

Міністерство освіти і науки України
Національний університет “Львівська політехніка”

Кафедра ЕОМ



ЗВІТ

до лабораторної роботи №1

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Дослідження базових конструкцій мови Java»

Варіант 5

Виконала:

ст. гр. КІ-305

Гринь С.М.

Прийняв:

доц. каф. ЕОМ

Іванов Ю. С.

Львів 2023

Тема: Дослідження базових конструкцій мови Java.

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.

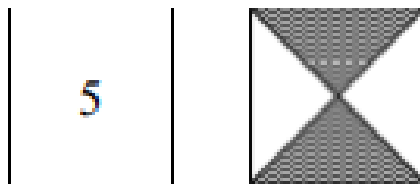
2. Автоматично згенерувати документацію до розробленої програми.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її

виконання та фрагменту згенерованої документації.

4. Дати відповідь на контрольні запитання.

Варіант 5



Код програми

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;

/**
 * Програма для створення квадратної матриці і збереження її у файлі.
 */
public class Lab1HrynKI305 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Запит на введення розміру матриці
        System.out.print("Введіть розмір квадратної матриці: ");
        int n = scanner.nextInt();

        String filler;

        do {

            // Запит на введення символу-заповнювача
            System.out.print("Введіть символ-заповнювач (символ довжиною 1): ");
            filler = scanner.next();
            if (filler.length() != 1) {
                System.out.println("Символ-заповнювач має бути лише одним символом.");
            }
        } while (filler.length() != 1);

        char fillerChar = filler.charAt(0);

        // Створення та заповнення матриці
        char[][] matrix = new char[n][n];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (i >= j && i + j >= n - 1 || i <= j && i + j <= n - 1) {
                    matrix[i][j] = fillerChar;
                }
            }
        }
    }
}
```

```
        } else {

            matrix[i][j] = ' ';

        }

    }

}

try {

    // Запис матриці у файл

    FileWriter writer = new FileWriter("matrix.txt");

    PrintWriter printWriter = new PrintWriter(writer);

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            printWriter.print(matrix[i][j] + " ");

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

        }

        printWriter.println();

        System.out.println();

    }

    printWriter.close();

    System.out.println("Матриця збережена у файлі 'matrix.txt'.");

} catch (IOException e) {

    // Обробка помилок при збереженні матриці у файлі

    System.err.println("Помилка при збереженні матриці у файлі.");

    e.printStackTrace();

}

scanner.close();

}

}
```

Результати роботи програми

```
Run: Lab1HrynKI305 x
C:\Users\User\.jdk\openjdk-19.0.1\bin\java.exe "-javaagent:C:\P
Введіть розмір квадратної матриці: 9
Введіть символ-заповнювач (символ довжиною 1): *
* * * * *
* * * * *
* * * * *
* * *
*
* * *
* * * * *
* * * * * * *
* * * * * * * * *
Матриця збережена у файлі 'matrix.txt'.

Process finished with exit code 0
```

```

1  * * * * *
2  * * * * *
3  * * * * *
4  * * * * *
5  * * * * *
6  * * * * *
7  * * * * *
8  * * * * *
9  * * * * *
10

```

PACKAGE

CLASS

TREE

INDEX

HELP

PACKAGE: DESCRIPTION | RELATED PACKAGES | CLASSES AND INTERFACES

Unnamed Package

Classes

Class	Description
Lab1HrynKI305	Програма для створення квадратної матриці і збереження її у файлі.

PACKAGE

CLASS

TREE

INDEX

HELP

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

Class Lab1HrynKI305

java.lang.Object[Ⓜ]
Lab1HrynKI305

public class Lab1HrynKI305
extends Object[Ⓜ]

Програма для створення квадратної матриці і збереження її у файлі.

Constructor Summary

Constructors

Constructor	Description
Lab1HrynKI305()	

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
static void	main(String [Ⓜ] [] args)	

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

clone[Ⓜ], equals[Ⓜ], finalize[Ⓜ], getClass[Ⓜ], hashCode[Ⓜ], notify[Ⓜ], notifyAll[Ⓜ], toString[Ⓜ], wait[Ⓜ], wait[Ⓜ], wait[Ⓜ]

Constructor Details

Lab1HrynKI305

public Lab1HrynKI305()

PACKAGE	CLASS	TREE	INDEX	HELP
-------------------------	-----------------------	----------------------	-----------------------	----------------------

SUMMARY: NESTED FIELD CONSTR METHOD	DETAIL: FIELD CONSTR METHOD
---	---------------------------------

Constructor Summary

Constructors	Description
<code>Lab1HrynKI305()</code>	

Method Summary

All Methods	Static Methods	Concrete Methods
Modifier and Type	Method	Description
static void	<code>main(String[] args)</code>	

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Details

Lab1HrynKI305
<code>public Lab1HrynKI305()</code>

Method Details

main
<code>public static void main(String[] args)</code>

Це Javadoc коментарі, які використовуються для створення документації API. Вони мають бути розміщені перед класом і можуть містити таку інформацію, як опис класу, автор, версія, параметри конструкторів та методів, інформація про поведінку тощо.

2. Які дескриптори використовуються при коментуванні методів?

Коментарі до методів розташовуються безпосередньо перед методами, які вони описують. Крім дескрипторів загального призначення для коментування методів використовуються дескриптори:

- `@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`.

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 має 8 основних (простих) типів, які не є класами та однаково представляються на будь-якій машині, де виконується програма: `boolean`, `char`, `byte`, `short`, `int`, `long`, `float`, `double`.

5. Як оголосити змінну-масив?

`тип_даних[] ім'я_змінної;`

де `тип_даних` - тип даних, який буде зберігатися в масиві, `ім'я_змінної` – ім'я змінної.

Наприклад, `int[] numbers;`

6. Які керуючі конструкції підтримує Java?

Умовні конструкції: `if`, `else`, `else if`, `switch`.

Циклічні конструкції: `for`, `while`, `do...while`.

Керування потоком виконання: `break`, `continue`.

Виключення: `try`, `catch`, `finally`.

7. В чому різниця між різними варіантами оператора `for`?

- `for` (ініціалізація; умова; інкремент): Традиційний `for`-цикл з ініціалізацією, умовою і кроком інкременту.

- `for` (тип_даних ім'я_змінної : колекція): Розширений `for` (`for-each`) для ітерації через елементи колекції (масиву, списку тощо).
- `for` (ініціалізація; умова; інкремент) { ... }: Звичайний `for` з блоком коду, в якому можна виконувати ітерації.

8. Як здійснити ввід з консолі?

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`:

```
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");`

Зробивши це ми отримаємо доступ до методів класу `PrintWriter`, які призначені для виведення даних простих типів і рядків:

`print` – виводить значення простих типів і рядків у текстовому вигляді;

`write` – призначений для виводу даних типу `char` і `String` у текстовий файл.

Висновок: досліджувала базові конструкції мови Java.