Міністерство освіти і науки України

Національний університет „Львівська політехніка”



**Звіт**

з лабораторної роботи №1

з дисципліни: “Кросплатформні засоби

програмування”

на тему: “ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ JAVA”

Варіант 14

Виконав: ст. гр. КІ-301

Матвіїшин М.З

Прийняв:

Майдан М.В

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.

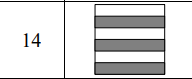
2. Автоматично згенерувати документацію до розробленої програми.

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 14**



**Хід роботи**

Лістинг програми:

**package** matviishyn\_LB;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.PrintWriter;

**import** java.util.Scanner;

/\*\*

\* Клас matviishyn\_LB - створення та запис матриці з двома різними символами у файл.

\*/

**public** **class** matviishyn\_LB {

**public** **static** **void** main(String[] args) **throws** FileNotFoundException {

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

**int** nRows = *getMatrixSize*(in);

**char**[][] arr = **new** **char**[nRows][nRows];

String filler1 = *getFillerSymbol*("перший", in);

String filler2 = *getFillerSymbol*("другий", in);

*writeMatrixToFile*(nRows, arr, filler1, filler2);

*printMatrixToConsole*(nRows, arr);

}

/\*\*

\* Метод для запису матриці у файл.

\*

\* **@param** nRows Розмір матриці.

\* **@param** arr Двовимірний масив матриці.

\* **@param** filler1 Перший символ-заповнювач.

\* **@param** filler2 Другий символ-заповнювач.

\* **@throws** FileNotFoundException Виняток, який виникає, якщо файл "MyFile.txt" не знайдено або не створено.

\*/

**private** **static** **void** writeMatrixToFile(**int** nRows, **char**[][] arr, String filler1, String filler2) **throws** FileNotFoundException {

File dataFile = **new** File("MyFile.txt");

PrintWriter fout = **new** PrintWriter(dataFile);

**for** (**int** i = 0; i < nRows; i++) {

**for** (**int** j = 0; j < nRows; j++) {

**char** filler = (i % 2 == 0) ? filler1.charAt(0) : filler2.charAt(0);

arr[i][j] = filler;

fout.print(arr[i][j] + " ");

}

fout.println();

}

fout.close();

}

/\*\*

\* Метод для виведення матриці в консоль.

\*

\* **@param** nRows Розмір матриці.

\* **@param** arr Двовимірний масив матриці.

\*/

**private** **static** **void** printMatrixToConsole(**int** nRows, **char**[][] arr) {

**for** (**int** i = 0; i < nRows; i++) {

**for** (**int** j = 0; j < nRows; j++) {

System.***out***.print(arr[i][j] + " ");

}

System.***out***.println(); // Перехід на новий рядок у консоль

}

}

/\*\*

\* Метод для отримання розміру квадратної матриці.

\*

\* **@return** Розмір квадратної матриці.

\*/

**private** **static** **int** getMatrixSize(Scanner in) {

**int** nRows;

**do** {

nRows = *readIntegerInput*("Введіть розмір квадратної матриці: ", in);

in.nextLine();

} **while** (nRows <= 0);

**return** nRows;

}

/\*\*

\* Метод для отримання символу-заповнювача.

\*

\* **@param** order Номер символу-заповнювача (перший або другий).

\* **@return** Символ-заповнювач.

\*/

**private** **static** String getFillerSymbol(String order, Scanner in) {

String filler;

**do** {

filler = *readStringInput*("Введіть " + order + " символ-заповнювач (лише один символ): ", in);

} **while** (filler.length() != 1);

**return** filler;

}

/\*\*

\* Метод для зчитування цілого числа з клавіатури.

\*

\* **@param** prompt Повідомлення, яке виводиться перед введенням.

\* **@param** in Об'єкт Scanner для зчитування введення.

\* **@return** Введене ціле число.

\*/

**private** **static** **int** readIntegerInput(String prompt, Scanner in) {

System.***out***.print(prompt);

**while** (!in.hasNextInt()) {

System.***out***.print("Будь ласка, введіть коректне ціле число: ");

in.nextLine();

}

**return** in.nextInt();

}

/\*\*

\* Метод для зчитування рядка з клавіатури.

\*

\* **@param** message Повідомлення, яке виводиться перед введенням.

\* **@return** Введений рядок.

\*/

**private** **static** String readStringInput(String message, Scanner in) {

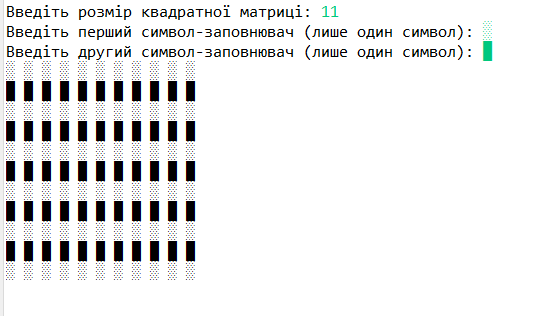
System.***out***.print(message);

**return** in.nextLine();

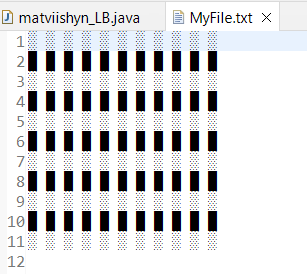
}

}

Екранний вивід в консолі



Запис в текстовому файлі



**КОНТРОЛЬНІ ПИТАННЯ**

1. Які дескриптори використовуються при коментуванні класів?

2. Які дескриптори використовуються при коментуванні методів?

3. Як автоматично згенерувати документацію?

4. Які прості типи даних підтримує Java?

5. Як оголосити змінну-масив?

6. Які керуючі конструкції підтримує Java?

7. В чому різниця між різними варіантами оператора for?

8. Як здійснити ввід з консолі?

9. Як здійснити ввід з текстового файлу?

10. Як здійснити запис у текстовий файл?

**Відповіді до контрольних запитань**

1. Дескриптори, які використовуються при коментуванні класів, включають Javadoc коментарі, які починаються з символу /\*\*
2. Для коментування методів також використовуються Javadoc коментарі, які розміщуються перед оголошенням методу. Вони містять інформацію про параметри методу, тип поверненого значення та опис його функціональності.
3. Документацію можна автоматично згенерувати за допомогою інструментів, таких як JavaDoc. Для цього вам потрібно використовувати команду javadoc у командному рядку, і вона створить HTML-файли документації з вашого коду на основі Javadoc коментарів
4. Прості типи даних, які підтримує Java, включають int, double, float, char, boolean, byte, short і long.
5. Змінну-масив в Java оголошують, вказавши тип даних, за яким слід слідувати квадратні дужки []. Наприклад: int[] numbers;
6. Java підтримує керуючі конструкції, такі як if, else, switch, while, do-while, for, break, і continue.
7. For повторює ітерацію до певного моменту (визначений наперед), а foreach проходить по всім об’єктам певного масиву.
8. Для ввіду з консолі в Java використовується клас Scanner. Вам потрібно створити об'єкт Scanner, а потім викликати його методи для зчитування різних типів даних з консолі.
9. Для ввіду з текстового файлу в Java використовується клас Scanner або BufferedReader. Вам потрібно створити об'єкт Scanner або BufferedReader, який буде зв'язаний з файлом, і використовувати його методи для зчитування даних з файлу.
10. Для запису у текстовий файл в Java використовуються класи FileWriter, BufferedWriter, а також PrintWriter. Ви створюєте об'єкт одного з цих класів, який буде пов'язаний з файлом, і використовуєте їх методи для запису даних у файл.