

Міністерство освіти і науки України
Національний університет „Львівська політехніка”
Кафедра ЕОМ



Звіт

до лабораторної роботи № 5
з дисципліни: «Кросплатформні засоби програмування»
«Файли» Варіант - 1

Виконав:
Студент групи КІ-306
Баран В. М.
Прийняв:
Іванов Ю. С.

Львів 2023

Мета: Оволодіти навичками використання засобів мови Java для роботи з потоками і файлами.

ЗАВДАННЯ

1. Створити клас, що реалізує методи читання/запису у текстовому і двійковому форматах результатів роботи класу, що розроблений у лабораторній роботі №5. Написати програму для тестування коректності роботи розробленого класу.
2. Для розробленої програми згенерувати документацію.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагмент згенерованої документації.
4. Дати відповідь на контрольні запитання.

Варіант завдання:

1. $y = \text{tg}(x)$

Код програми:

File App.java

```
package org.example;

import java.io.*;
import java.util.Scanner;

/**
 * Class App
 * @version 1.0
 */
public class App {
    public static void main(String[] args) throws IOException {
        CalcWFio obj = new CalcWFio();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter data: ");
        double data = s.nextDouble();
        obj.calculate(data);
        System.out.println("Result is: " + obj.getResult());
        obj.writeResTxt("textRes.txt");
        obj.writeResBin("BinRes.bin");
        obj.readResBin("BinRes.bin");
        System.out.println("Result is: " + obj.getResult());
        obj.readResTxt("textRes.txt");
        System.out.println("Result is: " + obj.getResult());
    }
}
```

```
package org.example;

import java.io.*;
import java.util.Scanner;

class CalcWFio {
    private double result;

    public void calculate(double x) {
        double radians = Math.toRadians(x); // Переводимо градуси в радіани
        result = Math.tan(radians);
    }

    public double getResult() {
        return result;
    }

    public void writeResTxt(String fName) {
        try (PrintWriter f = new PrintWriter(fName)) {
            f.printf("%f", result);
        } catch (FileNotFoundException ex) {
            System.out.print(ex.getMessage());
        }
    }

    public void readResTxt(String fName) {
        try {
            File f = new File(fName);
            if (f.exists()) {
                try (Scanner s = new Scanner(f)) {
                    result = s.nextDouble();
                }
            } else {
                throw new FileNotFoundException("File " + fName + " not found");
            }
        } catch (FileNotFoundException ex) {
            System.out.print(ex.getMessage());
        }
    }

    public void writeResBin(String fName) {
        try (DataOutputStream f = new DataOutputStream(new FileOutputStream(fName))) {
            f.writeDouble(result);
        } catch (IOException ex) {
            System.out.print(ex.getMessage());
        }
    }

    public void readResBin(String fName) {
        try (DataInputStream f = new DataInputStream(new FileInputStream(fName))) {
            result = f.readDouble();
        } catch (IOException ex) {
            System.out.print(ex.getMessage());
        }
    }
}
```

Результат виконання програми:

```
Enter data: 50
Result is: 1.19175359259421
Result is: 1.19175359259421
Result is: 1.191754
```

Фрагмент згенерованої документації

PACKAGECLASSTREEINDEXHELP

SUMMARY: NESTED | FIELD | CONSTR | METHODDETAIL: FIELD | CONSTR | METHODSEARCH

Package org.example

Class App

java.lang.Object[Ⓜ]
org.example.App

public class App
extends Object[Ⓜ]

Class App

Constructor Summary

Constructors

Constructor	Description
App()	

Method Summary

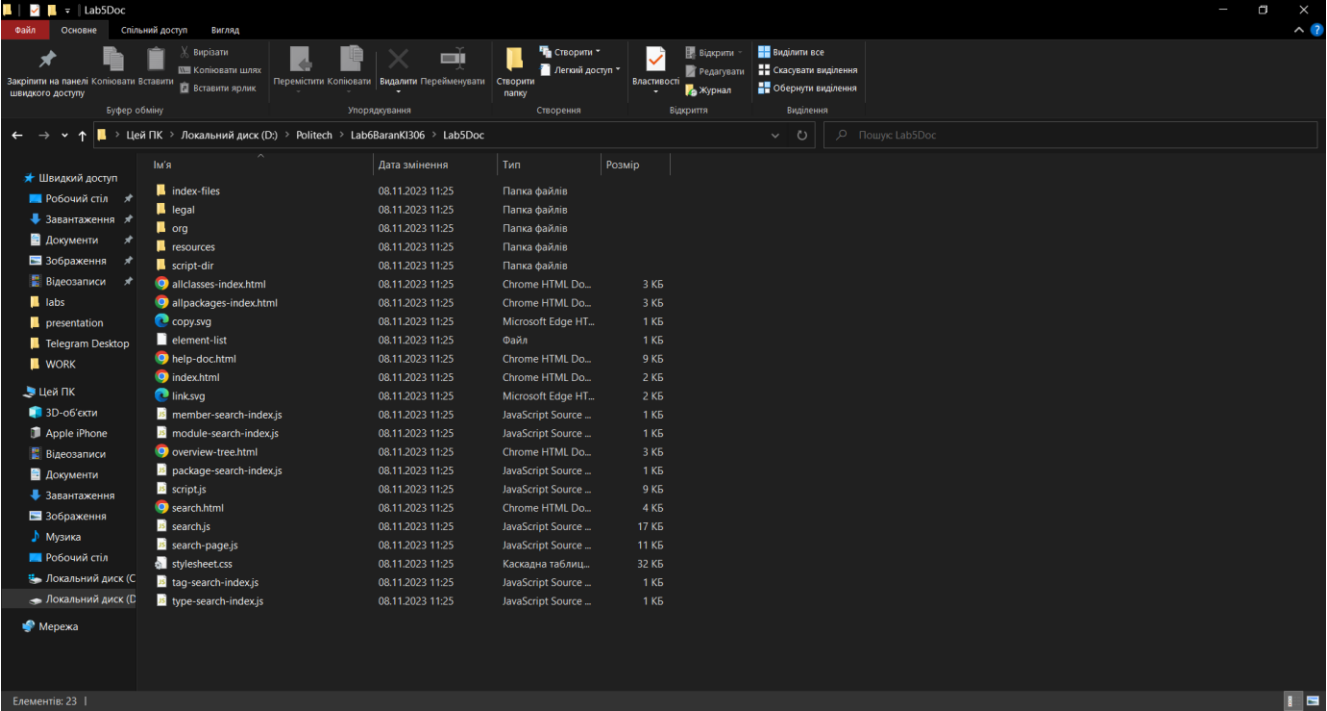
All MethodsStatic MethodsConcrete Methods

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

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

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

Constructor Details



Відповіді на контрольні запитання

1. Розкрийте принципи роботи з файловою системою засобами мови Java.

- Для читання і запису файлів використовуються класи, які успадковуються від `InputStream` і `OutputStream` для байтового рівня та

Reader і Writer для текстового рівня.

2. Охарактеризуйте клас Scanner.

- Scanner в Java використовується для зчитування вхідних даних, включаючи рядки, числа та інші типи даних з різних джерел, таких як стандартний ввід, файли або рядки

3. Наведіть приклад використання класу Scanner.

- Scanner scanner = new Scanner(System.in);
System.out.print("Enter your name: "); String name = scanner.nextLine();
System.out.println("Hello, " + name + "!");

4. За допомогою якого класу можна здійснити запис у текстовий потік?

- Для запису в текстовий потік використовується клас PrintWriter.

5. Охарактеризуйте клас PrintWriter.

- Клас PrintWriter використовується для запису даних у текстовий потік та надає

методи для виводу рядків та інших типів даних у текстовому форматі.

6. Розкрийте методи читання/запису двійкових даних засобами мови Java.

- Для читання та запису двійкових даних використовують класи
DataInputStream
та DataOutputStream.

7. Призначення класів DataInputStream і DataOutputStream.

- Клас DataInputStream використовується для зчитування примітивних типів даних з байтового потоку.

- Клас DataOutputStream використовується для запису примітивних типів даних у байтовий потік.

8. Який клас мови Java використовується для здійснення довільного доступу до файлів.

- Для довільного доступу до файлів використовується клас
RandomAccessFile,
який дозволяє читати та записувати дані в будь-яку позицію файлу без необхідності читати або записувати дані послідовно.

9. Охарактеризуйте клас RandomAccessFile.

- Для довільного доступу до файлів використовується клас
RandomAccessFile, який дозволяє читати та записувати дані в будь-яку позицію файлу.

10. Який зв'язок між інтерфейсом DataOutput і класом DataOutputStream? -

Клас DataOutputStream реалізує інтерфейс DataOutput.

Інтерфейс DataOutput визначає методи для запису примітивних типів даних у байтовий потік.

Клас `DataOutputStream` надає реалізацію цих методів для запису даних у бінарний формат.

Висновок

У ході виконання даної лабораторної роботи, я отримала навички роботи з засобами мови програмування Java для роботи з потоками і файлами. Ознайомившись з концепцією потоків, я змогла створювати та керувати паралельними виконавчими процесами у моїх програмах. Крім того, я вивчила методи для взаємодії з файловою системою, зчитування та запису даних в текстові файли.