Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи №6

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Файли»

**Виконав:**

студент групи КІ-35

Береговський І. С.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2022

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

**Завдання (варіант № 28)**

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: • Розкрийте принципи роботи з файловою системою засобами мови Java. • Охарактеризуйте клас Scanner. • Наведіть приклад використання класу Scanner. • За допомогою якого класу можна здійснити запис у текстовий потік? • Охарактеризуйте клас PrintWriter. • Розкрийте методи читання/запису двійкових даних засобами мови Java. • Призначення класів DataInputStream і DataOutputStream. • Який клас мови Java використовується для здійснення довільного доступу до файлів. • Охарактеризуйте клас RandomAccessFile. • Який зв’язок між інтерфейсом DataOutput і класом DataOutputStream?

**Текст програми**

**CalcException.java**

package ki.berehovskyi.lab5;

class CalcException extends ArithmeticException

{

public CalcException(){}

public CalcException(String cause)

{

super(cause);

}

}

**FioApp.java**

package ki.berehovskyi.lab6;

import java.io.\*;

import java.util.\*;

public class FioApp {

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

boolean IsCatched = true;

CalcWFio obj = new CalcWFio();

Scanner s = new Scanner(System.in);

System.out.print("Enter data: ");

double data = s.nextDouble();

try

{

obj.calculate(data);

}

catch (CalcException ex)

{

IsCatched = false;

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

}

if(IsCatched) {

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());

}

}

}

**CalcWFio.java**

package ki.berehovskyi.lab6;

import java.io.\*;

import java.util.\*;

public class CalcWFio {

private double result, convToRad;

public void writeResTxt(String fName) throws FileNotFoundException

{

PrintWriter f = new PrintWriter(fName);

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

f.close();

}

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());

}

}

public void writeResBin(String fName) throws FileNotFoundException, IOException

{

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

f.writeDouble(result);

f.close();

}

public void readResBin(String fName) throws FileNotFoundException, IOException

{

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

result = f.readDouble();

f.close();

}

public void calculate(double x) throws CalcException

{

convToRad = Math.toRadians(2\*x);

try {

result = Math.tan(convToRad);

if (result==Double.NaN || result==Double.NEGATIVE\_INFINITY || result==Double.POSITIVE\_INFINITY || x\*2==90 || x\*2== -90)

throw new ArithmeticException();

}

catch(ArithmeticException ex){

if (convToRad == Math.PI/2.0 || convToRad == -Math.PI/2.0)

throw new CalcException("Exception reason: Illegal value of X for tangent calculation");

else

throw new CalcException("Unknown reason of the exception during exception calculation");

}

}

public double getResult()

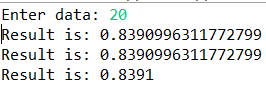
{

return result;

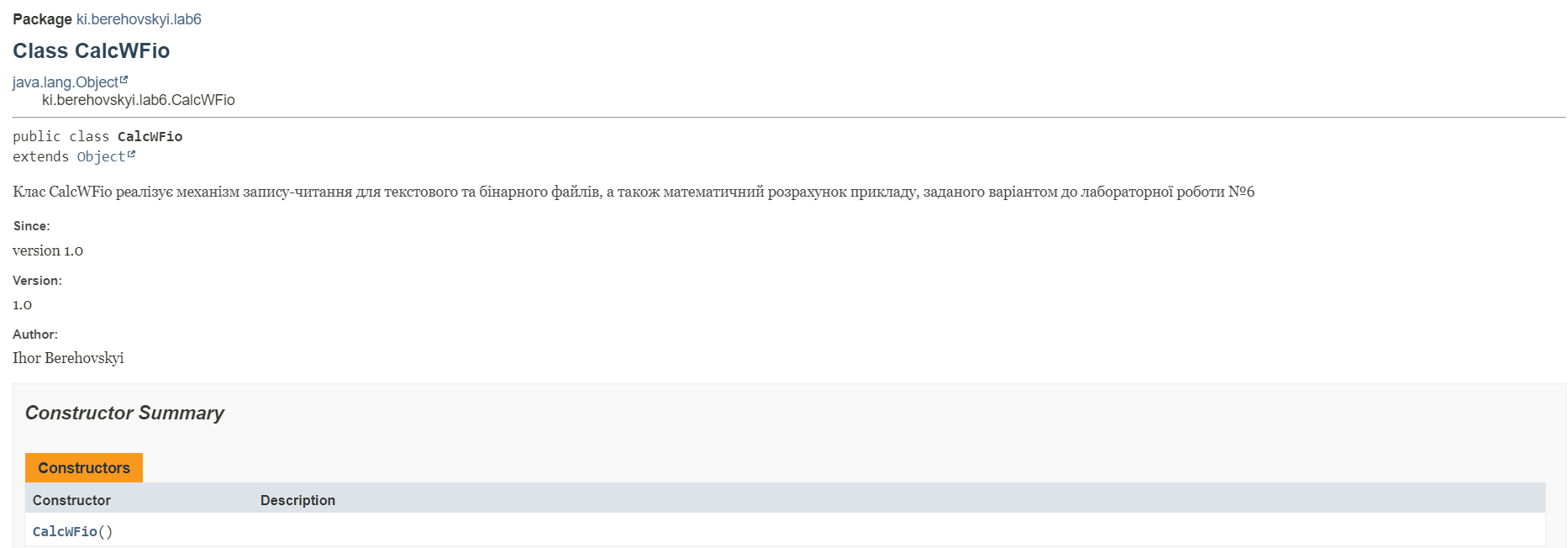
}

}

**Результат виконання програми**



**Фрагмент згенерованої документації**



**Відповіді на контрольні запитання**

• Охарактеризуйте клас Scanner.

Він найкраще підходить для читання текстових потоків. На відміну від InputStreamReader і FileReader, що дозволяють лише читати текст, він має велику кількість методів, які здатні читати як рядки, так і окремі примітивні типи з подальшим їх перекодуванням до цих типів, робити шаблонний аналіз текстового потоку, здатний працювати без потоку даних та ще багато іншого.

**Висновок**

На даній лабораторній роботі я створив клас, що реалізує методи читання/запису у текстовому і двійковому форматах результатів роботи класу, що розроблений у лабораторній роботі №5. У кінці роботи виявив, що даний код досить гнучкий і його можна у подальшому використовувати для реалізації читання/запису в інших проектах на мові Java. Також, ознайомившись з методичними вказівками, я повторно закріпив вивчений матеріал.