

Міністерство освіти і науки України
Національний університет „Львівська політехніка”



Звіт

з лабораторної роботи №2

дисципліни: **“Кросплатформні засоби програмування”**

на тему: **“Дослідження базових конструкцій мови Java”**

Виконав:

ст. гр. КІ-34

Скалій Т.В.

Прийняв:

Іванов Ю.С.

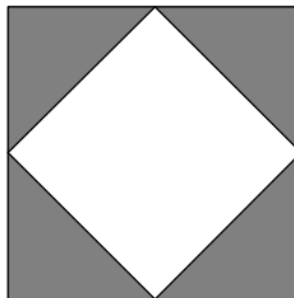
Львів – 2022

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 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 №20



Лістинг програми:

```
import java.io.*;
import java.util.*;
/**
 * Клас Lab2SkaliiKI34 реалізує програму до лабораторної роботи №2
 *
 * @author Tetiana Skalii
 * @version 1.0
 * @since version 1.0
 */

public class Lab2SkaliiKI34 {
    private static Scanner in;

    /**
```

```

* Статичний метод main є точкою входу в програму
*
* @param args
* @throws FileNotFoundException
*
*/
public static void main(String[] args) throws FileNotFoundException {
    int nRows;
    int h;
    int bridge = 0;
    char[][] arr;
    String filler;
    boolean check = true;
    in = new Scanner(System.in);
    File dataFile = new File("Lab2SkaliiKI34.txt");
    PrintWriter fOut = new PrintWriter(dataFile);

    do {
        System.out.print("Input odd number(>=3), which will be responsible
for the size of the matrix: ");
        nRows = in.nextInt();
        in.nextLine();

        if(nRows<3 || nRows%2==0){
            System.out.println("Your number is incorrect, please try
again");
        }
        else{
            check = false;
        }
    }while(check);
    h=nRows-1;
    nRows-=1;
    arr = new char[nRows][];
    for (int i =0;i<nRows;i++){
        arr[i] = new char[h];
        if(i<nRows/2){
            h-=2;
            if(i==(nRows/2)-1){
                h=2;
            }
        }else if(i == nRows/2){
            h=2;
        } else {
            h+=2;
            arr[i] = new char[h];
        }
    }

    do {
        System.out.print("Input filler symbol: ");
        filler = in.nextLine();
        if(filler.length() == 1){
            check = false;
        }else if(filler.length() == 0){
            System.out.println("No symbol found");
            System.exit(0);
        }
    }
}

```

```

    }
    else{
        System.out.println("Too much symbols");
        System.exit(0);
    }
}while(check);

for(int i = 0; i<nRows; i++){
    for(int j =0;j< arr[i].length/2;j++){
        arr[i][j] = (char) filler.codePointAt(0);
        System.out.print(arr[i][j] + " ");
        fOut.print(arr[i][j] + " ");
        bridge = j;
    }
    for(int a = 0;a<(nRows+1)-arr[i].length;a++){
        System.out.print(" ");
        fOut.print(" ");
    }
    for(int j = bridge+1;j<arr[i].length;j++){
        arr[i][j] = (char) filler.codePointAt(0);
        System.out.print(arr[i][j] + " ");
        fOut.print(arr[i][j] + " ");
    }
    if(i==(nRows/2)-1){
        System.out.println();
        fOut.println();
    }
    System.out.println();
    fOut.println();
}
fOut.flush();
fOut.close();
}
}

```

Результат виконання програми:

```

Input odd number(>=3), which will be responsible for the size of the matrix: 9
Input filler symbol: &
& & & &  & & & &
& & &      & & &
& &        & &
&          &
&          &

&          &
& &        & &
& & &      & & &
& & & &  & & & &

```

Рис.1.Результат виконання програми

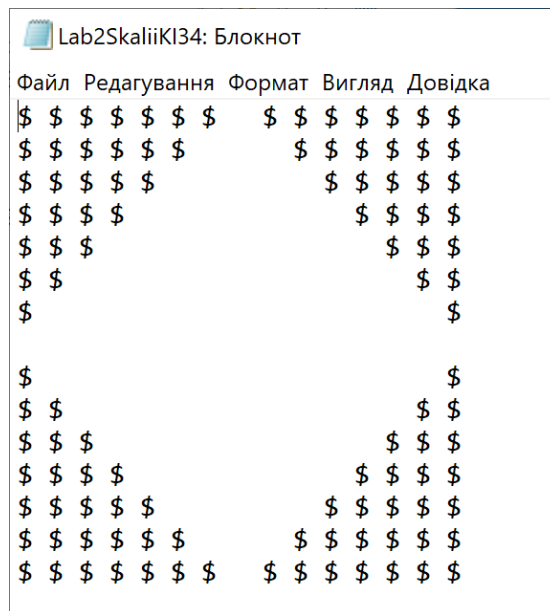


Рис.2.Виведення масиву у текстовий файл

Згенерована документація:

Unnamed Package

Classes

Class	Description
Lab2SkaliiKI34	Клас Lab2SkaliiKI34 реалізує програму до лабораторної роботи №2

Рис.3.Вмістиме вкладки Package

PACKAGE CLASSES USE TREE INDEX HELP

JAVADOC NESTED FIELD CONSTRUCTOR METHOD DETAIL FIELD CONSTRUCTOR METHOD

88/100 Search

Class Lab2SkaliiKI34

Java.lang.Object
Lab2SkaliiKI34

public class Lab2SkaliiKI34
extends Object

Клас Lab2SkaliiKI34 реалізує програму до лабораторної роботи №2

Since:
version 1.0

Version:
1.0

Author:
Tetiana Shali

Constructor Summary

Constructors

Constructor	Description
Lab2SkaliiKI34()	

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method	Description
static void	main(String[] args)	Стартовий метод майже в кожній програмі

Methods inherited from class java.lang.Object:
equals, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Details

Lab2SkaliiKI34

public Lab2SkaliiKI34()

Method Details

main

public static void main(String[] args)
(throws java.lang.Exception)

Стартовий метод майже в кожній програмі

Parameters:
args

Throws:
java.lang.Exception

Рис.4.Вмістиме вкладки Class

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 відкривання 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 ознайомила з базовими конструкціями мови Java та оволоділа навиками написання й автоматичного документування простих консольних програм мовою Java.