

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: «ВИКЛЮЧЕННЯ »

Виконав:

студент групи КІ-35

Ничай В.Б.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

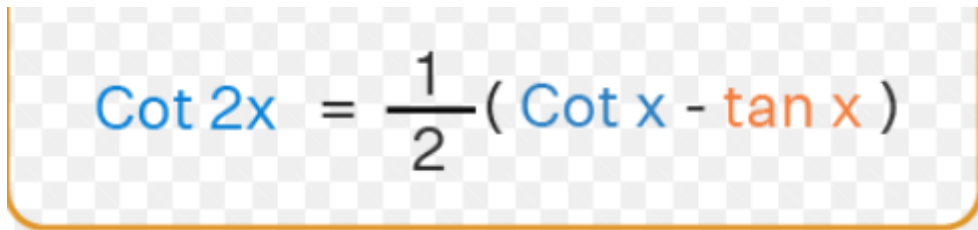
Львів – 2022

Мета: оволодіти навиками використання механізму виключень при написанні програм мовою Java.

Завдання

1. Створити клас, що реалізує метод обчислення виразу заданого варіантом. Написати на мові Java та налагодити програму-драйвер для розробленого класу. Результат обчислень записати у файл. При написанні програми застосувати механізм виключень для виправлення помилкових ситуацій, що можуть виникнути в процесі виконання програми. Програма має розміщуватися в пакеті Група.Прізвище.Lab5 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
2. Автоматично згенерувати документацію до розробленого пакету.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
4. Дати відповідь на контрольні запитання.

20. $y = \text{tg}(x) \text{ctg}(2x)$


$$\text{Cot } 2x = \frac{1}{2} (\text{Cot } x - \tan x)$$

Текст програми

EquationsApp.java

```
package Nychai.Lab5;

import java.util.Scanner;
import java.io.*;
import static java.lang.System.out;
/**
 * Class <code>EquationsApp</code> test class
 * @author Nychai Volodymyr KI-35
 * @version 1.0
 */
public class EquationsApp {
    /**
     * @param args
     */
    public static void main(String []args){
        try {
            out.print("Enter file name: ");
            Scanner in = new Scanner(System.in);
            String fName = in.nextLine();
            PrintWriter fout = new PrintWriter(new File(fName));

            try {
```

```

        Equation eq = new Equation();
        out.print("Enter X: ");
        fout.print(eq.Calc(in.nextInt()));
    }
    finally {
        fout.flush();
        fout.close();
    }
}
catch (CalcException ex) {
    out.print(ex.getMessage());
}
}
catch (FileNotFoundException ex) {
    out.println("Exception reason: Perhaps wrong file path");
}
}
}
}

```

Equation.java

```

package Nychai.Lab5;
/**
 * Class <code>Equation</code> check exception and calc
 * @throws CalcException
 * @author Nychai Volodymyr KI-35
 * @version 1.0
 */
public class Equation {
    public double Calc (int x) throws CalcException{
        double y, rad;
        rad = x * Math.PI / 180.0;

        try {
            double ctg = 1/Math.tan(rad);
            y = Math.tan(rad)* ((ctg-Math.tan(rad))/2);
            if (y==Double.NaN || y==Double.NEGATIVE_INFINITY ||
                y==Double.POSITIVE_INFINITY || x==90 || x== -90) {
                throw new ArithmeticException();
            }
        }
        catch (ArithmeticException ex){
            if (rad==Math.PI/2.0 || rad==Math.PI/2.0)
                throw new CalcException("Exception reason: Illegal value of X
for tangent calculation");
            else if (x==0)
                throw new CalcException("Exception reason: X = 0");
            else
                throw new CalcException("Unknown reason of the exception
during exception calculation");
        }
        return y;
    }
}

```

CalcException.java

```
package Nychai.Lab5;

public class CalcException extends ArithmeticException{
    public CalcException() {}

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

Результат роботи програми

```
0.4986267123353101|
```

Фрагмент згенерованої документації

All Classes and Interfaces

All Classes and Interfaces

Classes

Exceptions

Class	Description
CalcException	
Equation	Class Equation check exception and calc
EquationsApp	Class EquationsApp test class

Package Nychai.Lab5

Class EquationsApp

java.lang.Object[Ⓜ]

Nychai.Lab5.EquationsApp

```
public class EquationsApp
extends ObjectⓂ
```

Class EquationsApp test class

Constructor Summary

Constructors

Constructor	Description
EquationsApp()	

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
static void	main(String [Ⓜ] [] args)	

Methods inherited from class java.lang.Object[Ⓜ]

clone[Ⓜ], equals[Ⓜ], finalize[Ⓜ], getClass[Ⓜ], hashCode[Ⓜ], notify[Ⓜ], notifyAll[Ⓜ], toString[Ⓜ], wait[Ⓜ], wait[Ⓜ], wait[Ⓜ]

Контрольні питання

1. Дайте визначення терміну «виключення».

Виключення – це механізм мови Java, що забезпечує негайну передачу керування блоку коду опрацювання критичних помилок при їх виникненні уникаючи процесу розкручування стеку

2. У яких ситуаціях використання виключень є виправданим?

Генерація виключень застосовується при:

- помилках введення, наприклад, при введенні назви неіснуючого файлу або Інтернет адреси з подальшим зверненням до цих ресурсів, що призводить до генерації помилки системним програмним забезпеченням;

- збогах обладнання;
- помилках, що пов'язані з фізичними обмеженнями комп'ютерної системи, наприклад, при заповненні оперативної пам'яті або жорсткого диску;
- помилках програмування, наприклад, при некоректній роботі методу, читанні елементів порожнього стеку, виходу за межі масиву тощо.

Висновок: на даній лабораторній роботі я ознайомився з поняттям виключення. Самостійно написав виключення до програми та протестував його. Навчився обробляти помилки на виключеннях. Дізнався про структуру синтаксису Java try – catch. На практиці укріпив знання.