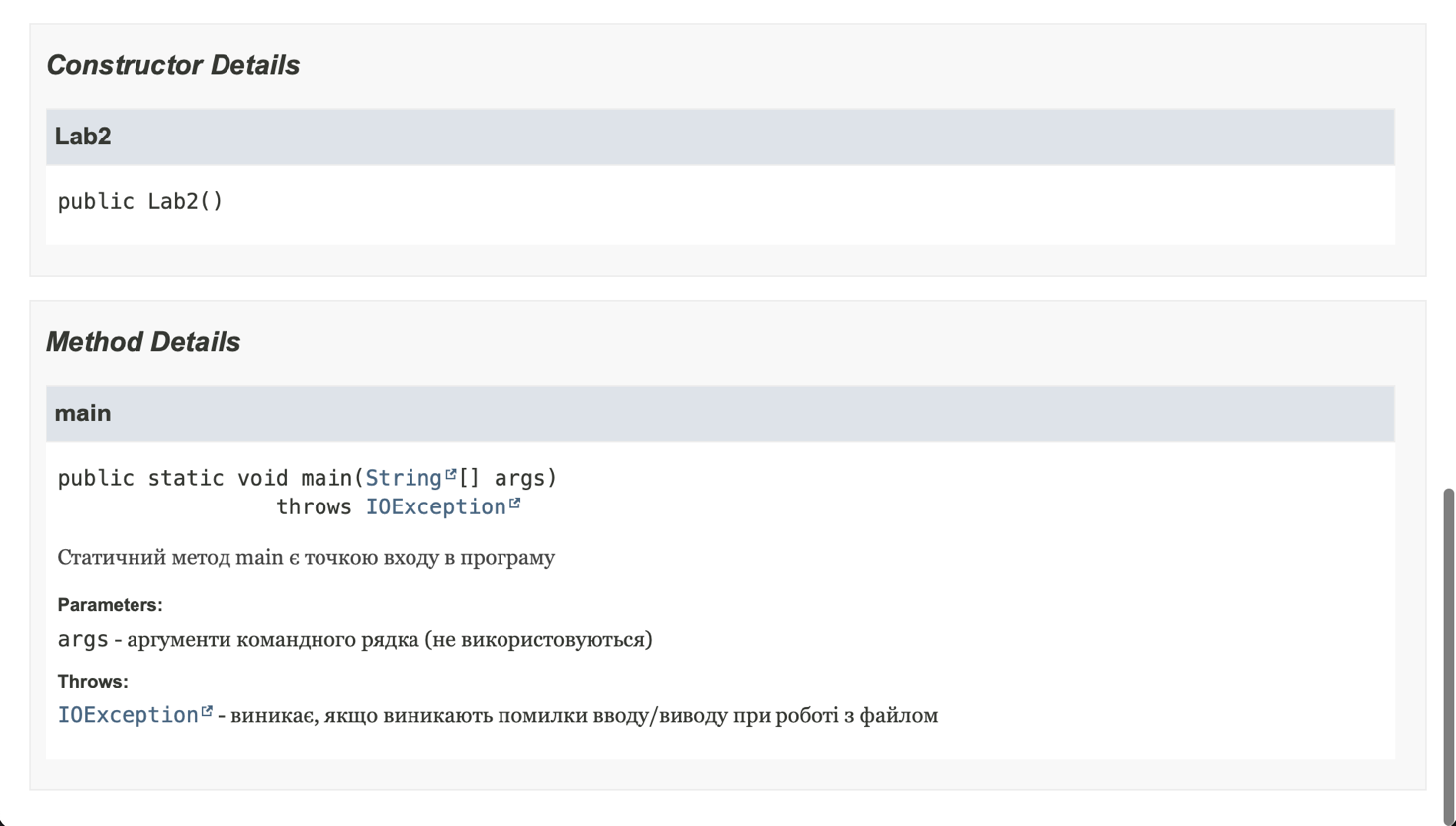
**Фрагмент згенерованої документації**

****

****

****

****

****

**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**

Під час виконання даної лабораторної роботи я ознайомилася з базовими конструкціями мови Java та оволоділа навиками написання й автоматичного документування простих консольних програм мовою Java.