Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ

Звіт

З лабораторної роботи №5

Варіант – 10

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Файли»

Виконав: ст. гр. КІ-306

Згурський Т.С.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

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

Main.java:

```
package org.example;


import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

/**
 * Клас Main для обчислення та запису результатів в файл з вхідного файлу.
 * @version 1.0
 */
public class Main {
    /**
     * Головний метод програми.
     *
     * @param args Масив аргументів командного рядка. В даному випадку не використовується.
     * @throws IOException Виникає, якщо сталася помилка при роботі з файлами.
     */
    public static void main(String[] args) throws IOException {
        // Ім'я вхідного та вихідного файлів
        String inputFileName = "input.txt";
        String outputFileName = "output.txt";
        String outputBinaryFileName = "output";

        try (BufferedReader reader = new BufferedReader(new FileReader(inputFileName))) {
            String line = reader.readLine();
            double x = Double.parseDouble(line);

            // Створення об'єкта калькулятора для обчислення виразу з параметром x
            ExpressionCalculator calculator = new ExpressionCalculator(x);

            try {
                // Обчислення виразу та збереження результату у вихідний файл
                double result = calculator.calculateExpression();
                calculator.saveResultToFile(result, outputFileName, outputBinaryFileName);
                System.out.println("Результат обчислення: " + result);
            } catch (ArithmeticException e) {
```

```
            System.err.println("Помилка обчислення: " + e.getMessage());
        } catch (IOException e) {
            System.err.println("Помилка при записі до файлу: " + e.getMessage());
        }
    } catch (IOException e) {
        System.err.println("Помилка при читанні з файлу: " + e.getMessage());
    }
  }
}
```

ExpressionCalculator.java:

```
package org.example;

import java.io.*;
/**
 * Клас для обчислення виразу та збереження результату у файл.
 */
public class ExpressionCalculator {
  private double x;
  /**
   * Конструктор для створення об'єкта ExpressionCalculator зі значенням x.
   *
   * @param x Значення x, для якого буде обчислюватися вираз.
   */
  public ExpressionCalculator(double x) {
    this.x = x;
  }

  /**
   * Обчислює вираз y = tg(x) / ctg(x).
   *
   * @return Результат обчислення виразу.
   * @throws ArithmeticException Виникає, якщо виникає помилка при обчисленні виразу.
   */
  public double calculateExpression() throws ArithmeticException {
    double tanX = Math.tan(x);
    double cotanX = 1 / Math.tan(x);

    if (Double.isInfinite(tanX) || Double.isNaN(tanX) || Double.isInfinite(cotanX) || Double.isNaN(cotanX)) {
      throw new ArithmeticException("Вираз не визначений (tg(x) або ctg(x) мають недопустиме
значення).");
    }

    if (Math.abs(cotanX) < 1e-6) {
      throw new ArithmeticException("Ділення на нуль у виразі (ctg(x) дуже мале).");
    }

    return tanX / cotanX;
  }

  /**
   * Метод для збереження результату обчислення у файл у текстовому і двійковому форматах.
   *
```

```
     * @param result      Результат обчислення виразу.
     * @param textFile    Ім'я файлу для текстового формату.
     * @param binaryFile  Ім'я файлу для двійкового формату.
     * @throws IOException Виникає, якщо сталася помилка при записі у файл.
     */
    public void saveResultToFile(double result, String textFile, String binaryFile) throws IOException {
        // Запис у текстовий файл
        try (PrintWriter textWriter = new PrintWriter(new FileWriter(textFile))) {
            textWriter.println("Результат обчислення виразу: " + result);
        }

        // Запис у двійковий файл
        try (DataOutputStream binaryWriter = new DataOutputStream(new FileOutputStream(binaryFile))) {
            binaryWriter.writeDouble(result);
        }
    }
}
```

Результат виконання програми

```
Результат обчислення: 0.08468460342425725
```

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.
2. Охарактеризуйте клас Scanner.
   Клас Scanner є інструментом для зчитування різноманітних типів даних з різних джерел, таких як консоль, файли та рядки.
3. Наведіть приклад використання класу Scanner.
   Scanner scanner = new Scanner(System.in);
   System.out.print("Введіть ціле число: ");
   int number = scanner.nextInt();
4. За допомогою якого класу можна здійснити запис у текстовий потік?
   Запис у текстовий потік можна здійснити за допомогою класу PrintWriter.
5. Охарактеризуйте клас PrintWriter.
   Клас PrintWriter представляє потік для запису символьних даних у текстовий файл.
6. 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.
7. Призначення класів DataInputStream і DataOutputStream.
   Класи DataInputStream і DataOutputStream використовуються для читання та запису примітивних типів даних у двійковому форматі.
8. Який клас мови Java використовується для здійснення довільного доступу до файлів.
   Для здійснення довільного доступу до файлів можна використовувати клас RandomAccessFile.
9. 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.

10. 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.