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-304

Мох. М. П.

**Прийняв:**

Олексів М.В.

**Львів – 2023**

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

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.

**Варіант 12:** y = sin(x) / tg(4x)

Main.java

package KI304.Moh.Lab5;  
  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
*/\*\*  
 \* The Main class serves as the entry point for the program, showcasing the interaction  
 \* between mathematical calculations, file operations, and user input/output.  
 \*  
 \* @author Marian Moh  
 \* @version 1.0  
 \* @since 2023-11-27  
 \*/*public class Main {  
  
 */\*\*  
 \* The main method serves as the entry point for the program. It orchestrates user input,  
 \* mathematical calculations, file operations, and exception handling.  
 \*  
 \* @param args The command-line arguments (unused).  
 \*/* public static void main(String[] args) {  
 System.*out*.println("Formula is: y = sin(x) / tg(4x)");  
 Scanner scanner = new Scanner(System.*in*);  
 double result;  
  
 while (true) {  
 System.*out*.print("\nEnter x: ");  
 try {  
 double x = scanner.nextDouble();  
 result = Calculator.*calculate*(x);  
  
 // Test methods of working with a regular file  
 FileTool.*saveResultToFile*(result);  
 String text = FileTool.*getResultFromFile*();  
 System.*out*.println("\n-- Text from file --\n" + text);  
  
 // Test methods of working with a binary file  
 FileTool.*saveResultToBinFile*(result);  
 result = FileTool.*getResultFromBinFile*();  
 System.*out*.println("\n-- Text from bin file --\n" + result);  
  
 break;  
 } catch (InputMismatchException e) {  
 System.*out*.println("Input miscast, please enter a double");  
 scanner.next();  
 } catch (ArithmeticException e) {  
 System.*out*.println("Divide by 0!" + e.getMessage());  
 } catch(FileNotFoundException e) {  
 System.*out*.println("File not found... ");  
 break;  
 } catch (IOException e) {  
 System.*out*.println("Input/Output exception... ");  
 break;  
 }  
 }  
 }  
}

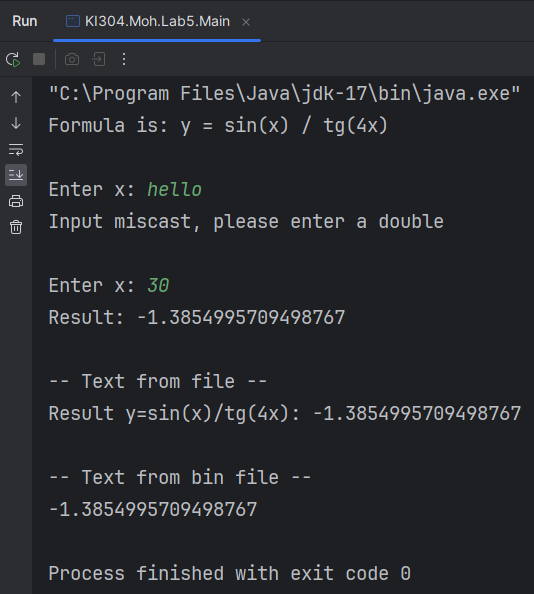
Calculator.java

package KI304.Moh.Lab5;  
  
*/\*\*  
 \* The Calculator class provides methods for mathematical calculations and result storage.  
 \* The results can be calculated using the formula y = sin(x) / tan(4x) and saved to a file.  
 \*  
 \* @author Marian Moh  
 \* @version 1.0  
 \* @since 2023-11-27  
 \*/*public class Calculator {  
 */\*\*  
 \* Holds the calculated result of the mathematical expression y = sin(x) / tan(4x).  
 \*/* private static double *result*;  
  
 */\*\*  
 \* Calculates the result of the mathematical expression y = sin(x) / tan(4x).  
 \*  
 \* @param x The input value for the calculation.  
 \* @return The result of the mathematical expression.  
 \* @throws ArithmeticException If the denominator (Math.tan(4 \* x)) is 0, indicating division by zero.  
 \*/* public static double calculate(double x) {  
 if(Math.*tan*(4 \* x) == 0.0) {  
 throw new ArithmeticException("Math.tan(4 \* " + x + ") is 0, and we can't divide by it!");  
 }  
 *result* = Math.*sin*(x) / Math.*tan*(4 \* x);  
 System.*out*.println("Result: " + *result*);  
 return *result*;  
 }  
}

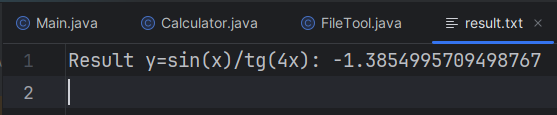
FileTool.java

package KI304.Moh.Lab5;  
  
import java.io.\*;  
  
*/\*\*  
 \* The FileTool class provides utility methods for saving and retrieving calculation results to and from files.  
 \*  
 \* @author Marian Moh  
 \* @version 1.0  
 \* @since 2023-11-27  
 \*/*public class FileTool {  
 */\*\*  
 \* The name of the file used to save the calculation results.  
 \*/* private static final String *FILE\_NAME* = "result.txt";  
  
 */\*\*  
 \* The name of the binary file used to save the calculation results.  
 \*/* private static final String *BIN\_FILE\_NAME* = "binResult.bin";  
  
 */\*\*  
 \* Saves the result of the calculation to a file.  
 \*  
 \* @throws FileNotFoundException If the specified file for result storage is not found.  
 \*/* public static void saveResultToFile(double result) throws FileNotFoundException {  
 try (PrintWriter printWriter = new PrintWriter(*FILE\_NAME*)) {  
 printWriter.println("Result y=sin(x)/tg(4x): " + result);  
 }  
 }  
  
 */\*\*  
 \* Retrieves the result of the calculation from the text file.  
 \*  
 \* @return The result as a string read from the text file.  
 \* @throws IOException If an I/O error occurs while reading from the file.  
 \*/* public static String getResultFromFile() throws IOException {  
 String line;  
 String text = "";  
 try(FileReader fileReader = new FileReader(*FILE\_NAME*);  
 BufferedReader bufferedReader = new BufferedReader(fileReader)) {  
 while ((line = bufferedReader.readLine()) != null) {  
 text += line;  
 }  
 }  
 return text;  
 }  
  
 */\*\*  
 \* Saves the result of the calculation to a binary file.  
 \*  
 \* @param result The result to be saved.  
 \* @throws IOException If an I/O error occurs while writing to the binary file.  
 \*/* public static void saveResultToBinFile(double result) throws IOException {  
 DataOutputStream out = new DataOutputStream(new FileOutputStream(*BIN\_FILE\_NAME*));  
 out.writeDouble(result);  
 }  
  
 */\*\*  
 \* Retrieves the result of the calculation from the binary file.  
 \*  
 \* @return The result read from the binary file.  
 \* @throws IOException If an I/O error occurs while reading from the binary file.  
 \*/* public static double getResultFromBinFile() throws IOException {  
 DataInputStream in = new DataInputStream(new FileInputStream(*BIN\_FILE\_NAME*));  
 return in.readDouble();  
 }  
}

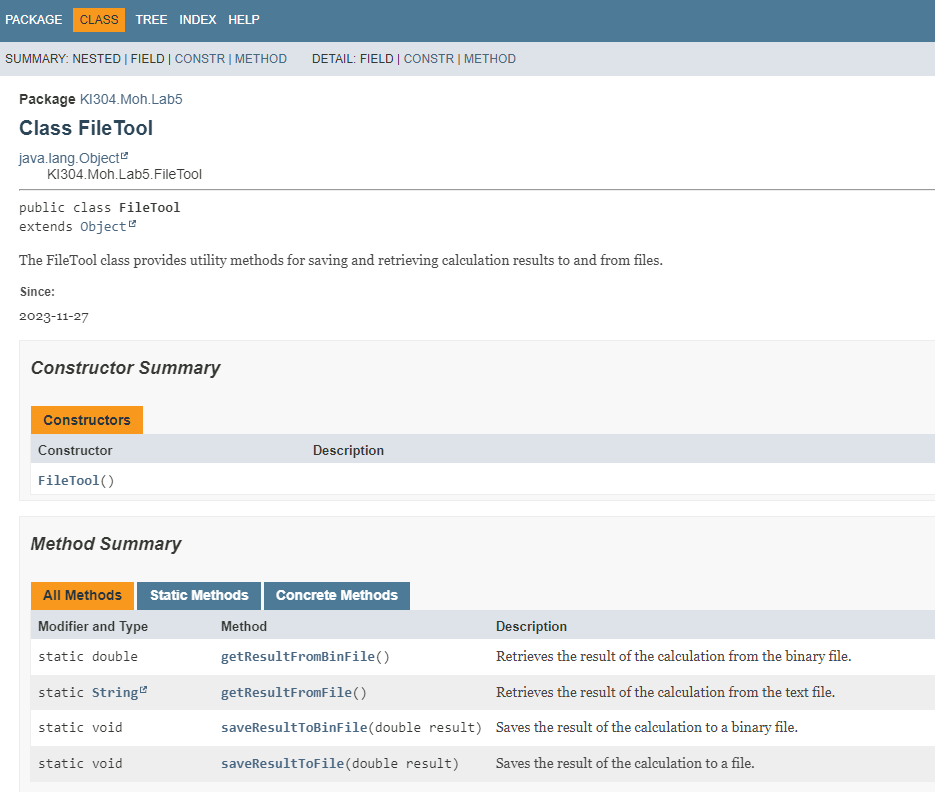
**Результат виконання програми:**



result.txt



**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**

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.