

**Міністерство освіти і науки України**

**Національний університет “Львівська політехніка”**

**Кафедра ЕОМ**



**Звіт**

**З лабораторної роботи №5**

**Варіант – 21**

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

**На тему:** «Виключення»

**Виконав:** ст. гр. КІ-35

Сухан Д. В.

**Перевірив:** доцент кафедри ЕОМ

Іванов Ю.С.

**Львів 2022**

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

### Завдання (варіант № 21)

$$21. y = \sin(3x - 5) / \operatorname{ctg}(2x)$$

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.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.

### 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 реалізує 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 є числом, то генеруємо
            виключення
            if (y==Double.NaN || y==Double.NEGATIVE_INFINITY
            || y==Double.POSITIVE_INFINITY || 2*x == 90 || 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 App 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

```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL

PS R:\5_semester\KZP\CPPT_Sukhan_DV_KI-35_2> & 'C:\Program Files'
Enter X: 22
0.8446104334095649
Enter X: 11
0.18967882364931632
Enter X: 22
0.8446104334095649
Enter X: 1
-0.0012187172797260464
Enter X: 45
Exception reason: Illegal value of X for tangent calculation
PS R:\5_semester\KZP\CPPT_Sukhan_DV_KI-35_2>

```

## Вміст файлу Result.txt

```

KI35 > Sukhan > Lab5 > Result.txt

1 Result: 0.8446104334095649
2 Result: 0.18967882364931632
3 Result: 0.8446104334095649
4 Result: -0.0012187172797260464
5 Exception reason: Illegal value of X for tangent calculation

```

# Фрагмент згенерованої документації

## App.html

PACKAGE

CLASSES

TREE

INDEX

HELP

SUMMARY | NESTED | FIELD | CONSTR | METHOD | METHOD

DETAIL | FIELD | CONSTR | METHOD

SEARCH

Package KISS Sukhan Lab5

Class App

java.lang.Object<sup>#</sup>

KISS Sukhan Lab5 App

public class App  
extends Object<sup>#</sup>

Kiss App reaktiye prograssy-apalisp

Since:  
version 1.0

Constructor Summary

Constructors

Constructor	Description
App()	

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<sup>#</sup>

clone<sup>#</sup>, equals<sup>#</sup>, finalize<sup>#</sup>, getClass<sup>#</sup>, hashCode<sup>#</sup>, notify<sup>#</sup>, notifyAll<sup>#</sup>, toString<sup>#</sup>, wait<sup>#</sup>, wait<sup>#</sup>, wait<sup>#</sup>

Constructor Details

App

public App()

## CalcException.html

PACKAGE

CLASSES

TREE

INDEX

HELP

SUMMARY | NESTED | FIELD | CONSTR | METHOD | METHOD

DETAIL | FIELD | CONSTR | METHOD

SEARCH

Package KISS Sukhan Lab5

Class CalcException

java.lang.Object<sup>#</sup>

java.lang.Throwable<sup>#</sup>

java.lang.Exception<sup>#</sup>

java.lang.RuntimeException<sup>#</sup>

java.lang.ArithmeticException<sup>#</sup>

KISS Sukhan Lab5 CalcException

All implemented interfaces:  
Serializable<sup>#</sup>

public class CalcException  
extends ArithmeticException<sup>#</sup>

Class CalcException more preides ArithmeticException

See Also:  
Serialized Form

Constructor Summary

Constructors

Constructor	Description
CalcException()	Constructor
CalcException(String <sup>#</sup> cause)	Constructor

Method Summary

Methods inherited from class java.lang.Throwable<sup>#</sup>

addSuppressed<sup>#</sup>, fillInStackTrace<sup>#</sup>, getCause<sup>#</sup>, getLocalizedMessage<sup>#</sup>, getMessage<sup>#</sup>, getStackTrace<sup>#</sup>, getSuppressed<sup>#</sup>, initCause<sup>#</sup>, printStackTrace<sup>#</sup>, printStackTrace<sup>#</sup>, printStackTrace<sup>#</sup>, setStackTrace<sup>#</sup>, toString<sup>#</sup>

Methods inherited from class java.lang.Object<sup>#</sup>

clone<sup>#</sup>, equals<sup>#</sup>, finalize<sup>#</sup>, getClass<sup>#</sup>, hashCode<sup>#</sup>, notify<sup>#</sup>, notifyAll<sup>#</sup>, wait<sup>#</sup>, wait<sup>#</sup>, wait<sup>#</sup>

## Realization.html

PACKAGE

CLASSES

TREE

INDEX

HELP

SUMMARY | NESTED | FIELD | CONSTR | METHOD | METHOD

DETAIL | FIELD | CONSTR | METHOD

SEARCH

Package KISS Sukhan Lab5

Class Realization

java.lang.Object<sup>#</sup>

KISS Sukhan Lab5 Realization

public class Realization  
extends Object<sup>#</sup>

Class Realization implements method for ( sin(3x-3) \* tg(2x) ) expression calculation

Constructor Summary

Constructors

Constructor	Description
Realization()	

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
double	calculate(double x)	

Methods inherited from class java.lang.Object<sup>#</sup>

clone<sup>#</sup>, equals<sup>#</sup>, finalize<sup>#</sup>, getClass<sup>#</sup>, hashCode<sup>#</sup>, notify<sup>#</sup>, notifyAll<sup>#</sup>, toString<sup>#</sup>, wait<sup>#</sup>, wait<sup>#</sup>, wait<sup>#</sup>

Constructor Details

Realization

public Realization()

Method Details

## Відповідь на контрольні запитання:

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

**Відповідь:** 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?

**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.