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 у Java»

**Виконав:**

студент групи КІ-307

Трофименко Є.Е.

**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 № 22)**

1. Створити клас, що реалізує методи читання/запису у текстовому і двійковому форматах результатів роботи класу, що розроблений у лабораторній роботі No4. 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.



**Вихідний код програми**

*EquationsApp.java*

package ki307.Trofymenko.Lab5;

import java.io.IOException;

/\*\*

\* Клас реалізації методу обчислення виразу заданого варіантом

\* **@version** 1.0

\*/

public class EquationsApp {

/\*\*

\* Конструктор за замовченням

\*/

public EquationsApp() {}

/\*\*

\* Вхідний метод програми

\* **@param** args визначення методу main

\*/

public static void main(String[] args) {

CalculateTheEquation calc = new CalculateTheEquation();

try {

calc.readResultFromBin("src/bin.bin");

System.***out***.println("Result of calculation: " + calc.doCalculation());

calc.writeResultToBin("src/bin.bin");

} catch (IOException e) {

System.***out***.println(e);

}

try {

calc.readResultFromTxt("src/txt.txt");

System.***out***.println("Result of calculation: " + calc.doCalculation());

calc.writeResultToTxt("src/txt.txt");

} catch (IOException e) {

System.***out***.println(e);

}

}

}

*CalculateTheEquationInterface.java*

package ki307.Trofymenko.Lab5;

public interface CalculateTheEquationInterface {

double doCalculation();

}

*CalculateTheEquation.java*

package ki307.Trofymenko.Lab5;

import java.io.\*;

import java.util.Scanner;

public class CalculateTheEquation implements CalculateTheEquationInterface, ReadWriteData {

private double variable = 0;

*@Override*

public double doCalculation() {

try {

variable = Math.*tan*(4\*variable) / (variable);

return variable;

} catch (ArithmeticException e){

System.***out***.println("Arithmetic exception: illegal value.");

}

return 0;

}

*@Override*

public void writeResultToTxt(String fileName) throws IOException {

PrintWriter f = new PrintWriter(fileName);

f.printf("%f ", variable);

f.close();

}

*@Override*

public void writeResultToBin(String fileName) throws IOException {

DataOutputStream f = new DataOutputStream(new FileOutputStream(fileName));

f.writeDouble(variable);

f.close();

}

*@Override*

public void readResultFromTxt(String fileName) throws IOException {

File f = new File (fileName);

if (f.exists())

{

Scanner scanner = new Scanner(f);

variable = scanner.nextDouble();

scanner.close();

}

else

throw new IOException("File " + fileName + "not found");

}

*@Override*

public void readResultFromBin(String fileName) throws IOException {

DataInputStream f = new DataInputStream(new FileInputStream(fileName));

variable = f.readDouble();

f.close();

}

}

*ReadWriteData.java*

package ki307.Trofymenko.Lab5;

import java.io.IOException;

public interface ReadWriteData {

void writeResultToTxt(String fileName) throws IOException;

void writeResultToBin(String fileName) throws IOException;

void readResultFromTxt(String fileName) throws IOException;

void readResultFromBin(String fileName) throws IOException;

}

**Результат виконання програми**

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

**Фрагмент згенерованої документації**

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Изображение выглядит как текст, линия, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, число, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, чек

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Відповіді на контрольні запитання**

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

*Принцип здійснення читання/запису даних нічим не відрізняється від такого принципу у інших мовах програмування. Все починається з створення потоку на запис або читання після чого викликаються методи, що здійснюють обмін інформацією. Після завершення обміну даними потоки необхідно закрити щоб звільнити ресурси.*

1. Охарактеризуйте клас Scanner.

*На відміну від InputStreamReader і FileReader, що дозволяють лише читати текст, він має велику кількість методів, які здатні читати як рядки, так і окремі примітивні типи з подальшим їх перекодуванням до цих типів, робити шаблонний аналіз текстового потоку, здатний працювати без потоку даних та ще багато іншого.*

1. Наведіть приклад використання класу Scanner.

*Приклад читання даних за допомогою класу Scanner з стандартного потоку вводу:*

*Scanner sc = new Scanner(System.in);*

*int i = sc.nextInt();*

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

*Для буферизованого запису у текстовий потік найкраще використовувати клас PrintWriter.*

1. Охарактеризуйте 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.*

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

*Читання двійкових даних примітивних типів з потоків здійснюється за допомогою класів, що реалізують інтерфейс DataInput, наприклад класом DataInputStream.*

*Запис двійкових даних примітивних типів у потоки здійснюється за допомогою класів, що реалізують інтерфейс DataOutput, наприклад класом DataOutputStream.*

1. Призначення класів DataInputStream і DataOutputStream.

*Вони виступають так званими фільтрами, що своїм комбінуванням забезпечують додаткову лише необхідну функціональність при читанні даних з файлу. Аналогічний підхід застосовано і при реалізації класів для обробки текстових даних, що успадковані від Reader і Writer.*

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

*Керування файлами з можливістю довільного доступу до них здійснюється за допомогою класу RandomAccessFile.*

1. Охарактеризуйте клас 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).*

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

*Клас DataOutputStream реалізується для запису двійкових даних примітивних типів у потоки для інтерфейсу DataOutput.*

**Висновок**

На данній лабораторній роботі я оволодів використанням засобів мови Java для роботи з потоками і файлами. Створив клас, що реалізує методи читання/запису у текстовому і двійковому форматах результатів роботи класу, що розроблений у лабораторній роботі №4.