Міністерство освіти і науки України

Національний університет «Львівська політехніка»



Звіт

до лабораторної роботи №5

з дисципліни: “Кросплатформні засоби програмування”

на тему: «Файли у Java»

Варіант №12

Виконала:

ст.гр. КІ-307

Коваль К. О.

Прийняв:

Іванов Ю. С.

**Львів 2023**

**Мета:** 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:

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

2. Охарактеризуйте клас Scanner.

3. Наведіть приклад використання класу Scanner.

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

5. Охарактеризуйте клас PrintWriter.

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

7. Призначення класів DataInputStream і DataOutputStream.

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

9. Охарактеризуйте клас RandomAccessFile.

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

**Варіант 12: y=sin(x)/tg(4x)**

**Виконання:**

package KI307.Koval.Lab5;

import java.util.Scanner;

import java.io.\*;

import static java.lang.System.out;

/\*\*

\* Class <code>Lab5KovalKI307</code> Implements driver for Equations class

\* @author Koval Kateryna

\* @version 1.2

\*/

public class Lab5KovalKI307 {

/\*\*

\* @param args

\* @throws IOException

\*/

public static void main(String[] args) throws IOException {

try

{

CalcWFio obj = new CalcWFio();

Scanner in = new Scanner(System.in);

try

{

try

{

Equations eq = new Equations();

out.print("Enter X: ");

double data = eq.calculate(in.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 from bin is: " + obj.getResult());

obj.readResTxt("textRes.txt");

System.out.println("Result from text is: " + obj.getResult());

}

finally

{

// Цей блок виконається за будь-яких обставин

}

}

catch (CalcException ex)

{

// Блок перехоплює помилки обчислень виразу

out.print(ex.getMessage());

}

in.close();

}

catch (FileNotFoundException ex)

{

// Блок перехоплює помилки роботи з файлом навіть якщо вони

// виникли у блоці finally

out.print("Exception reason: Perhaps wrong file path");

}

}

}

/\*\*

\* Class <code>CalcException</code> more precises ArithmeticException

\* @author Koval Kateryna

\* @version 1.2

\*/

class CalcException extends ArithmeticException

{

public CalcException(){}

public CalcException(String cause)

{

super(cause);

}

}

/\*\*

\* Class <code>Equations</code> implements method for (sin(x)/tg(4x)) expression calculation

\* @author Koval Kateryna

\*

\* @version 1.2

\*/

class Equations

{

/\*\*

\* Method calculates the (sin(x)/tg(4x)) expression

\* @param <code>x</code> Angle in degrees

\* @throws CalcException

\*/

public double calculate(double x) throws CalcException

{

double y, rad;

rad = x \* Math.PI / 180.0;

try

{

y = Math.sin(rad)/Math.tan(4\*rad);

// Якщо результат не є числом, то генеруємо виключення

if (y==Double.NaN || y==Double.NEGATIVE\_INFINITY ||

y==Double.POSITIVE\_INFINITY || x==90 || x== -90)

throw new ArithmeticException();

}

catch (ArithmeticException ex)

{

// створимо виключення вищого рівня з поясненням причини

// виникнення помилки

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 exception calculation");

}

return y;

}

}

/\*\*

\* Class <code>CalcWFio</code> work with binary and text files expression calculation

\* @author Koval Kateryna

\* @version 1.2

\*/

class CalcWFio {

/\*\*

\* Method write result to text file

\*

\* @param <code>fName</code> Name of the file

\* @throws

\*/

public void writeResTxt(String fName) throws FileNotFoundException {

PrintWriter f = new PrintWriter(fName);

f.printf("%f ", result);

f.close();

}

/\*\*

\* Method read result from text file

\*

\* @param <code>fName</code> Name of the file

\* @throws

\*/

public void readResTxt(String fName) {

try {

File f = new File(fName);

if (f.exists()) {

Scanner s = new Scanner(f);

result = s.nextDouble();

s.close();

} else

throw new FileNotFoundException("File " + fName + "not found");

} catch (FileNotFoundException ex) {

System.out.print(ex.getMessage());

}

}

/\*\*

\* Method write result to binary file

\*

\* @param <code>fName</code> Name of the file

\* @throws

\*/

public void writeResBin(String fName) throws FileNotFoundException, IOException {

DataOutputStream f = new DataOutputStream(new FileOutputStream(fName));

f.writeDouble(result);

f.close();

}

/\*\*

\* Method read result from binary file

\*

\* @param <code>fName</code> Name of the file

\* @throws

\*/

public void readResBin(String fName) throws FileNotFoundException, IOException {

DataInputStream f = new DataInputStream(new FileInputStream(fName));

result = f.readDouble();

f.close();

}

/\*\*

\* Method set x for result

\*/

public void calculate(double x) {

result = x;

}

/\*\*

\* Method return result

\*/

public double getResult() {

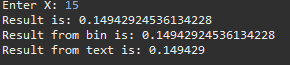
return result;

}

private double result;

}

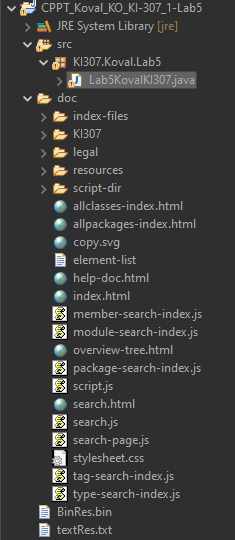
**Результат:**

****

****

****

**Документація:**

****

**Відповіді на контрольні запитання:**

1. \*\*Принципи роботи з файловою системою в Java:\*\*

- Використання класів `File` для представлення файлів та каталогів.

- Для читання та запису використовуються класи, такі як `FileReader`, `FileWriter`, `BufferedReader`, і `BufferedWriter`.

- Для операцій з бінарними даними використовуються класи `FileInputStream` та `FileOutputStream`.

2. \*\*Характеристика класу Scanner:\*\*

- `Scanner` — це клас в Java, що дозволяє здійснювати введення з різних джерел, таких як клавіатура, файли, рядки тощо.

- Забезпечує можливість читати дані різних типів.

3. \*\*Приклад використання класу Scanner:\*\*

```java

import java.util.Scanner;

public class ScannerExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter your name: ");

String name = scanner.nextLine();

System.out.println("Hello, " + name + "!");

scanner.close();

}

}

```

4. \*\*Клас для запису у текстовий потік:\*\*

- `FileWriter` або `BufferedWriter` для текстового запису.

5. \*\*Характеристика класу PrintWriter:\*\*

- `PrintWriter` — це клас в Java, який надає методи для виводу форматованого тексту у потік виведення.

6. \*\*Методи читання/запису двійкових даних в Java:\*\*

- Для читання: `FileInputStream`, `DataInputStream`, `BufferedInputStream`.

- Для запису: `FileOutputStream`, `DataOutputStream`, `BufferedOutputStream`.

7. \*\*Призначення класів DataInputStream і DataOutputStream:\*\*

- `DataInputStream` та `DataOutputStream` — це класи, які надають методи для читання/запису примітивних типів даних та рядків.

8. \*\*Клас для довільного доступу до файлів:\*\*

- `RandomAccessFile`.

9. \*\*Характеристика класу RandomAccessFile:\*\*

- `RandomAccessFile` — це клас в Java, який дозволяє здійснювати довільний доступ до файлу для читання та запису.

10. \*\*Зв’язок між інтерфейсом DataOutput і класом DataOutputStream:\*\*

- Клас `DataOutputStream` реалізує інтерфейс `DataOutput`.

- `DataOutput` визначає методи для запису примітивних типів даних та рядків в потік виведення.

**Висновок:** на даній лабораторній роботі я оволоділа навиками використання засобів мови Java для роботи з потоками і файлами.