Міністерство освіти і науки України

Національний університет "Львівська політехніка"

Звіт з лабораторної роботи №5

з курсу “Кросплатформні засоби програмування”

Файли у Java

Виконав: студент гр. KI-306

Шаповал Віталій

Прийняв: к.т.н. Олексів М.В.

Львів 2024 р.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Зубчаті масиви

Бібліотека класів мови Java має більше 60 класів для роботи з потоками. Потаками у мові Java називаються об'єкти з якими можна здійснювати обмін даними. Цими об'єктами найчастіше є файли, проте ними можуть бути стандартні пристрої вводу/виводу, блоки пам'яті і мережеві підключення тощо. Класи по роботі з потоками об'єднані у кілька ієрархій, що призначені для роботи з різними видами даних, або забезпечувати додаткову корисну функціональність, наприклад, підтримку ZIP архівів.

Варіант № 28 **y=1/ctg(2x)**

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.

Код програми:

Файл ./ce306/shapoval/lab5/CalcException.java:

```
  package ce306.shapoval.lab5;

/**
* Class <code>CalcException</code> more precises ArithmeticException
* @author Shapoval
* @version 1.0
*/
public class CalcException extends ArithmeticException {
    public CalcException(){}

    public CalcException(String cause) {
        super(cause);
    }
}
```

Файл ./ce306/shapoval/lab5/CalcWFio.java:

```
package ce306.shapoval.lab5;

import java.io.*;
```

```
import java.util.Scanner;

/**
 * The <code>CalcWFio</code> class is responsible for performing calculations
 * and reading/writing the results to both text and binary files.
 */
public class CalcWFio {
    private double result;

    /**
     * Default constructor that initializes the result to 0.0.
     */
    public CalcWFio() {
        result = 0.0;
    }

    /**
     * Parameterized constructor that accepts a value of <code>x</code>
     * and performs a calculation.
     *
     * @param x The input value for calculation
     */
    public CalcWFio(double x) {
        calculate(x);
    }

    /**
     * This method performs a calculation using the provided value of
<code>x</code>.
     *
     * @param x The input value for calculation
     */
    public void calculate(double x) {
        result = (new Equation()).calculate(x);
    }

    /**
     * Returns the result of the calculation.
     *
     * @return The result as a <code>double</code>
     */
    public double getResult() {
        return result;
    }

    /**
     * Writes the result to a text file.
     *
     * @param fname The name of the text file
     * @throws FileNotFoundException If the file cannot be found or created
     */
    public void writeResTxt(String fname) throws FileNotFoundException {
        PrintWriter fout = new PrintWriter(fname);
```

```
        fout.printf("%f", result);
        fout.close();
    }

    /**
     * Reads the result from a text file.
     *
     * @param fname The name of the text file
     */
    public void readResTxt(String fname) {
        try {
            File fin = new File(fname);
            if (fin.exists()) {
                Scanner s = new Scanner(fin);
                result = s.nextDouble();
                s.close();
            } else {
                throw new FileNotFoundException("File " + fname + " not found");
            }
        } catch (FileNotFoundException ex) {
            System.out.println(ex.getMessage());
        }
    }

    /**
     * Writes the result to a binary file.
     *
     * @param fname The name of the binary file
     * @throws IOException If an I/O error occurs
     * @throws FileNotFoundException If the file cannot be found or created
     */
    public void writeResBin(String fname) throws IOException,
FileNotFoundException {
        DataOutputStream dout = new DataOutputStream(new
FileOutputStream(fname));
        dout.writeDouble(result);
        dout.close();
    }

    /**
     * Reads the result from a binary file.
     *
     * @param fname The name of the binary file
     * @throws IOException If an I/O error occurs
     */
    public void readResBin(String fname) throws IOException {
        DataInputStream din = new DataInputStream(new FileInputStream(fname));
        result = din.readDouble();
        din.close();
    }
}
```

Файл ./ce306/shapoval/lab5/Equation.java:

```
package ce306.shapoval.lab5;
/**
* Class <code>Equations</code> implements method for ( y = 1/ctg(2x)) expression
* calculation
* @author Shapoval
* @version 1.0
*/
public class Equation {

    /**
     * Calculate value of ( y = 1/ctg(2x)) expression for a given angle x in
     * degrees.
     * @param x angle in degrees
     * @return value of ( y = 1/ctg(2x)) for given angle x
     * @throws CalcException if given argument is equal or near to 45 degrees
     */
    public double calculate(double x) throws CalcException {
        x = x % 90;
        double y;
        double rad = x * Math.PI / 180;

        try {
            y = Math.tan(2 * rad);
            if (y == Double.NaN ||
                y == Double.POSITIVE_INFINITY ||
                y == Double.NEGATIVE_INFINITY ||
                x == -45.0 ||
                x == 45.0
                ) {
                throw new ArithmeticException();
            }
        } catch (ArithmeticException ex) {
            if (x == 45.0 || x == -45.0) {
                throw new CalcException("Exception reason: Illegal value of x");
            } else {
                throw new CalcException("Unknown reason");
            }
        }

        return y;
    }
}
```

Файл ./FioApp.java:

```
import ce306.shapoval.lab5.CalcWFio;

import java.util.Scanner;
import java.io.*;

/**
 * The <code>FioApp</code> class implements a console application
```

```
 * for calculating a result using the <code>CalcWFio</code> class
 * and saving/reading the result from text and binary files.
 */
public class FioApp {

    /**
     * The <code>main</code> method is the entry point of the program.
     * It prompts the user for the value of <code>x</code>, performs
calculations,
     * and writes/reads the results from files.
     *
     * @param args Command line arguments
     * @throws IOException If an I/O error occurs
     * @throws FileNotFoundException If the specified file is not found
     */
    public static void main(String[] args) throws IOException,
FileNotFoundException {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter value of x: ");
        double x = in.nextDouble();

        // Create an instance of CalcWFio and perform calculations
        CalcWFio calc = new CalcWFio(x);
        calc.calculate(x);
        System.out.println("Result: " + calc.getResult());

        // Write result to text file and read it back
        calc.writeResTxt("res.txt");
        calc.readResTxt("res.txt");
        System.out.println("Result: " + calc.getResult());

        // Write result to binary file and read it back
        calc.writeResBin("res.bin");
        calc.readResBin("res.bin");
        System.out.println("Result: " + calc.getResult());
    }
}
```

Виконання програми:

```
● › cd ../Lab5
● › java ./FioApp.java
  Enter value of x: 67
  Result: -1.0355303137905703
  Result: -1.03553
  Result: -1.03553
⊗ › java ./FioApp.java
  Enter value of x: 45
  Exception in thread "main" ce306.shapoval.lab5.CalcException: Exception reason: Illegal value of x
          at ce306.shapoval.lab5.Equation.calculate(Equation.java:34)
          at ce306.shapoval.lab5.CalcWFio.calculate(CalcWFio.java:18)
          at ce306.shapoval.lab5.CalcWFio.<init>(CalcWFio.java:14)
          at FioApp.main(FioApp.java:28)
```

рис 1. результат виконання завдання в терміналі

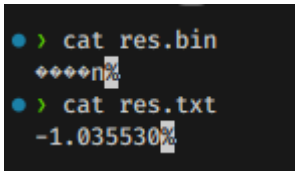

рис 2. результат виконання у текстовому файлі

Висновок

На цій лабораторній роботі я поглибив свої знання мови Java, зокрема у роботі з файлами у текстовому та двійковому форматах. Я створив клас для читання та запису даних, який базується на попередньо розробленому класі у лабораторній роботі №4, та перевірив його коректність за допомогою тестової програми. Я також ознайомився з процесом генерації документації за допомогою коментарів у коді, що значно полегшило автоматизацію цього процесу. Крім того, я завантажив проєкт на GitHub, що дозволяє зручно відстежувати зміни та співпрацювати над кодом у майбутньому. Ця лабораторна робота також дала мені змогу відпрацювати навички роботи з GitHub відповідно до методичних вказівок. У ході виконання завдання я закріпив навички структуризації програм, роботи з винятками та коректного оброблення введення даних.