

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
Сабадаш Ю.А.
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.

8. $y = \sin(x) / \sin(2x - 4)$

Текст програми

Main.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 {
                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");
    }
}
}

```

Calculate.java

```

/**
 * @throws CalcException
 * @author Sabadash Yurii KI-35
 * @version 1.0
 */

public class Calculate {
    public double calc(double x){

        //8. y=sin(x)/sin(2x-4)
        double rad=x*Math.PI/180;
        double y;
        double rad2 = ((2 * x -4) * Math.PI/180) ;

        try{

            y=Math.sin(rad)/Math.sin(rad2);
            if(y== Double.NEGATIVE_INFINITY || y==Double.POSITIVE_INFINITY ||
y==Double.NaN || Math.abs((rad2)%Math.PI)==0)
                throw new ArithmeticException();
        }

        catch(ArithmeticException e){
            if(Math.abs(rad2%Math.PI)==0)
                throw new CalcException("Exception reason: divide by zero");
            else
                throw new CalcException("Exception reason: Unknown reason of
the exception during exception calculation");
        }

        return y;
    }
}

```

CalcException.java

```

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

```

```
        super(cause);
    }
}
```

Результат роботи програми

```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe"
Enter X: 3
y = 1.4996192262908286

Process finished with exit code 0
|
```

Фрагмент згенерованої документації

PACKAGE

CLASS

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHODDETAIL: FIELD | CONSTR | METHOD

Class Calculate

java.lang.Object[↗]

Calculate

public class Calculate

extends Object[↗]

Constructor Summary

Constructors

Constructor	Description
Calculate()	

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
double	calc(double x)	

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

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

Constructor Details

Calculate

public Calculate()

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;
- збоях 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.