Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 1

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ JAVA»

**Виконав:**

студент групи КІ-301

*Солтисяк О.Р.*

**Прийняв:**

доцент кафедри ЕОМ

*Іванов Ю. С.*

Львів – 2024

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

**Завдання (варіант № 6)**

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.

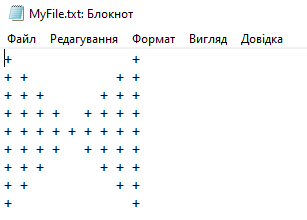
**Вихідний код програми:**

package org.example;  
  
import java.util.Scanner;  
import java.io.FileNotFoundException;  
import java.io.PrintWriter;  
  
public class Lab1SoltysyakKI301 {  
  
 private static final String *FILE\_NAME* = "MyFile.txt";  
  
 public static void main(String[] args) {  
 int size = *inputSize*();  
 char filler = *inputFiller*();  
  
 char[][] matrix = *generateMatrix*(size, filler);  
  
 *showMatrix*(matrix, size);  
 *saveMatrixToFile*(matrix, size);  
 }  
  
 public static int inputSize() {  
 Scanner scanner = new Scanner(System.*in*);  
 int size;  
  
 while (true) {  
 try {  
 System.*out*.print("Enter size of square matrix: ");  
 size = scanner.nextInt();  
 if (size > 1) break;  
 else System.*out*.println("Size must be greater than 1.");  
 } catch (Exception e) {  
 System.*out*.println(e.getMessage() + ". Please enter a number.");  
 scanner.next();  
 }  
 }  
 return size;  
 }  
  
 public static char inputFiller() {  
 Scanner scanner = new Scanner(System.*in*);  
 char filler;  
  
 while (true) {  
 System.*out*.print("Enter character to fill matrix: ");  
 String str = scanner.nextLine().trim();  
  
 if (!str.isEmpty()) {  
 filler = str.charAt(0);  
 break;  
 } else {  
 System.*out*.println("Please enter a non-empty character.");  
 }  
 }  
  
 return filler;  
 }  
  
 public static char[][] generateMatrix(int size, char filler) {  
 char[][] matrix = new char[size][];  
  
 for (int i = 0; i < size; i++) {  
 // Визначаємо кількість заповнених символів у рядку  
 int filledCount = 0;  
 for (int j = 0; j < size; j++) {  
 if ((i >= j && i + j < size) || (i <= j && i + j >= size - 1)) {  
 filledCount++;  
 }  
 }  
  
 // Ініціалізуємо рядок зубчастого масиву з потрібною довжиною  
 matrix[i] = new char[filledCount];  
  
 int index = 0;  
 for (int j = 0; j < size; j++) {  
 if ((i >= j && i + j < size) || (i <= j && i + j >= size - 1)) {  
 matrix[i][index++] = filler;  
 }  
 }  
 }  
 return matrix;  
 }  
  
 public static void showMatrix(char[][] matrix, int size) {  
 for (int i = 0; i < size; i++) {  
 int index = 0;  
 for (int j = 0; j < size; j++) {  
 if ((i >= j && i + j < size) || (i <= j && i + j >= size - 1)) {  
 System.*out*.print(matrix[i][index++] + " ");  
 } else {  
 System.*out*.print(" ");  
 }  
 }  
 System.*out*.println();  
 }  
 }  
  
 public static void saveMatrixToFile(char[][] matrix, int size) {  
 try (PrintWriter writer = new PrintWriter(*FILE\_NAME*)) {  
 for (int i = 0; i < size; i++) {  
 int index = 0;  
 for (int j = 0; j < size; j++) {  
 if ((i >= j && i + j < size) || (i <= j && i + j >= size - 1)) {  
 writer.print(matrix[i][index++] + " ");  
 } else {  
 writer.print(" ");  
 }  
 }  
 writer.println();  
 }  
 } catch (FileNotFoundException e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

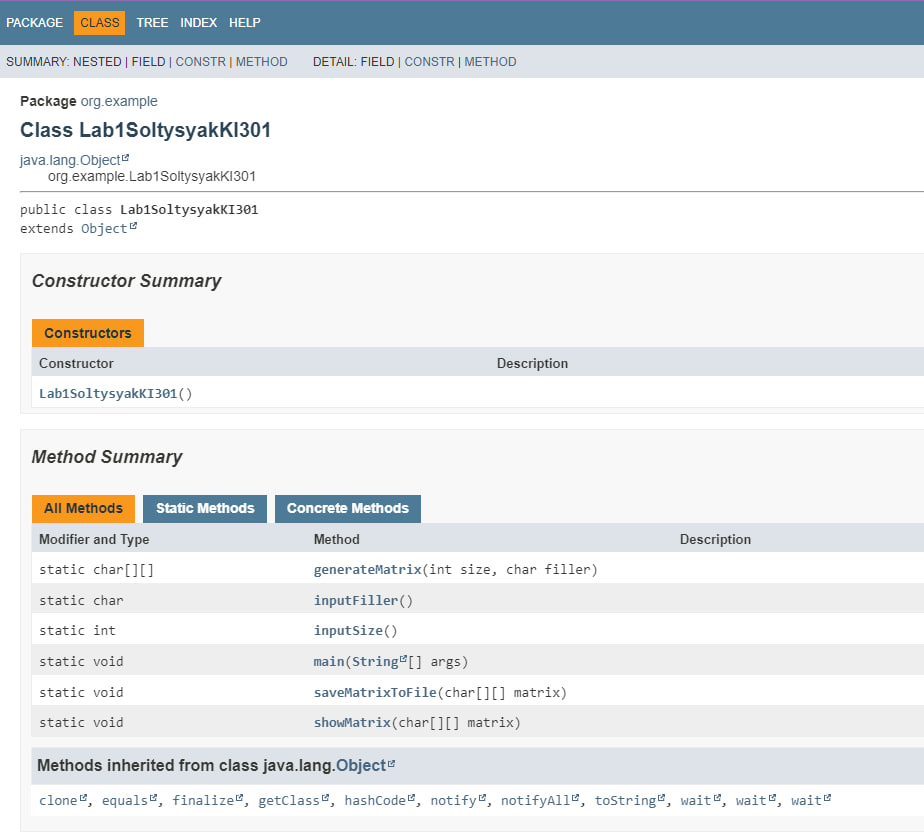
**Результат виконання програми:**

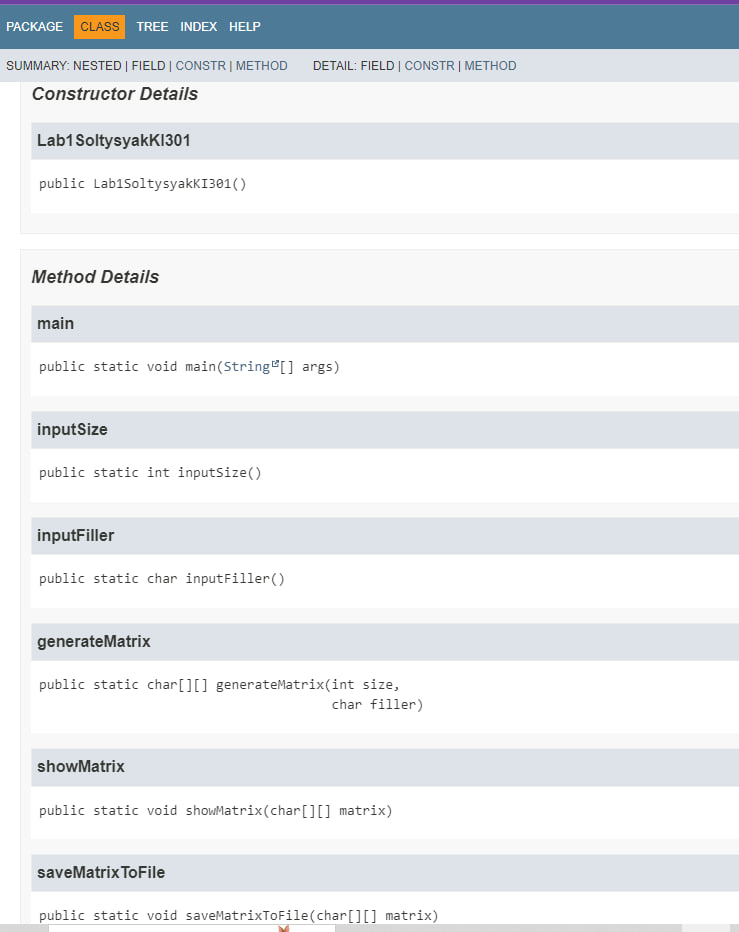


**Вміст файлу MyFile.txt:**



**JavaDoc:**



  
  
  
**Відповіді на контрольні запитання:**

1. Які дескриптори використовуються при коментуванні класів?

/\*\* ... \*/: JavaDoc-коментарі, що дозволяють створювати докладну документацію для класу.

/\* ... \*/: Звичайні багаторядкові коментарі для додавання загальних коментарів до класу.

// ...: Однорядкові коментарі для коротких пояснень чи коментарів до окремих рядків коду.

1. Які дескриптори використовуються при коментуванні методів?

*• @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.

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 документацію до пакету.

1. Які прості типи даних підтримує Java?

byte: 8-бітне ціле число.

short: 16-бітне ціле число.

int: 32-бітне ціле число.

long: 64-бітне ціле число.

float: 32-бітне число з рухомою комою (число з плаваючою точкою).

double: 64-бітне число з рухомою комою (число з плаваючою точкою).

char: 16-бітний символ Unicode.

boolean: Логічний тип, може бути true або false.

1. Як оголосити змінну-масив?

Приклади оголошення неініціалізованого одновимірного масиву типу int:

*int[] arr;*

*int arr[];*

1. Які керуючі конструкції підтримує Java?

if-else: Умовний оператор для виконання коду на основі умови.

switch: Мультиплікаційний оператор для вибору одного з багатьох можливих шляхів виконання коду на основі значення виразу.

for: Цикл для повторення коду певну кількість разів або на основі ітерації.

while: Цикл, який виконується, поки умова істинна.

do-while: Цикл, який виконується принаймні один раз, після чого перевіряється умова.

break: Використовується для виходу із циклу або вибору виразу.

continue: Використовується для переходу до наступної ітерації циклу.

return: Використовується для повернення значення з методу.

throw: Використовується для викидання виняткової ситуації.

try-catch-finally: Використовується для обробки виняткових ситуацій.

1. В чому різниця між різними варіантами оператора for?

У Java цей оператор має 2 різновиди:

• конструкція в стилі С/С++ з полем ініціалізації, логічною умовою та кроком;

• конструкція з *синтаксисом foreach* (foreach дозволяє послідовно перебирати всі елементи набору даних без застосування лічильника.).

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*, наприклад:

*Scanner in = new Scanner(System.in);*

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*:

*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:**

На даній лабораторній роботі я ознайомився з інтегрованим середовищем розробки, також ознайомився з базовими конструкціями мови Java та оволодів навиками написання й автоматичного документування простих консольних програм мовою Java.