

Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



Звіт

З лабораторної роботи №5

Варіант – 3

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Файли у Java»

Виконав: ст. гр. КІ-306

Братівник Д. А.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2023

Мета роботи: оволодіти навиками 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 №4. Написати програму для тестування коректності роботи розробленого класу.
2. Для розробленої програми згенерувати документацію.
3. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її

виконання та фрагменту згенерованої документації та завантажити його у ВНС.

5. Дати відповідь на контрольні запитання.

Варіант завдання: $y = \sin(x)/\cos(x)$

Код програми:

Файл EquationsWithIO.java:

```
package org.example;

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

/**
 * Клас EquationsWithIO містить методи для роботи з рівняннями та
 * введенням/виведенням результатів у текстовій та бінарній форматах.
 */
public class EquationsWithIO {
    /**
     * Записує результат обчислення рівняння у текстовий файл.
     *
     * @param fName ім'я файлу, у який записується результат
     * @param x значення для обчислення рівняння
     * @throws FileNotFoundException виникає, якщо файл не
     * знайдено
     */
    public void writeResultTxt(String fName, int x) throws
    FileNotFoundException {
        PrintWriter f = new PrintWriter(fName);
        Equations eq = new Equations();
        try {
            double result = eq.calculate(x);
            f.printf("%f", result);
        } catch (CalcException ex) {
```

```

        f.print(ex.getMessage());
    }
    f.close();
}

/**
 * Записує результат обчислення рівняння у бінарний файл.
 *
 * @param fName ім'я файлу, у який записується результат
 * @param x значення для обчислення рівняння
 * @throws IOException виникає, якщо сталася помилка
вводу/виводу
 */
public void writeResultBin(String fName, int x) throws
IOException {
    Equations eq = new Equations();
    try (DataOutputStream f = new DataOutputStream(new
FileOutputStream(fName))) {
        double result = eq.calculate(x);
        f.writeDouble(result);
    } catch (CalcException ex) {
        System.out.println("Exception reason: " +
ex.getMessage());
    }
}

/**
 * Читає результат обчислення рівняння з бінарного файлу та
виводить його на консоль.
 *
 * @param fName ім'я файлу, з якого читається результат
 * @throws IOException виникає, якщо сталася помилка
вводу/виводу
 */
public void readResultBin(String fName) throws IOException {
    try (DataInputStream f = new DataInputStream(new
FileInputStream(fName))) {
        double result = f.readDouble();
        System.out.println("Result from binary file: " +
result);
    }
}

/**
 * Читає результат обчислення рівняння з текстового файлу та
виводить його на консоль.
 *
 * @param fName ім'я файлу, з якого читається результат
 * @throws FileNotFoundException виникає, якщо файл не
знайдено
 */
public void readResultTxt(String fName) throws
FileNotFoundException {
    File file = new File(fName);

```

```

        if (file.exists()) {
            try (Scanner scanner = new Scanner(file)) {
                if (scanner.hasNextDouble()) {
                    double result = scanner.nextDouble();
                    System.out.println("Result from text file: " +
result);
                } else {
                    System.out.println(scanner.nextLine());
                }
            }
        } else {
            throw new FileNotFoundException("File not found: " +
fName);
        }
    }

    /**
     * Точка входу у програму. Демонструє використання методів
класу EquationsWithIO.
     */
    * @param args аргументи командного рядка
    */
    public static void main(String[] args) {
        EquationsWithIO equationsWithIO = new EquationsWithIO();

        try {
            // Запис результату у текстовий файл
            equationsWithIO.writeResultTxt("result.txt", 90);
            // Запис результату у бінарний файл
            equationsWithIO.writeResultBin("result.bin", 45);
            // Читання результату з бінарного файлу
            equationsWithIO.readResultBin("result.bin");
            // Читання результату з текстового файлу
            equationsWithIO.readResultTxt("result.txt");
        } catch (IOException e) {
            System.out.println("IO Exception: " + e.getMessage());
        }
    }
}

```

Файл CalcException.java:

```

package org.example;

/**
 * Клас CalcException є підкласом ArithmeticException та
використовується для представлення винятків, пов'язаних з
обчисленнями.
 */
class CalcException extends ArithmeticException {
    /**
     * Конструктор за замовчуванням.
     */
    public CalcException() {}
}

```

```

    /**
     * Конструктор, який приймає рядок, який вказує на причину
    винятку.
     *
     * @param cause рядок, який містить причину винятку
     */
    public CalcException(String cause) {
        super(cause);
    }
}

/**
 * Клас Equations містить метод calculate для обчислення значення
    рівняння.
 */
class Equations {
    /**
     * Обчислює значення рівняння для заданого кута.
     *
     * @param x кут, для якого обчислюється рівняння
     * @return результат обчислення рівняння
     * @throws CalcException виникає, якщо сталася помилка під час
    обчислення рівняння
     */
    public double calculate(int x) throws CalcException {
        double y, rad;
        rad = x * Math.PI / 180.0;

        try {
            y = Math.sin(rad) / Math.cos(rad);
            if (Double.isNaN(y) || Double.isInfinite(y) || x == 90
            || x == -90) {
                throw new ArithmeticException();
            }
        } catch (ArithmeticException ex) {
            if (rad == Math.PI / 2.0 || rad == -Math.PI / 2.0) {
                throw new CalcException("Exception reason: Illegal
            value of X for tangent calculation");
            } else if (x == 0) {
                throw new CalcException("Exception reason: X =
            0");
            } else {
                throw new CalcException("Unknown reason of the
            exception during expression calculation");
            }
        }

        return y;
    }
}

```

Результати роботи програми:

```
/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=60246:/Applications/IntelliJ I
Result from binary file: 0.9999999999999999
Exception reason: Illegal value of X for tangent calculation

Process finished with exit code 0
```

Фрагмент згенерованої документації

doclab2

HTML

allclasses-index.html

HTML

allpackages-index.html

copy.svg

copy.svg

element-list

HTML

help-doc.html

index-files

HTML

index.html

legal

JS

member-search-index.js

JS

module-search-index.js

org

HTML

overview-tree.html

JS

package-search-index.js

resources

script-dir

JS

script.js

JS

search-page.js

HTML

search.html

JS

search.js

stylesheet.css

stylesheet.css

JS

tag-search-index.js

JS

type-search-index.js

PACKAGE

CLASS

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

SEARCH

Package org.example

Class EquationsWithIO

java.lang.Object[?]
org.example.EquationsWithIO

Inheritance Tree

public class EquationsWithIO
extends Object[?]

Клас EquationsWithIO містить методи для роботи з рівняннями та введенням/виведенням результатів у текстовій та бінарній форматах.

Constructor Summary

Constructors

Constructor	Description
EquationsWithIO()	

Method Summary

All Methods

Static Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
static void	main(String [?] [] args)	Точка входу у програму.
void	readResultBin(String [?] fName)	Читає результат обчислення рівняння з бінарного файлу та виводить його на консоль.
void	readResultTxt(String [?] fName)	Читає результат обчислення рівняння з текстового файлу та виводить його на консоль.
void	writeResultBin(String [?] fName, int x)	Записує результат обчислення рівняння у бінарний файл.
void	writeResultTxt(String [?] fName, int x)	Записує результат обчислення рівняння у текстовий файл.

Methods inherited from class java.lang.Object[?]
equals[?], getClass[?], hashCode[?], notify[?], notifyAll[?], toString[?], wait[?], wait[?], wait[?]

Відповіді на контрольні запитання

1. Розкрийте принципи роботи з файловою системою засобами мови Java.
 - Для читання і запису файлів використовуються класи, які

успадковуюються 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 отримав навички роботи з засобами мови програмування Java для роботи з потоками і файлами. Ознайомившись з концепцією потоків, я зміг створювати та керувати паралельними виконавчими процесами у моїх програмах. Крім того, я вивчив методи для взаємодії з файловою системою, зчитування та запису даних в текстові файли.