Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 3

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «ФАЙЛИ У JAVA»

**Виконав:**

студент групи *КІ-307*

*Сімчук Святослав*

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

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

**Завдання:**

1. Створити клас, що реалізує методи читання/запису у текстовому і двійковому форматах результатів роботи класу, що розроблений у лабораторній роботі №4. Написати програму для тестування коректності роботи розробленого класу.

2. Для розробленої програми згенерувати документацію.

3. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.

5. Дати відповідь на контрольні запитання.

Варіант 18: y=tg(x)/(sin(4x) – 2cos(x))

Файл ResultFileHandler.java

/\*\*

\* Package for LAB\_05

\*/

package ki307.simchuk.lab5;

import java.io.\*;

import java.nio.\*;

/\*\*

\* Class which implement file with result

\*/

public class ResultFileHandler {

/\*\*

\* Method for write result in text file

\* @param filePath name of the file

\* @param result value that will be written

\*/

public static void writeTextResult(String filePath, double result) {

try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath))) {

writer.write("Result: " + result);

} catch (IOException e) {

System.err.println("Error writing to file: " + e.getMessage());

}

}

/\*\*

\* Method for reading result for text file

\* @param filePath name of the file

\* @return zero if throws error

\* @throws FileNotFoundException if file not find

\* @throws IOException input output error

\*/

public static double readTextResult(String filePath) throws FileNotFoundException, IOException {

try(BufferedReader reader = new BufferedReader(new FileReader(filePath))) {

String line = reader.readLine();

if (line != null) {

String resultString = line.replace("Result: ", "");

return Double.parseDouble(resultString);

}

} catch (IOException e) {

System.err.println("Error reading from file: " + e.getMessage());

}

return 0.0;

}

/\*\*

\* Method for write result in binary file

\* @param filePath name of the file

\* @param result zero if throws error

\*/

public static void writeBinaryResult(String filePath, double result) {

try (DataOutputStream dos = new DataOutputStream(new FileOutputStream(filePath))) {

dos.writeDouble(result);

} catch (IOException e) {

System.err.println("Error writing to binary file: " + e.getMessage());

}

}

/\*\*

\* Method for reading result for binary file

\* @param filePath name of the file

\* @return zero if throws error

\*/

public static double readBinaryResult(String filePath) {

try (DataInputStream dis = new DataInputStream(new FileInputStream(filePath))) {

return dis.readDouble();

} catch (IOException e) {

System.err.println("Error reading from binary file: " + e.getMessage());

}

return 0.0;

}

public static String readFromFile(String filePath) throws IOException {

StringBuilder content = new StringBuilder();

try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {

String line;

while ((line = reader.readLine()) != null) {

content.append(line).append("\n");

}

} catch (IOException e) {

System.err.println("Error reading from file: " + e.getMessage());

}

return content.toString();

}

public static void writeToFile(String filePath, String data) throws IOException {

try (RandomAccessFile raf = new RandomAccessFile(filePath, "rw")) {

long fileLength = raf.length();

raf.seek(fileLength);

raf.write(data.getBytes());

} catch (IOException e) {

System.err.println("Error writing to file: " + e.getMessage());

}

}

}

Файл ResultFileHandler.java

/\*\*

\* Package for LAB\_05

\*/

package ki307.simchuk.lab5;

import java.io.FileNotFoundException;

import java.io.IOException;

/\*\*

\* Class Main which realize ResultFileHandler

\*/

public class Main {

/\*\*

\* Main method

\* @param args arguments

\* @throws FileNotFoundException if file is not found

\* @throws IOException input output error

\*/

public static void main(String[] args) throws FileNotFoundException, IOException {

double result = 42.0; // Personalized Value

// Test for text file

ResultFileHandler.writeTextResult("text\_result.txt", result);

double textResult = ResultFileHandler.readTextResult("text\_result.txt");

System.out.println("Text Result: " + textResult);

// Test of binary format

ResultFileHandler.writeBinaryResult("binary\_result.bin", result);

double binaryResult = ResultFileHandler.readBinaryResult("binary\_result.bin");

System.out.println("Binary Result: " + binaryResult);

String sourceFilePath = "/Users/simch/eclipse-workspace/Lab01/MyFile.txt";

String destinationFilePath = "Res.txt";

try {

String data = ResultFileHandler.readFromFile(sourceFilePath);

ResultFileHandler.writeToFile(destinationFilePath, data);

System.out.println("Data successfully read from and written to files.");

} catch (IOException e) {

System.err.println("An error occurred: " + e.getMessage());

}

}

}

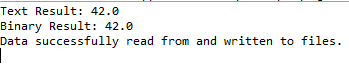
Файл text\_result.txt

Result: 42.0

Файл binary\_result.txt

@E

Вивід консолі:



Висновок:

Під час виконання лабораторної роботи №5 я оволодів навиками використання засобів мови Java для роботи з потоками і файлами.