## Міністерство освіти і науки України

# Національний університет "Львівська політехніка"

Кафедра ЕОМ



Звіт

## З лабораторної роботи №5

## **Варіант** – 21

**3 дисципліни:** «Кросплатформенні засоби програмування»

На тему: «Виключення»

Виконав: ст. гр. КІ-35

Сухан Д. В.

Перевірив: доцент кафедри ЕОМ

Іванов Ю.С.

**Meтa:** Оволодіти навиками використання механізму виключень при написанні програм мовою Java.

#### Завдання (варіант № 21)

# 21. $y=\sin(3x-5)/\cot(2x)$

- 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. Lab 5 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.

## Хід роботи:

## Текст програми

## CalcException.java

```
package KI35.Sukhan.Lab5;

/**

* Class <code>CalcException</code> more precises ArithmeticException

* @author Denys Sukhan

* @version 1.0

*/

//Клас CalcException peaлiaye користувацьке виключення

public class CalcException extends ArithmeticException {
    /**
    *Constructor
    */
```

```
public CalcException(){}

/**

*Constructor

* @param cause

*/

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

## Realization.java

```
package KI35.Sukhan.Lab5;
* Class <code>Realization</code> implements method for (
sin(3x-5) * tg(2x) ) expression
* calculation
* @author Denys Sukhan
 * @version 1.0
//Клас Realization реалізує обчислення виразу ( sin(3x-5) *
tg(2x))
public class Realization {
     * @param x
     * @throws CalcException
    public double calculate(double x) throws CalcException{
        double y, sin_rad ,tg_rad;
        sin_rad = Math.toRadians(3*x - 5);
        tg rad = Math.toRadians(2*x);
```

```
try {
            y = Math.sin(sin rad)*( Math.tan(tg rad));
                  // Якщо результат не \epsilon числом, то генеру\epsilonмо
виключення
             if (y==Double.NaN | | y==Double.NEGATIVE INFINITY
\parallel y==Double.POSITIVE INFINITY \parallel 2*x == 90 \parallel 2*x == -90 )
                 throw new ArithmeticException();
        }
        catch(ArithmeticException ex){
             // створимо виключення вищого рівня з поясненням
причини
             // виникнення помилки
             if (tg_rad == Math.PI/2.0 || tg_rad == -
Math.PI/2.0)
                 throw new CalcException("Exception reason:
Illegal value of X for tangent calculation");
             else
                 throw new CalcException("Unknown reason of the
exception during exception calculation");
        System.out.println(y);
        return y;
    }
```

# App.java

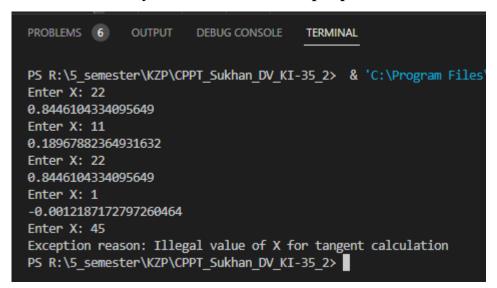
```
package KI35.Sukhan.Lab5;
import java.util.Scanner;
import java.io.*;
import static java.lang.System.out;

/**
* Клас Арр реалізує програму-драйвер
*
```

```
@author Sukhan Denys
 @version 1.0
 @since version 1.0
public class App {
     * @param args
    public static void main(String[] args) {
        try
            Scanner in = new Scanner(System.in);
            PrintWriter fout = new PrintWriter(new
File("KI35/Sukhan/Lab5/Result.txt"));
             try
            {
                for(int i = 0;i<6;i++){
                    Realization eq = new Realization();
                    out.print("Enter X: ");
                    fout.print("Result: "+
eq.calculate(in.nextInt())+"\n");
                }
             catch (CalcException ex)
              {
                out.print(ex.getMessage());
                fout.print(ex.getMessage());
             finally
                 fout.flush();
```

```
fout.close();
}
catch (FileNotFoundException ex)
{
    out.print("Exception reason: Perhaps wrong file path");
}
}
```

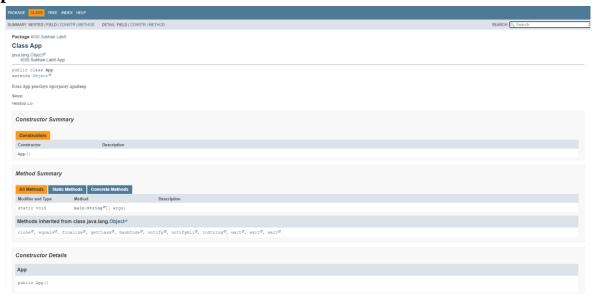
## Результат виконання програми



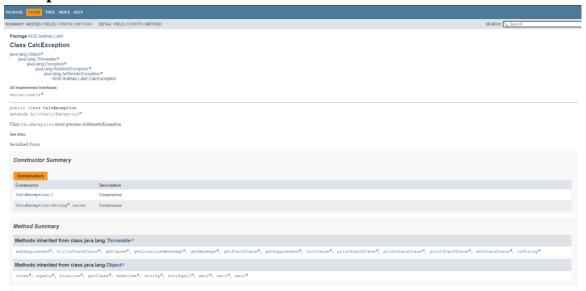
## Вміст файлу Result.txt

## Фрагмент згенерованої документації

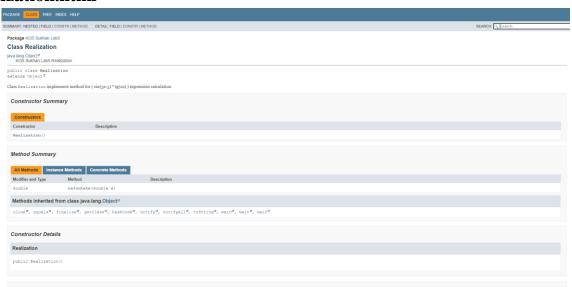
## App.html



## CalcException.html



#### Realization.html



#### Відповідь на контрольні запитання:

#### 1. Дайте визначення терміну «виключення».

**Відповідь:** <u>Виключення</u> – 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?

**Відповідь:** помилках введення, збоях обладнання, помилках, що пов'язані з фізичними обмеженнями комп'ютерної системи та помилках програмування

#### 3. Яка ієрархія виключень використовується у мові Java?

**Відповідь:** Всі виключення в мові Java поділяються на контрольовані і неконтрольовані та спадкуються від суперкласу Throwable. Безпосередньо від цього суперкласу спадкуються 2 класи Error і Exception.

#### 4. Як створити власний клас виключень?

Відповідь: заекстендити Error або Exception(або дочірні).

# 5. Який синтаксис оголошення методів, що можуть генерувати виключення?

#### Відповідь:

```
[access modifier] [return type] [name]([parameters]) throws Exception {
...
```

## 6. Які виключення слід вказувати у заголовках методів і коли?

**Відповідь:** Оголошувати слід лише всі контрольовані виключення. Якщо цього не зробити, то компілятор видасть повідомлення про помилку. Якщо метод оголошує, що він може генерувати виключення певного класу, то він може також генерувати виключення і його підкласів.

## 7. Як згенерувати контрольоване виключення?

#### Відповідь:

- 1. throw new Exception();
- 2. Exception ex = new Exception(); throw ex;

## 8. Розкрийте призначення та особливості роботи блоку try.

Відповідь: Блок виконання коду.

## 9. Розкрийте призначення та особливості роботи блоку catch.

Відповідь: Блок перехоплення виключення.

## 10. Розкрийте призначення та особливості роботи блоку finally.

**Відповідь:** Блок гарантованого виконання коду після try-catch блоку.

**Висновок:** Я оволодів навиками використання механізму виключень при написанні програм мовою Java.