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

Дзьобань М. Б.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2022

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

**Завдання (варіант № 3)**

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.

**Варіант 3:** y=sin(x)/cos(x)

1. Автоматично згенерувати документацію до розробленого пакету.
2. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
3. Дати відповідь на контрольні запитання

**Текст програми**

**Main.java**

import KI\_35.Dzoban.Lab6.\*;  
  
  
import java.util.Scanner;  
import java.io.\*;  
import static java.lang.System.*out*;  
  
*/\*\*  
 \* Class Main implements driver for Calculation class  
 \** ***@author*** *Mykola Dzoban  
 \** ***@version*** *1.0.0  
 \*/*public class Main {  
 */\*\*  
 \** ***@param*** *args Main class argument  
 \*/* public static void main(String[] args) throws FileNotFoundException, IOException {  
 try  
 {  
 Calculation obj = new Calculation();  
 Scanner s = new Scanner(System.*in*);  
 *out*.print("Enter file name: ");  
 String fName = s.nextLine();  
 PrintWriter fout = new PrintWriter(new File(fName));  
 try  
 {  
 try  
 {  
 System.*out*.print("Enter data: ");  
 double data = s.nextDouble();  
 obj.calculate(data);  
 System.*out*.println("Result is: " + obj.getResult());  
 fout.print("Result is: " + obj.getResult());  
 RandomAccessFile raf = new RandomAccessFile("randomFile.txt", "rw");  
 raf.write("RandomAccessFile".getBytes());  
  
 obj.writeResTxt("textRes.txt");  
 obj.writeResBin("BinRes.bin");  
  
 obj.readResBin("BinRes.bin");  
 System.*out*.println("Result from BinRes file is: " + obj.getResult());  
 obj.readResTxt("textRes.txt");  
 System.*out*.println("Result from textRes file is: " + obj.getResult());  
  
 }  
 finally  
 {  
 fout.flush();  
 fout.close();  
 }  
 }  
 catch (CalcException ex)  
 {  
 *out*.print(ex.getMessage());  
// FileWriter fstream =new FileWriter(fName);  
// BufferedWriter out=new BufferedWriter(fstream);  
// out.write(ex.toString());  
// out.close();  
 }  
 }  
 catch (FileNotFoundException ex)  
 {  
 *out*.print("Exception reason: Perhaps wrong file path");  
 }  
 }  
}

**Результат виконання програми**

Enter file name: test.txt

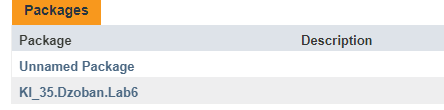
Enter data: 77

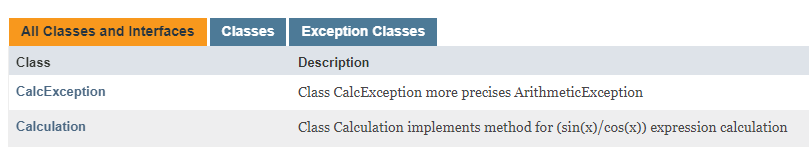
Result is: 4.331475874284157

Result from BinRes file is: 4.331475874284157

Result from textRes file is: 4.331476

**Фрагмент згенерованої документації**





**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**

Під час виконання лабораторної, я оволоділа навиками використання засобів мови Java для роботи з потоками і файлами.

**Додаток**

**Calculation.java**

package KI\_35.Dzoban.Lab6;  
  
import java.io.\*;  
import java.util.\*;  
  
*/\*\*  
 \* Class Calculation implements method for (sin(x)/cos(x)) expression calculation  
 \** ***@author*** *Mykola Dzoban  
 \** ***@version*** *1.0.0  
 \*/*public class Calculation {  
 private double y;  
 */\*\*  
 \* Method calculate the (sin(x)/cos(x)) expression  
 \** ***@param*** *x Angel in degrees  
 \** ***@throws*** *CalcException  
 \*/* public void calculate(double x) throws CalcException {  
 double rad;  
 rad = (x \* Math.*PI*) / 180.0;  
  
 try {  
 y = (Math.*sin*(rad) / Math.*cos*(rad));  
  
  
  
  
 if (y == Double.*NaN* || y == Double.*NEGATIVE\_INFINITY* || y == Double.*POSITIVE\_INFINITY* || 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  
 throw new CalcException("Unknown reason of the exception during exception calculation");  
 }  
 }  
  
 */\*\*  
 \** ***@return*** *Result of the (sin(x)/cos(x)) expression  
 \*/* public double getResult(){  
 return y;  
 }  
  
 public void writeResTxt(String fName) throws FileNotFoundException  
 {  
 PrintWriter f = new PrintWriter(fName);  
 f.printf("%f ",y);  
 f.close();  
 }  
  
 public void readResTxt(String fName)  
 {  
 try  
 {  
 File f = new File (fName);  
 if (f.exists())  
 {  
 Scanner s = new Scanner(f);  
 y = 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(y);  
 f.close();  
 }  
  
 public void readResBin(String fName) throws FileNotFoundException, IOException  
 {  
 DataInputStream f = new DataInputStream(new FileInputStream(fName));  
 y = f.readDouble();  
 f.close();  
 }  
  
}

**CalcExeption.java**

package KI\_35.Dzoban.Lab6;  
  
*/\*\*  
 \* Class CalcException more precises ArithmeticException  
 \** ***@author*** *Mykola Dzoban  
 \** ***@version*** *1.0.0  
 \*/*public class CalcException extends ArithmeticException {  
 public CalcException(){}  
 public CalcException(String message)  
 {  
 super(message);  
 }  
}