

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра «Електронних обчислювальних машин»



Звіт
з лабораторної роботи № 6
з дисципліни: «Кросплатформенні засоби програмування»
на тему: «Файли»

Виконав:

студент групи КІ-35

Андрусяк М.В.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

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

Завдання (варіант № 27)

1. Створити клас, що реалізує методи читання/запису у текстовому і двійковому форматах результатів роботи класу, що розроблений у лабораторній роботі №5. 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 N-27

$$27. y = 1/\operatorname{tg}(2x)$$

Текст програми

CalcTgApp

```
package Lab5AndrusiakKI35;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Random;
import java.util.Scanner;

public class CalcTgApp {
    public static void main(String[] args) throws IOException {
        System.out.println("Your expression : 1/tg(2x)\n");
        System.out.print("Enter x : \n");
        Scanner in = new Scanner(System.in);

        if (!in.hasNextDouble()) {
            throw new CalcException("Invalid input!");
        }

        FileService fS = new FileService();

        fS.calcTg(in.nextDouble());

        fS.writeResTxt("Calc.txt");
        fS.writeResBin("Calcbin.bin");
    }
}
```

```

        fS.readResTxt("Calc.txt");
        System.out.println(fS.getResult());
        fS.readResBin("Calcbin.bin");
        System.out.println(fS.getResult());
    }
}

```

CalcTg

```

package Lab5AndrusiakKI35;

public class CalcTg {

    public double calc(double x) throws CalcException {
        double rad;
        double res;
        rad = x * Math.PI / 180;

        try {
            res = (1.0 / Math.tan(2 * rad));
            // Якщо результат не є числом, то генеруємо виключення
            if (res==Double.NaN || res==Double.NEGATIVE_INFINITY ||
res==Double.POSITIVE_INFINITY || x==45 || x== -45) // при PI/4 - вийде
нескінченність
                throw new ArithmeticException();
        }
        catch (ArithmeticException ex)
        {
            // створимо виключення вищого рівня з поясненням причини //
ВІНИКНЕННЯ ПОМИЛКИ
            if (rad==Math.PI/4.0 || rad==Math.PI/4.0)
                throw new CalcException("Exception reason: Illegal value of X
for tangent calculation");
            else
                throw new CalcException("Unknown reason of the exception during
exception calculation");
        }
        return res;
    }
}

```

CalcException

```

package Lab5AndrusiakKI35;

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

```

FileService

```
package Lab5AndrusiakKI35;

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

public class FileService {
    private double result;

    public void writeResTxt(String fName) throws FileNotFoundException {
        PrintWriter f = new PrintWriter(fName);
        f.printf("%f ", result);
        f.close();
    }

    public void readResTxt(String fName) {
        try
        {
            File f = new File(fName);
            if (f.exists())
            {
                Scanner s = new Scanner(f);
                result = s.nextDouble();
                s.close();
            }
            else
                throw new FileNotFoundException("File " + fName + "not found");
        }
        catch (FileNotFoundException ex)
        {
            System.out.print(ex.getMessage());
        }
    }

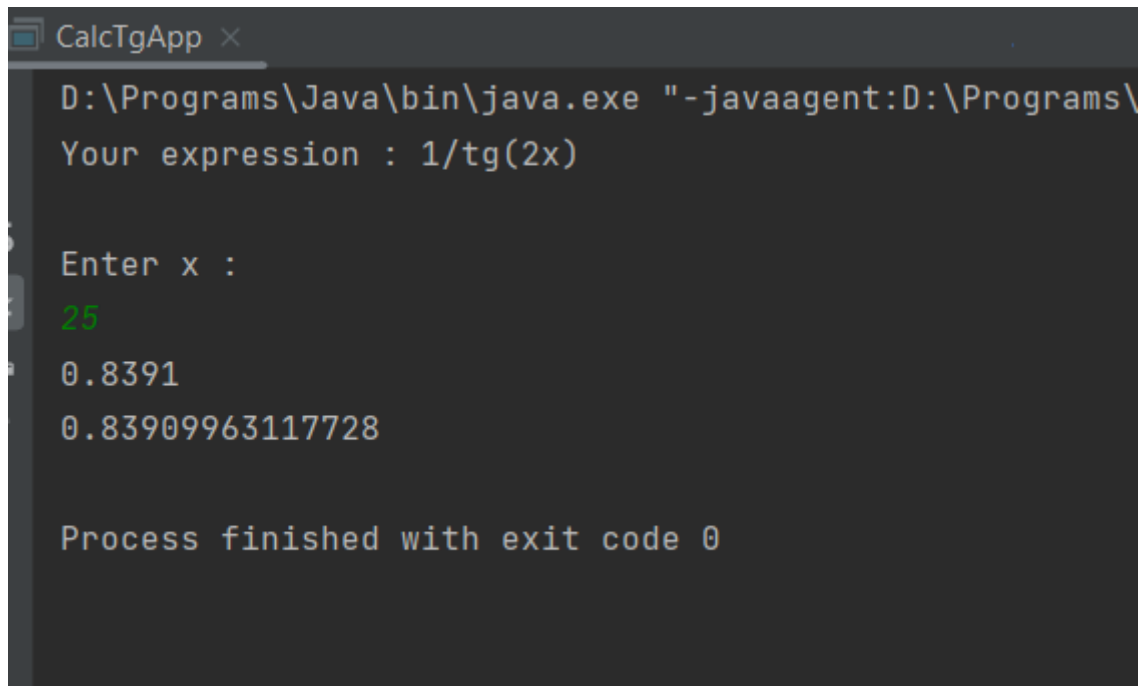
    public void writeResBin(String fName) throws IOException {
        DataOutputStream f = new DataOutputStream(new FileOutputStream(fName));
        f.writeDouble(result);
        f.close();
    }

    public void readResBin(String fName) throws IOException {
        DataInputStream f = new DataInputStream(new FileInputStream(fName));
        result = f.readDouble();
        f.close();
    }

    public void calcTg(Double x) {
        CalcTg obj = new CalcTg();
        result = obj.calc(x);
    }

    public double getResult() {
        return result;
    }
}
```

Результат виконання програми



```
CalcTgApp x
D:\Programs\Java\bin\java.exe "-javaagent:D:\Programs\
Your expression : 1/tg(2x)

Enter x :
25
0.8391
0.83909963117728

Process finished with exit code 0
```

Відповіді на контрольні запитання

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

Для буферизованого запису у текстовий потік найкраще використовувати клас **`PrintWriter`**. Цей клас має методи для виводу рядків і чисел у текстовому форматі: `print`, `println`, `printf`, - принцип роботи яких співпадає з аналогічними методами `System.out`.

Висновок : на даній лабораторній роботі я ознайомився зі засобами роботи з потоками і файлами мови Java .В результаті виконання роботи, я створив клас, що реалізує методи читання/запису у текстовому і двійковому форматах результатів роботи класу, що розроблений у лабораторній роботі №5.