

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

Лис Б. Л.

Прийняв:

викл. каф. ЕОМ

Іванов Ю. С.

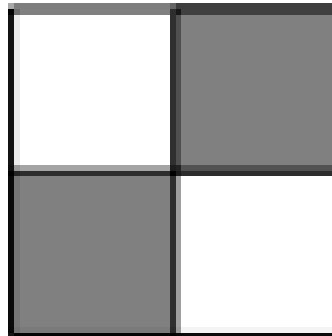
Львів 2022

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;
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: 11.



Лістинг програми:

```
import java.io.*;
import java.util.*;

/**
 * Клас Lab2LysKI34 реалізує програму до лабораторної роботи №2
 *
 * @author Lys Bohdan
 * @version 1.0
 * @since version 1.0
 */

public class Lab2LysKI34 {
    /**
     * 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 FileNotFoundException {
    int arrsize;
    char[][] arr;
    String filler;

    File myFile = new File("Lab2LysKI34.txt");
    Scanner myScanFile = new Scanner(myFile);
    Scanner myScanSys = new Scanner(System.in);
    PrintWriter myWriter = new PrintWriter(myFile);

    System.out.print("Input size of matrix: ");
    arrsize = myScanSys.nextInt();
    myScanSys.nextLine();

    System.out.print("Input filler symbol: ");
    filler = myScanSys.nextLine();

    arr = new char[arrsize][(int) arrsize / 2];
    for (int i = 0; i < arrsize; i++) {
        if (i < (int) arrsize / 2) {
            for (int j = 0; j < (int) arrsize / 2; j++)
                /*System.out.print("\t");*/
            for (int j = 0; j < (int) arrsize / 2; j++) {
                arr[i][j] = ' ';
            }
        }
        if (filler.length() == 1) {
            for (int i = 0; i < arrsize; i++) {
                if (i < (int) arrsize / 2) {
                    for (int j = 0; j < (int) arrsize / 2; j++) {
                        System.out.print(" ");
                        myWriter.print(" ");
                    }
                }
                for (int j = 0; j < (int) arrsize / 2; j++) {
                    arr[i][j] = (char) filler.codePointAt(0);
                    ;
                    System.out.print(arr[i][j] + " ");
                    myWriter.print(arr[i][j] + " ");
                }
                myWriter.println();
                System.out.println();
            }
        }

        if (filler.length() == 0) {
            System.out.print("\nNo filler symbol entered");
            System.exit(0);
        }
        if (filler.length() > 1) {
            System.out.print("\nToo many filler symbol");
            System.exit(0);
        }
        myWriter.flush();
        myWriter.close();
    }
}

```

Результат виконання програми:

```
C:\Users\mrpet\.jdk\openjdk-18.0.2\bin\java.exe "-javaagent:C:\Pro
Input size of matrix: 10
Input filler symbol: xc

Too many filler symbol
Process finished with exit code 0
```

Переривання програми при введенні кількох символ-заповнювачів

```
C:\Users\mrpet\.jdk\openjdk-18.0.2\bin\java.exe "-javaagent:C:
Input size of matrix: 10
Input filler symbol:

No filler symbol entered
Process finished with exit code 0
```

Переривання програми при не введенні символ-заповнювача

```
C:\Users\mrpet\.jdk\openjdk-18.0.2\bin\java.exe "-javaagent:C:\Prog
Input size of matrix: 10
Input filler symbol: b

  b b b b b
  b b b b b
  b b b b b
  b b b b b
  b b b b b
b b b b b
b b b b b
b b b b b
b b b b b
b b b b b

Process finished with exit code 0
```

Успішне виконання програми

Lab2LysKI34.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

```
      b b b b b
      b b b b b
      b b b b b
      b b b b b
      b b b b b
b b b b b
b b b b b
b b b b b
b b b b b
b b b b b
```

Успішне виведення масиву у файл

PACKAGE	CLASS	USE	TREE	INDEX	HELP
PACKAGE: DESCRIPTION RELATED PACKAGES CLASSES AND INTERFACES					
Unnamed Package					
Classes					
Class	Description				
Lab2LysKI34	Клас Lab2LysKI34 реалізує програму до лабораторної роботи №2				

Згенерована документація

PACKAGE
CLASS
USE
TREE
INDEX
HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD

Class Lab2LysKI34

java.lang.Object[Ⓢ]
Lab2LysKI34

```
public class Lab2LysKI34
extends ObjectⓈ
```

Клас Lab2LysKI34 реалізує програму до лабораторної роботи №2

Since:
version 1.0

Version:
1.0

Author:
Lys Bohdan

Constructor Summary

Constructors

Constructor	Description
Lab2LysKI34()	

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method	Description
static void	main(String [Ⓢ] [] args)	Статичний метод main є точкою входу в програму

Methods inherited from class java.lang.Object[Ⓢ]

equals[Ⓢ], getClass[Ⓢ], hashCode[Ⓢ], notify[Ⓢ], notifyAll[Ⓢ], toString[Ⓢ], wait[Ⓢ], wait[Ⓢ], wait[Ⓢ]

Інформація про клас Lab2LysKI34

Відповіді на контрольні запитання:

1. Які дескриптори використовуються при коментуванні класів?

При коментуванні класів використовуються такі дескриптори:

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

2. Які дескриптори використовуються при коментуванні методів?

Крім дескрипторів загального призначення для коментування методів використовуються дескриптори:

- @param
- @return
- @throw

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 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.

5. 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}};
```

6. Які керуючі конструкції підтримує Java?

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

7. В чому різниця між різними варіантами оператора for?

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

8. 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.

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

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. 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.