Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**Звіт**

до лабораторної роботи №1

З дисципліни: «Кросплатформенні засоби програмування»

На тему: «Дослідження базових конструкцій мови Java»

**Варіант 8**

Виконав:

Киянець А.М.

Ст. групи КІ- 306

Прийняв:

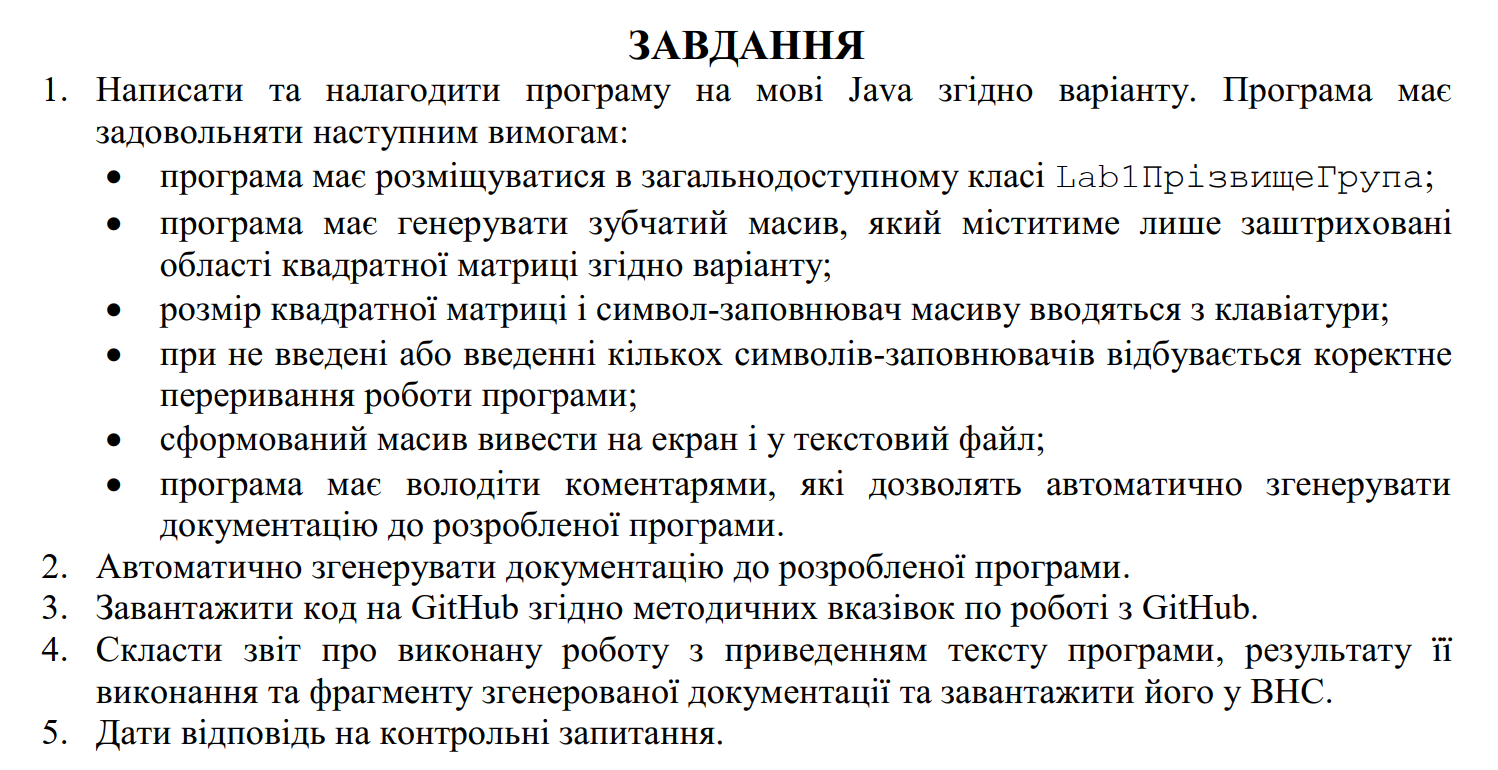
Іванов Ю.С.

Львів 2023

Мета

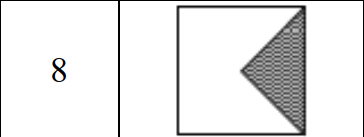
Ознайомитися з базовими конструкціями мови Java та оволодіти навиками написання й автоматичного документування простих консольних програм мовою Java.

Виконання роботи



Завдання

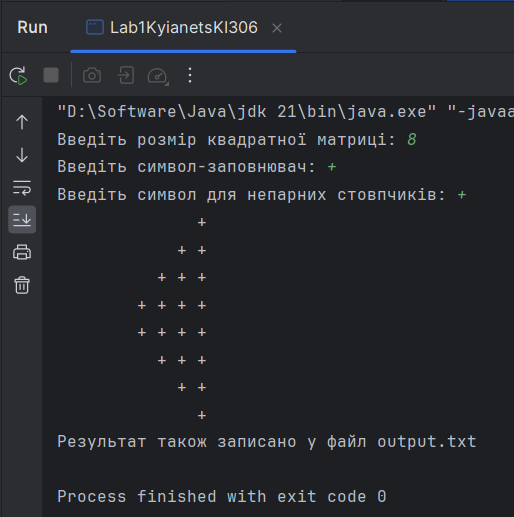
Вивести зубачатий масив, у якому заповнені лише заштриховані області відповідно до варіанту



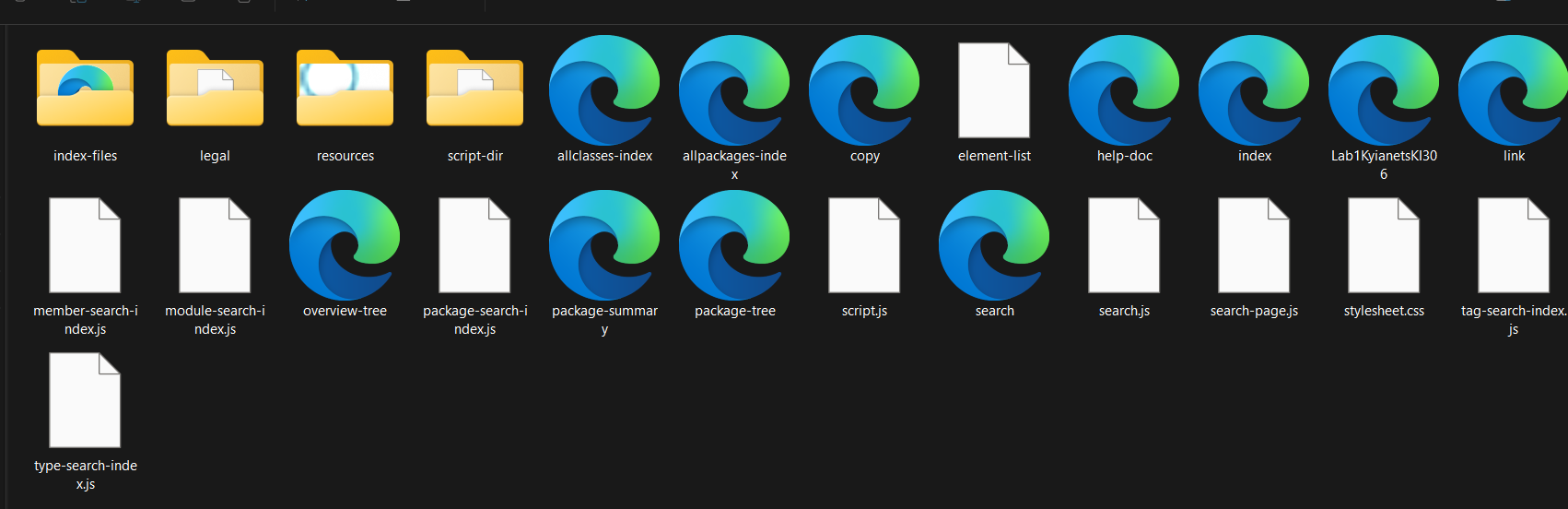
Код програми

import java.io.FileWriter;  
import java.io.PrintWriter;  
import java.util.Scanner;  
  
public class Lab1KyianetsKI306 {  
 public static void main(String[] args) {  
 try {  
 // Введення розміру матриці та символу-заповнювача  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введіть розмір квадратної матриці: ");  
 int size = scanner.nextInt();  
 scanner.nextLine(); // Зчитати залишок рядка після введення числа  
  
 System.*out*.print("Введіть символ-заповнювач: ");  
 String fillChar = scanner.nextLine();  
  
 System.*out*.print("Введіть символ для непарних стовпчиків: ");  
 char oddColumnChar = scanner.nextLine().charAt(0);  
  
 // Генеруємо зубчатий масив  
 char[][] toothedArray = *generateToothedArray*(size, fillChar, oddColumnChar);  
  
 // Повертаємо результат на 90 градусів проти годинникової стрілки  
 toothedArray = *rotateArray90*(toothedArray);  
 toothedArray = *flipArrayHorizontal*(toothedArray);  
  
 // Виводимо зубчатий масив на екран  
 *printArray*(toothedArray);  
  
 // Вивід у файл  
 *writeArrayToFile*(toothedArray, "output.txt");  
  
 } catch (Exception e) {  
 System.*out*.println("Помилка: " + e.getMessage());  
 }  
 }  
  
 // Метод для генерації зубчатого масиву з різними символами для непарних стовпчиків  
 private static char[][] generateToothedArray(int size, String fillChar, char oddColumnChar) {  
 char[][] toothedArray = new char[size][];  
  
 // Генеруємо зубчатий масив  
 for (int i = 0; i < size; i++) {  
 toothedArray[i] = new char[size];  
 for (int j = 0; j < size; j++) {  
 // Визначаємо символ в залежності від умови  
 if (j >= i && j < size - i) {  
 toothedArray[i][j] = (j % 2 == 0) ? fillChar.charAt(0) : oddColumnChar;  
 } else {  
 toothedArray[i][j] = ' ';  
 }  
 }  
 }  
 return toothedArray;  
 }  
  
  
  
  
 // Метод для повороту масиву на 90 градусів  
 private static char[][] rotateArray90(char[][] array) {  
 char[][] rotatedArray = new char[array.length][array[0].length];  
  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array[i].length; j++) {  
 // Поворот елементів масиву на 90 градусів проти годинникової стрілки  
 rotatedArray[i][j] = array[j][array.length - 1 - i];  
 }  
 }  
 return rotatedArray;  
 }  
  
 // Метод для перевертання масиву по горизонталі  
 private static char[][] flipArrayHorizontal(char[][] array) {  
 char[][] flippedArray = new char[array.length][array[0].length];  
  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array[i].length; j++) {  
 // Перевертання елементів масиву по горизонталі  
 flippedArray[i][j] = array[i][array[i].length - 1 - j];  
 }  
 }  
 return flippedArray;  
 }  
  
 // Метод для виведення масиву на екран  
 private static void printArray(char[][] array) {  
 for (char[] row : array) {  
 for (char element : row) {  
 System.*out*.print(element + " ");  
 }  
 System.*out*.println();  
 }  
 }  
  
 // Метод для запису масиву у файл  
 private static void writeArrayToFile(char[][] array, String fileName) {  
 try (PrintWriter writer = new PrintWriter(new FileWriter(fileName))) {  
 for (char[] row : array) {  
 for (char element : row) {  
 writer.print(element + " ");  
 }  
 writer.println();  
 }  
 System.*out*.println("Результат також записано у файл " + fileName);  
 } catch (Exception e) {  
 System.*out*.println("Помилка при записі у файл: " + e.getMessage());  
 }  
 }  
}

Робота модифікованої програми після захисту

****

Сформована Java документація

****

Відповіді на контрольні запитання

1. /\*\* text @ … \*/

2. До методів застосовуються коментарі такі ж, як і до класів(зап. 1)

3. 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. Double, int, char, float, byte, short, Boolean, long.

5. int array[] = new int[size]; double array[][] = new double[size][size];

6. цикл for та for в стилі foreach, перевірка умови оператор if else if, оператори переривання break та continue.

7. for (ініціалізація лічильника; логічна умова; модифікація лічильника) оператори)

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. Оператор циклу for з синтаксисом foreach має наступний вигляд:

for (змінна : набір даних)

оператори

При опрацюванні циклу змінній послідовно присвоюється кожен елемент набору даних (наприклад, елемент масиву) після чого виконується оператор.

8. Для введення інформації з консолі необхідно створити об’єкт класу Scanner і зв’язати його з стандартним потоком вводу System.in.

9. Популярним механізмом виводу на консоль є використання методу print об’єкту out з пакету System, який виводить переданий через параметр текстовий рядок на екран.

10. 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"));

Висновок

Написано програму виведення зубчатого масиву, освоєно створення документації у тому числі атоматичної.