Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



**Звіт**

з лабораторної роботи № 5

**з дисципліни:** «Кросплатформенні засоби програмування»

**на тему:** «Файли»

**Виконав:**

студент групи *КІ-306*

*Яцків Адріян*

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

**Львів – 2023**

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

**Завдання: Варіант 1 y=tg(x)**

1. Створити клас, що реалізує методи читання/запису у текстовому і двійковому

форматах результатів роботи класу, що розроблений у лабораторній роботі No5.

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.

**Вихідний код програми**

**EquationApp**

package KI306.Yatskiv.Lab5;  
  
import java.io.\*;  
import java.util.Scanner;  
  
*/\*\*  
 \* Дана програма є класом драйвером який тестує роботу класу Calco  
 \** ***@author*** *Petro Bokalo KI-306  
 \** ***@version*** *1.0  
 \** ***@since*** *version 1.0  
 \*/*public class EquationApp {  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.print("Input your X in degrees: ");  
 if (input.hasNextDouble()) {  
 double inputAngle = input.nextDouble();  
 Calc c = new Calc(inputAngle);  
 System.*out*.println("Normalized angle: " + c.normalizeAngle() + " degrees");  
 try {  
 System.*out*.print("Tg of x = " + c.result() + "\n");  
 c.in\_fileDecimal(); // Запис у десятковому форматі  
 c.in\_fileBinary(); // Запис у двійковому форматі  
 } catch (CalcException ex) {  
 System.*out*.print(ex.getMessage());  
 }  
 c.close\_files();  
 input.close();  
  
 System.*out*.println("Data from decimal file:");  
 c.read\_fileDecimal();  
  
 System.*out*.println("\nData from binary file:");  
 c.read\_fileBinary();  
  
 c.close\_files();  
 } else {  
 System.*out*.print("Invalid data inputted \n");  
 }  
 }  
}

**Calc.java**

package KI306.Yatskiv.Lab5;  
  
import java.io.\*;  
import java.nio.ByteBuffer;  
  
*/\*\*  
 \* Тут реалізований клас Calc який рахує tg введеного числа  
 \** ***@author*** *Adriyan Yatskiv KI-306  
 \** ***@version*** *1.0  
 \** ***@since*** *version 1.0  
 \*/*class CalcException extends ArithmeticException {  
 public CalcException() {  
 }  
  
 public CalcException(String cause) {  
 super(cause);  
 }  
}  
  
public class Calc {  
 private double x;  
 private double tg = 0;  
 private PrintWriter foutDecimal;  
 private FileOutputStream foutBinary;  
 private FileInputStream finDecimal;  
 private FileInputStream finBinary;  
 public Calc(double x) {  
 this.x = x;  
 }  
  
 private double calculateRad() {  
 return x \* Math.*PI* / 180.0;  
 }  
  
  
 public double result() throws CalcException {  
 try {  
 double rad = calculateRad();  
 tg = Math.*tan*(rad);  
  
 if (Double.*isNaN*(tg) || Double.*isInfinite*(tg) || x == 90 || x == -90) {  
 throw new ArithmeticException();  
 }  
 } catch (ArithmeticException ex) {  
 double rad = calculateRad();  
  
 if (rad == Math.*PI* / 2.0 || rad == -Math.*PI* / 2.0) {  
 throw new CalcException("Exception reason: Illegal value of X for tangent calculation\n");  
 } else {  
 throw new CalcException("Unknown reason of the exception during exception calculation");  
 }  
 }  
 return tg;  
 }  
  
 public double normalizeAngle() {  
 // Normalize the angle to the range [0, 360] degrees  
 x = x % 360;  
 if (x < 0) {  
 x += 360;  
 }  
 return x;  
 }  
  
 public void in\_fileDecimal() {  
 try {  
 foutDecimal = new PrintWriter(new BufferedWriter(new FileWriter("Lab5Decimal.txt")));  
 foutDecimal.printf("Tg of x in decimal format = %.10f%n", result());  
 } catch (IOException e) {  
 System.err.println("Can't use the file!!!\n");  
 }  
 }  
  
 public void in\_fileBinary() {  
 try {  
 String fileName = "Lab5Binary.bin";  
 foutBinary = new FileOutputStream(fileName);  
 foutBinary.write(DoubleToBinary(result()));  
 } catch (IOException e) {  
 System.err.println("Can't use the file!!!\n");  
 }  
 }  
  
 public void read\_fileDecimal() {  
 try {  
 finDecimal = new FileInputStream("Lab5Decimal.txt");  
 BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(finDecimal));  
  
 String line;  
 while ((line = bufferedReader.readLine()) != null) {  
 System.out.println(line);  
 }  
  
 bufferedReader.close();  
 finDecimal.close();  
 } catch (IOException e) {  
 System.err.println("Can't read the file!!!\n");  
 }  
 }  
  
 public void read\_fileBinary() {  
 try {  
 finBinary = new FileInputStream("Lab5Binary.bin");  
 byte[] buffer = new byte[8];  
 int bytesRead;  
 while ((bytesRead = finBinary.read(buffer)) != -1) {  
 long bits = ByteBuffer.wrap(buffer).getLong();  
 double result = Double.longBitsToDouble(bits);  
 System.out.printf("Tg of x in binary format = %.10f%n", result);  
 }  
 } catch (IOException e) {  
 System.err.println("Can't read the file!!!\n");  
 }  
 }  
  
 public void close\_files() {  
 if (foutDecimal != null) {  
 foutDecimal.close();  
 }  
 if (foutBinary != null) {  
 try {  
 foutBinary.close();  
 } catch (IOException e) {  
 System.err.println("Error while closing binary file");  
 }  
 }  
 if (finDecimal != null) {  
 try {  
 finDecimal.close();  
 } catch (IOException e) {  
 System.err.println("Error while closing decimal file");  
 }  
 }  
 if (finBinary != null) {  
 try {  
 finBinary.close();  
 } catch (IOException e) {  
 System.err.println("Error while closing binary file");  
 }  
 }  
 }  
  
 // Метод для перетворення результату в двійковий формат  
 private byte[] DoubleToBinary(double number) {  
 long bits = Double.doubleToLongBits(number);  
 ByteBuffer buffer = ByteBuffer.allocate(Long.BYTES);  
 buffer.putLong(bits);  
 return buffer.array();  
 }  
}

**Результат виконання програми**

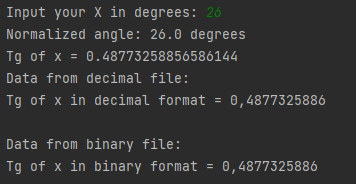
****

Рис.1 – Скріншот виконання програми

**Відповіді на контрольні запитання**

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

Принципи роботи з файловою системою в Java базуються на використанні класів, таких як File, FileInputStream, FileOutputStream, BufferedReader, BufferedWriter, Scanner, PrintWriter, RandomAccessFile, і так далі. Ці класи дозволяють читати та записувати дані в файли, виконувати операції з каталогами, перевіряти доступність файлів, видаляти та переміщати файли, створювати нові файли тощо.

1. Охарактеризуйте клас Scanner.

Клас Scanner є інструментом для зчитування різноманітних типів даних з різних джерел, таких як консоль, файли та рядки.

1. Наведіть приклад використання класу Scanner.

Scanner scanner = new Scanner(System.in);

System.out.print("Введіть ціле число: ");

int number = scanner.nextInt();

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

Запис у текстовий потік можна здійснити за допомогою класу PrintWriter.

1. Охарактеризуйте клас PrintWriter.

Клас PrintWriter представляє потік для запису символьних даних у текстовий файл.

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

Для читання і запису двійкових даних можна використовувати класи FileInputStream і FileOutputStream для потокового читання і запису байтів, або класи DataInputStream і DataOutputStream для роботи з примітивними типами даних.

1. Призначення класів DataInputStream і DataOutputStream.

Класи DataInputStream і DataOutputStream використовуються для читання та запису примітивних типів даних у двійковому форматі.

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

Для здійснення довільного доступу до файлів можна використовувати клас RandomAccessFile.

1. Охарактеризуйте клас RandomAccessFile.

Клас RandomAccessFile надає можливість зчитувати та записувати дані у файлі з довільним доступом, тобто переміщатися у файлі та читати/писати дані з будь-якої позиції.

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

Інтерфейс DataOutput визначає методи для запису примітивних даних у двійковому форматі. Клас DataOutputStream реалізує цей інтерфейс і дозволяє записувати дані у двійковому форматі до потоку.

**Висновок**

Ознайомився з використанням потоків та написав клас що отримує та записує дані з файлу та записує у форматі двійковому та текстовому. Розробив програму драйвер яка використовує даний клас.