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»

**Виконала:**

студентка групи *КІ-306*

*Максимчук К.С.*

**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 9** y=tg(x)/3x**):**

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**

import java.io.FileNotFoundException;  
import java.io.IOException;  
  
// Press Shift twice to open the Search Everywhere dialog and type `show whitespaces`,  
// then press Enter. You can now see whitespace characters in your code.  
public class Main {  
 public static void main(String[] args) throws FileNotFoundException, IOException {  
 WriteReadCalcRes fWriteReader = new WriteReadCalcRes();  
 int angle = 50;  
 Equation equation = new Equation();  
 fWriteReader.setNumber(9);  
 System.*out*.println(fWriteReader.getNumber());  
 try {  
 double result = equation.Calculate(angle);  
 System.*out*.println("Result: " + result);  
 fWriteReader.setNumber(result);  
 fWriteReader.WriteToFileTxt("resTxt.txt");  
 fWriteReader.WriteToFileBin("resBin.bin");  
 } catch (CalcException e) {  
 System.*err*.println("Calculation Error: " + e.getMessage());  
 }  
  
 System.*out*.println("Write result and read from txt file");  
 fWriteReader.ReadFromFileTxt("resTxt.txt");  
 System.*out*.println("Result of reading txt file is : " + fWriteReader.getNumber());  
  
 fWriteReader.ReadFromFileBin("resBin.bin");  
 System.*out*.println("Result of reading bin file is : " + fWriteReader.getNumber());  
  
 fWriteReader.WriteRandomNumber("Number1.txt");  
 }  
 }

public class Equation {  
 public double Calculate(double x) throws CalcException {  
 //y=tg(x)/3x  
 double y, rad;  
 rad = x \* Math.*PI* / 180.0;  
  
 try {  
 y = Math.*tan*(x)/(x \* 3);  
 //Math.tan(rad)  
 // If the result is not a valid number or x is at a critical point, throw ArithmeticException  
 if (Double.*isNaN*(y) || Double.*isInfinite*(y) || x == 90 || x == -90 || x == 0) {  
 throw new ArithmeticException();  
 }  
 } catch (ArithmeticException ex) {  
 // Handle the ArithmeticException and throw a higher-level CalcException with an explanation  
 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;  
 }  
}

public class CalcException extends ArithmeticException{  
 public CalcException(){}  
 public CalcException(String cause)  
 {  
 super(cause);  
 }  
}

import java.io.\*;  
import java.util.Scanner;  
  
public class WriteReadCalcRes {  
 private double number;  
  
 public double getNumber() {  
 return number;  
 }  
  
 public void setNumber (double number) {  
 this.number = number;  
 }  
  
 public void WriteToFileTxt(String fName) throws FileNotFoundException {  
 PrintWriter f = new PrintWriter(fName);  
 f.printf("%f ",number);  
 f.close();  
 }  
 public void ReadFromFileTxt(String fName)  
 {  
 try  
 {  
 File f = new File (fName);  
 if (f.exists())  
 {  
 Scanner s = new Scanner(f);  
 number = s.nextDouble();  
 s.close();  
 }  
 else  
 throw new FileNotFoundException("File " + fName + " not found");  
 }  
 catch (FileNotFoundException ex)  
 {  
 System.*out*.println(ex.getMessage());  
 }  
 }  
 public void WriteToFileBin(String fName) throws IOException {  
 DataOutputStream f = new DataOutputStream(new FileOutputStream(fName));  
 f.writeDouble(number);  
 f.close();  
 }  
 public void ReadFromFileBin(String fName) throws IOException {  
 DataInputStream f = new DataInputStream(new FileInputStream(fName));  
 number = f.readDouble();  
 f.close();  
 }  
 public void WriteRandomNumber(String fName) throws IOException {  
 RandomAccessFile num = new RandomAccessFile(fName, "rw");  
 num.writeDouble(number);  
 num.close();  
 }  
// public void ReadRandomNumber(String fName) throws IOException {  
// RandomAccessFile num = new RandomAccessFile(fName, "r");  
// number = num.readDouble();  
// num.close();  
// }  
}

**Результат виконання програми**

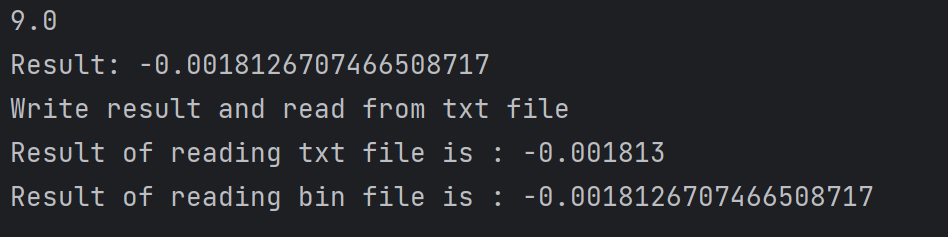
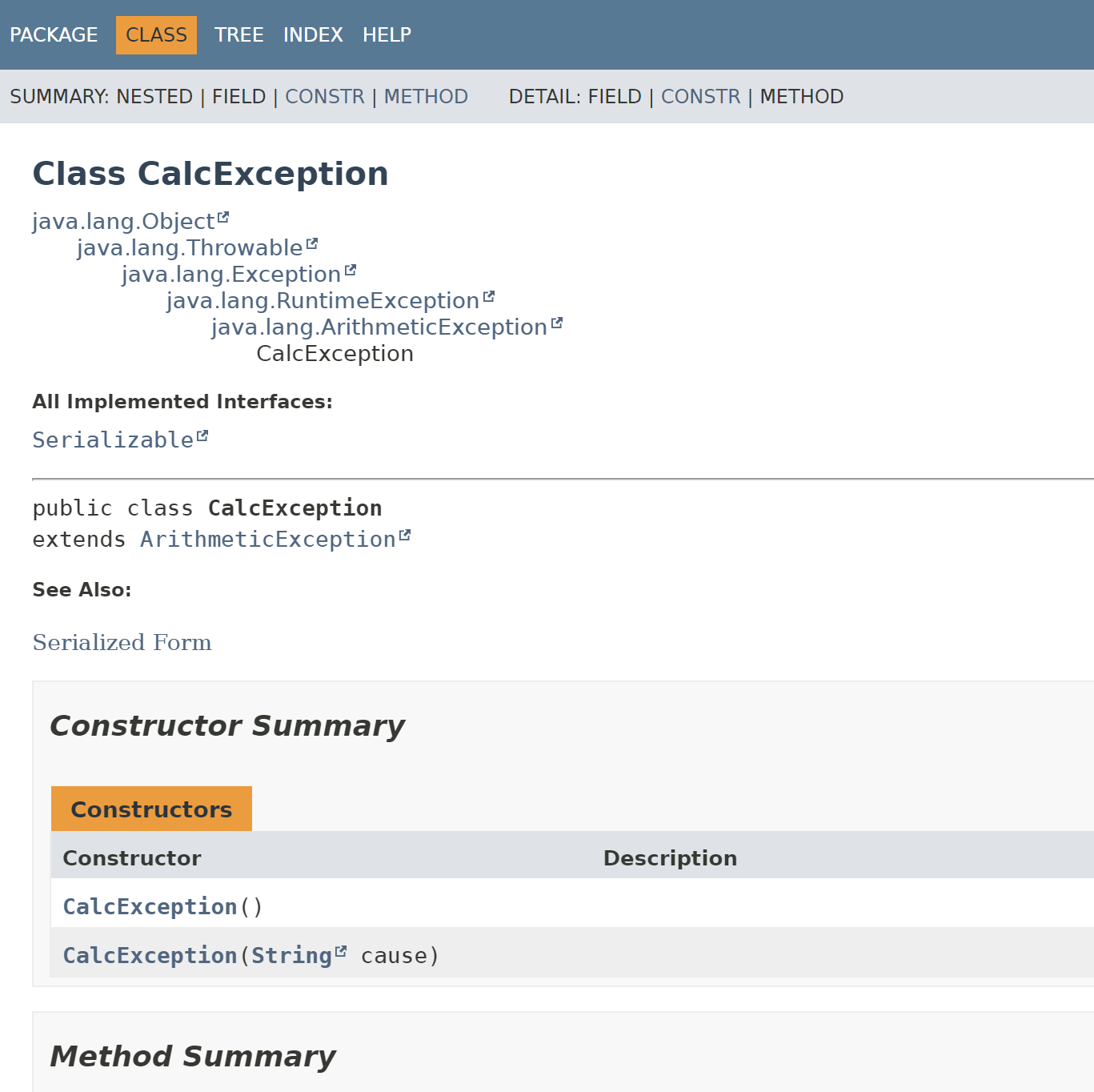
****

Рис.1 – Скріншот виконання програми

**Фрагмент згенерованої документації**



**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**

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