Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра МКС



**Звіт**

З лабораторної роботи №6

З дисципліни: «Кросплатформні засоби програмування»

На тему:

«Файли»

Виконав: ст.гр. КІ-35

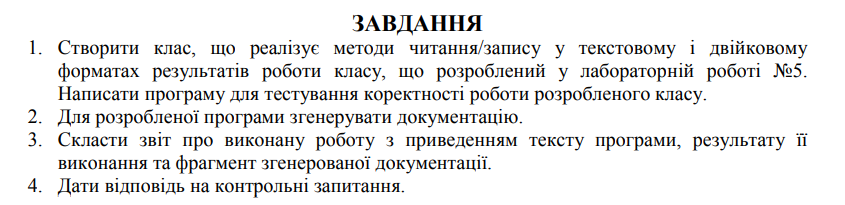
Куденчук Владислав

Прийняв:

Іванов Ю. С.

**Львів 2022**

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





Виконання:

Код:

**Class CalcExeption**

package KI\_35.Kudenchuk.Lab6;  
  
*/\*\*  
 \* Class <code>CalcException</code> more precises ArithmeticException  
 \** ***@author*** *EOM Stuff  
 \** ***@version*** *1.0  
 \*/*class CalcException extends ArithmeticException  
{  
 public CalcException(){}  
  
 public CalcException(String cause)  
 {  
 super(cause);  
 }  
}

**Class CalcWFio**

package KI\_35.Kudenchuk.Lab6;  
  
import java.io.\*;  
import java.util.Scanner;  
  
class CalcWFio extends CalcException  
{  
 public static void writeResTxt(String fName, String data) throws FileNotFoundException  
 {  
  
// PrintWriter f = new PrintWriter(fName);  
  
// RandomAccessFile f = new RandomAccessFile(fName, "rw");  
 RandomAccessFile f = new RandomAccessFile(fName, "rw");  
 try {  
 f.write(data.getBytes());  
 f.close();  
 }  
 catch (IOException e) {  
 e.printStackTrace();  
 }  
// f.printf(data);  
// 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 IOException  
 {  
 DataOutputStream f = new DataOutputStream(new FileOutputStream(fName));  
 f.writeDouble(result);  
 f.close();  
 }  
 public void readResBin(String fName) throws IOException  
 {  
 DataInputStream f = new DataInputStream(new FileInputStream(fName));  
 result = f.readDouble();  
 f.close();  
 }  
 public void calculate(double x)  
 {  
 result = Equations.*calculate*(x);  
 }  
 public double getResult()  
 {  
 return result;  
 }  
 private double result;  
}

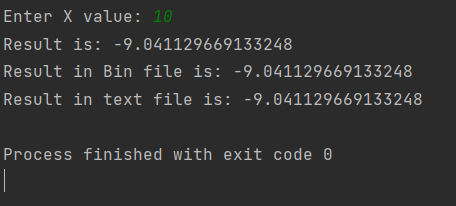
**Class Equations**

package KI\_35.Kudenchuk.Lab6;  
  
*/\*\*  
 \* Class <code>Equations</code> implements method for (x-4)/sin(3x-1) expression  
 \* calculation  
 \** ***@author*** *Kudenchuk Vladyslav  
 \** ***@version*** *1.0  
 \*/*class Equations  
{  
 public static double calculate(double x) throws CalcException  
 {  
 double y, rad;  
 rad = 3 \* x \* Math.*PI* / 180.0f - 1.0f;  
 try  
 {  
 y = (x - 4.0f) / Math.*sin*(3 \* x - 1.0f);  
 if (Double.*isNaN*(y) || y==Double.*NEGATIVE\_INFINITY* ||  
 y==Double.*POSITIVE\_INFINITY* || rad==0)  
 throw new ArithmeticException();  
 }  
 catch (ArithmeticException ex)  
 {  
 if (rad==0)  
 throw new CalcException("Exception reason: division by zero (sin(3 \* x - 1) == 0)");  
 else  
 throw new CalcException("Unknown reason of the exception during exception calculation");  
 }  
 return y;  
 }  
}

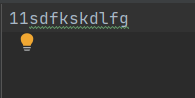
**Class FioApp**

package KI\_35.Kudenchuk.Lab6;  
import java.io.\*;  
import java.util.\*;  
  
public class FioApp {  
 */\*\** ***@param*** *args Cline arguments  
 \*/* public static void main(String[] args) throws IOException  
 {  
 CalcWFio obj = new CalcWFio();  
 Scanner s = new Scanner(System.*in*);  
 System.*out*.print("Enter X value: ");  
 double x = s.nextDouble();  
 obj.calculate(x);  
 System.*out*.println("Result is: " + obj.getResult());  
  
 CalcWFio.*writeResTxt*("file.txt", "11sdfkskdlfg");  
 obj.writeResBin("BinRes.bin");  
 obj.readResBin("BinRes.bin");  
 System.*out*.println("Result in Bin file is: " + obj.getResult());  
// obj.readResTxt("textRes.txt");  
 System.*out*.println("Result in text file is: " + obj.getResult());  
 }  
}

Console:



Text file:



**Відповіді на контрольні запитання**

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

Для створення файлових потоків і роботи з ними у Java є 2 класи, що успадковані від InputStream і OutputStream це - FileInputStream і FileOutputStream. Як і їх суперкласи вони мають методи лише для байтового небуферизованого блокуючого читання/запису даних та керуванням потоками. Компонуючи ці класи між собою і досягається необхідна кінцева функціональність потоку. Так одні класи, як FileInputStream, забезпечують елементарний доступ до файлів, інші, як PrintWriter, надають додаткової функціональності по високорівневій обробці даних, що пишуться у файл. Ще інші, наприклад, BufferedInputStream забезпечують буферизацію.

1. **Охарактеризуйте клас PrintWriter.**

Для буферизованого запису у текстовий потік найкраще використовувати клас PrintWriter. Цей клас має методи для виводу рядків і чисел у текстовому форматі: print, println, printf, - принцип роботи яких співпадає з аналогічними методами Systen.out.

**9. Охарактеризуйте клас RandomAccessFile.**

Керування файлами з можливістю довільного доступу до них здійснюється за допомогою класу RandomAccessFile. Відкривання файлу в режимі запису і читання/запису здійснюється за допомогою конструктора, що приймає 2 параметри – посилання на файл (File file) або його адресу (String name) та режим відкривання файлу (String mode): RandomAccessFile(File file, String mode); RandomAccessFile(String name, String mode).

**Висновок:** оволодів навиками використання засобів мови Java для роботи з потоками і файлами.