Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



**Звіт**

з лабораторної роботи № 5-6

**з дисципліни:** «Кросплатформенні засоби програмування»

**на тему:** «Файли та параметризоване програмування»

**Виконав:**

студент групи КІ-302

Козак О.П.

**Прийняв:**

Майдан М.В.

**Львів – 2023**

**Мета роботи:** оволодіти навиками використання засобів мови Java для роботи з потоками і файлами.

**Завдання ( Варіант 10 )**

1. Створити клас, що реалізує методи читання/запису у текстовому і двійковому

форматах результатів роботи класу, що розроблений у лабораторній роботі No5.

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.

**Вихідний код програми**

class ExpressionCalculator<T> {

    private x: T;

    constructor(x: T) {

        this.x = x;

    }

    calculateExpression(): number {

        const tanX = Math.tan(Number(this.x));

        const cotanX = 1 / Math.tan(Number(this.x));

        if (Number.isNaN(tanX) || Number.isNaN(cotanX) || Math.abs(cotanX) < 1e-6 || Number(this.x) % 3.14 === 0) {

            throw new Error("Error calculating: the expression is undefined or dividing by zero.");

        }

        return tanX / cotanX;

    }

    saveResultToFile(textFile: string, binaryFile: string): void {

        const result = this.calculateExpression();

        const resultText = `Result of the expression calculation: ${result}`;

        const textBlob = new Blob([resultText], { type: 'text/plain' });

        const textUrl = URL.createObjectURL(textBlob);

        const textLink = document.createElement('a');

        textLink.href = textUrl;

        textLink.download = textFile;

        textLink.click();

        const binaryBlob = new Blob([new Float64Array([result]).buffer], { type: 'application/octet-stream' });

        const binaryUrl = URL.createObjectURL(binaryBlob);

        const binaryLink = document.createElement('a');

        binaryLink.href = binaryUrl;

        binaryLink.download = binaryFile;

        binaryLink.click();

        const resultTextElement = document.getElementById("resultText");

        if (resultTextElement) {

            resultTextElement.textContent = resultText;

        }

    }

}

function calculateAndSave() {

    const xInput = document.getElementById("inputX") as HTMLInputElement;

    const x = parseFloat(xInput.value);

    if (isNaN(x)) {

        alert("Please enter a valid value for x.");

        return;

    }

    const calculator = new ExpressionCalculator(x);

    try {

        calculator.saveResultToFile("output.txt", "output.bin");

    } catch (error) {

        alert("Error: " + error.message);

    }

}

**Відповіді на контрольні запитання**

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

Принципи роботи з файловою системою в Java базуються на використанні класів, таких як File, FileInputStream, FileOutputStream, BufferedReader, BufferedWriter, Scanner, PrintWriter, RandomAccessFile, і так далі. Ці класи дозволяють читати та записувати дані в файли, виконувати операції з каталогами, перевіряти доступність файлів, видаляти та переміщати файли, створювати нові файли тощо.

1. Охарактеризуйте клас Scanner.

Клас Scanner є інструментом для зчитування різноманітних типів даних з різних джерел, таких як консоль, файли та рядки.

1. Наведіть приклад використання класу Scanner.

Scanner scanner = new Scanner(System.in);

System.out.print("Введіть ціле число: ");

int number = scanner.nextInt();

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

Запис у текстовий потік можна здійснити за допомогою класу PrintWriter.

1. Охарактеризуйте клас PrintWriter.

Клас PrintWriter представляє потік для запису символьних даних у текстовий файл.

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

Для читання і запису двійкових даних можна використовувати класи FileInputStream і FileOutputStream для потокового читання і запису байтів, або класи DataInputStream і DataOutputStream для роботи з примітивними типами даних.

1. Призначення класів DataInputStream і DataOutputStream.

Класи DataInputStream і DataOutputStream використовуються для читання та запису примітивних типів даних у двійковому форматі.

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

Для здійснення довільного доступу до файлів можна використовувати клас RandomAccessFile.

1. Охарактеризуйте клас RandomAccessFile.

Клас RandomAccessFile надає можливість зчитувати та записувати дані у файлі з довільним доступом, тобто переміщатися у файлі та читати/писати дані з будь-якої позиції.

1. Який зв’язок між інтерфейсом DataOutput і класом DataOutputStream?

Інтерфейс DataOutput визначає методи для запису примітивних даних у двійковому форматі. Клас DataOutputStream реалізує цей інтерфейс і дозволяє записувати дані у двійковому форматі до потоку.

**Висновок**

Ознайомився з використанням потоків та написав клас що отримує та записує дані з файлу та записує у форматі двійковому та текстовому. Розробив програму драйвер яка використовує даний клас.