Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



**Звіт**

З лабораторної роботи №5

Варіант – 6

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Виключення»

Виконав: ст. гр. КІ-35

Павлик С. В.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2022

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

**Завдання(Варіант 6)**

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:

1. Як створити власний клас виключень?

2. Як згенерувати контрольоване виключення?

**Варіант завдання: 6. y=tg(x)/sin(2x)**

**Код програми:**

**Файл CalcExeption.java**

package LAB\_05;

import java.io.\*;

import java.util.\*;

/\*\*

\* Class CalcExeption Implements driver for given expression calculator

\* @author Pavlyk Serhii

\* @version 1.0

\*/

public class CalcExeption

{

/\*\*

\* Method main start point to the program

\*/

public static void main(String[] args)

{

try

{

Scanner input = new Scanner(System.in);

PrintWriter fout = new PrintWriter(new File("result.txt"));

try

{

try

{

Equations eq = new Equations();

System.out.print("Enter X in degrees: ");

double res = eq.calculate(input.nextDouble());

System.out.print("tg(x) / sin (2x) = "+ res);

fout.print(res);

}

finally

{

fout.flush();

fout.close();

input.close();

}

}

catch (CalcException ex)

{

System.out.print(ex.getMessage());

}

}

catch (FileNotFoundException ex)

{

System.out.print("Exception reason: Perhaps wrong file path");

}

}

}

/\*\*

\* Class CalcException realize class implements a subclass

\* that catches arithmetic exception and provides an explanation

\* why this exception occurred

\* @author Pavlyk Serhii

\* @version 1.0

\*/

class CalcException extends ArithmeticException

{

/\*\*

\* Constructor

\*/

public CalcException()

{

}

/\*\*

\* Constructor

\* @param <code>cause</code> explanatory message, explane why the exception occurred

\*/

public CalcException(String cause)

{

super(cause);

}

}

/\*\*

\* Class realize method for tg(x) / sin (2x) expression calculation

\* @author Pavlyk Serhii

\* @version 1.0

\*/

class Equations

{

/\*\*

\* Method calculates the tg(x) / sin (2x) expression

\* @param <code>x</code> Angle in degrees

\* @throws CalcException

\* @return result of calculation

\*/

public double calculate(double x) throws CalcException

{

double y;

try

{

y = Math.tan(Math.toRadians(x))/Math.sin(Math.toRadians(2\*x));

if (Double.isNaN(y) || Double.isInfinite(y) || Double.isInfinite(-y) || x == 90 || x == -90)

throw new ArithmeticException();

}

catch (ArithmeticException ex)

{

if (x==90 || x==-90)

throw new CalcException("Exception reason: Illegal value of X for tg calculation");

else if (2\*x==0)

throw new CalcException("Exception reason: sin 2x = 0");

else

throw new CalcException("Unknown reason of the exception during exception calculation");

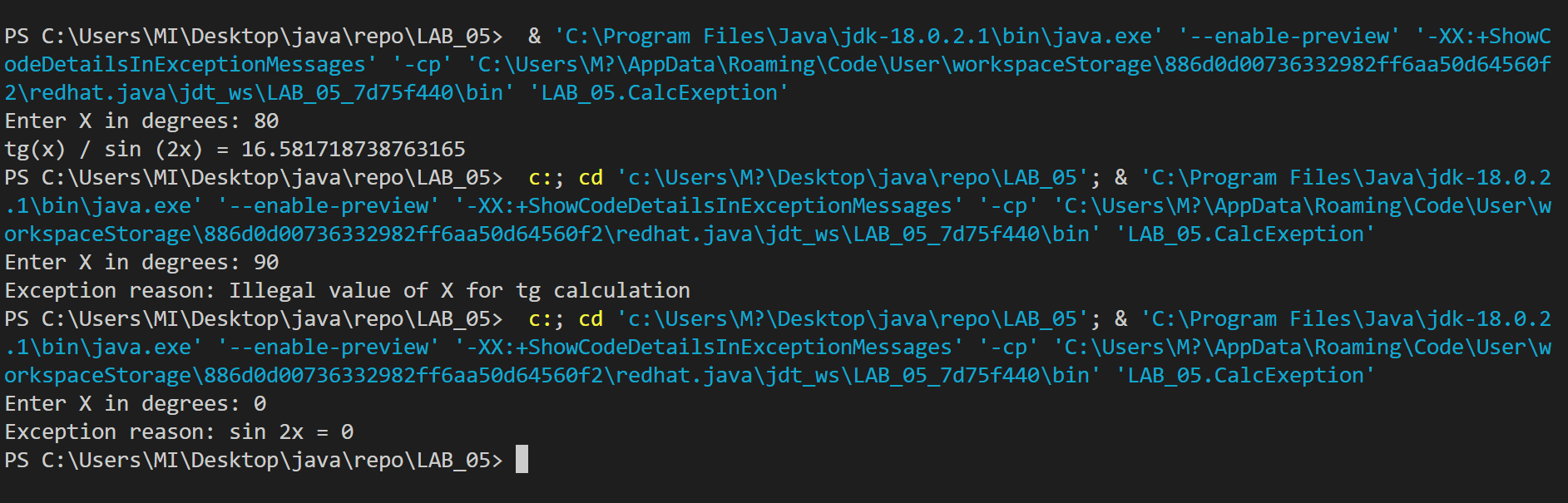
}

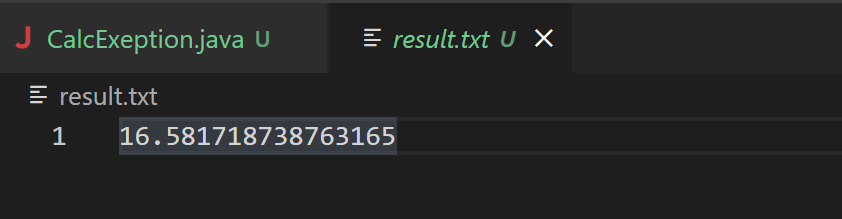
return y;

}

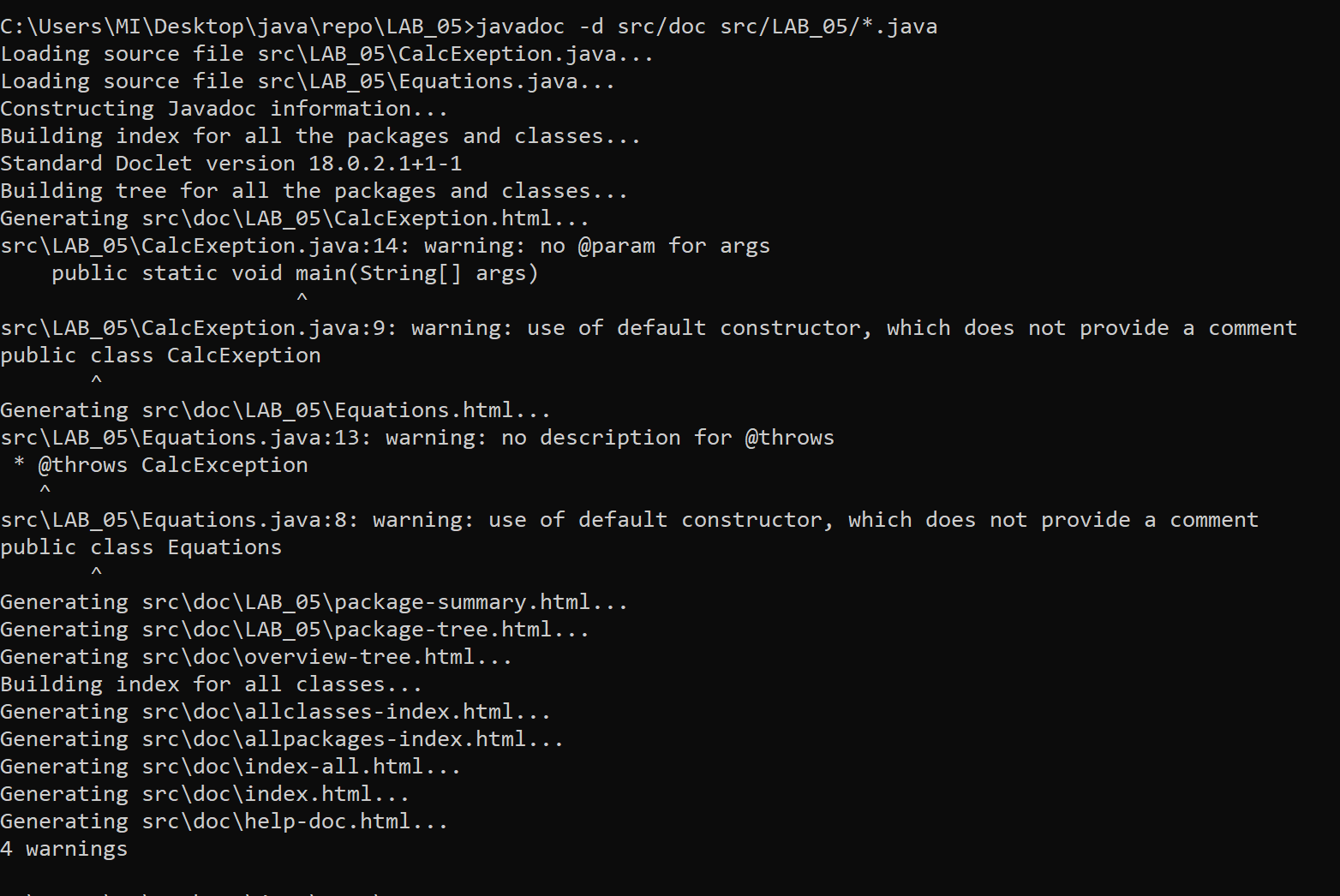
}

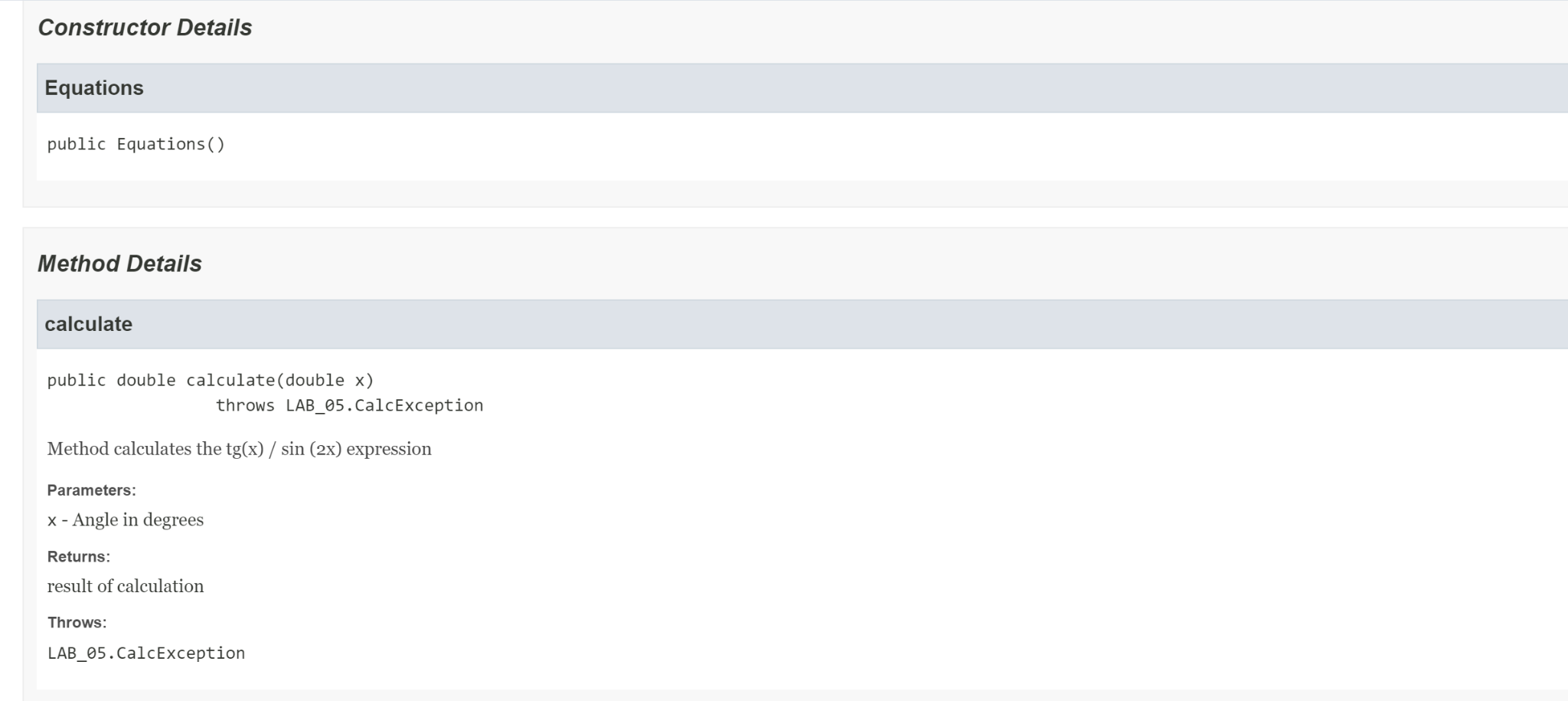
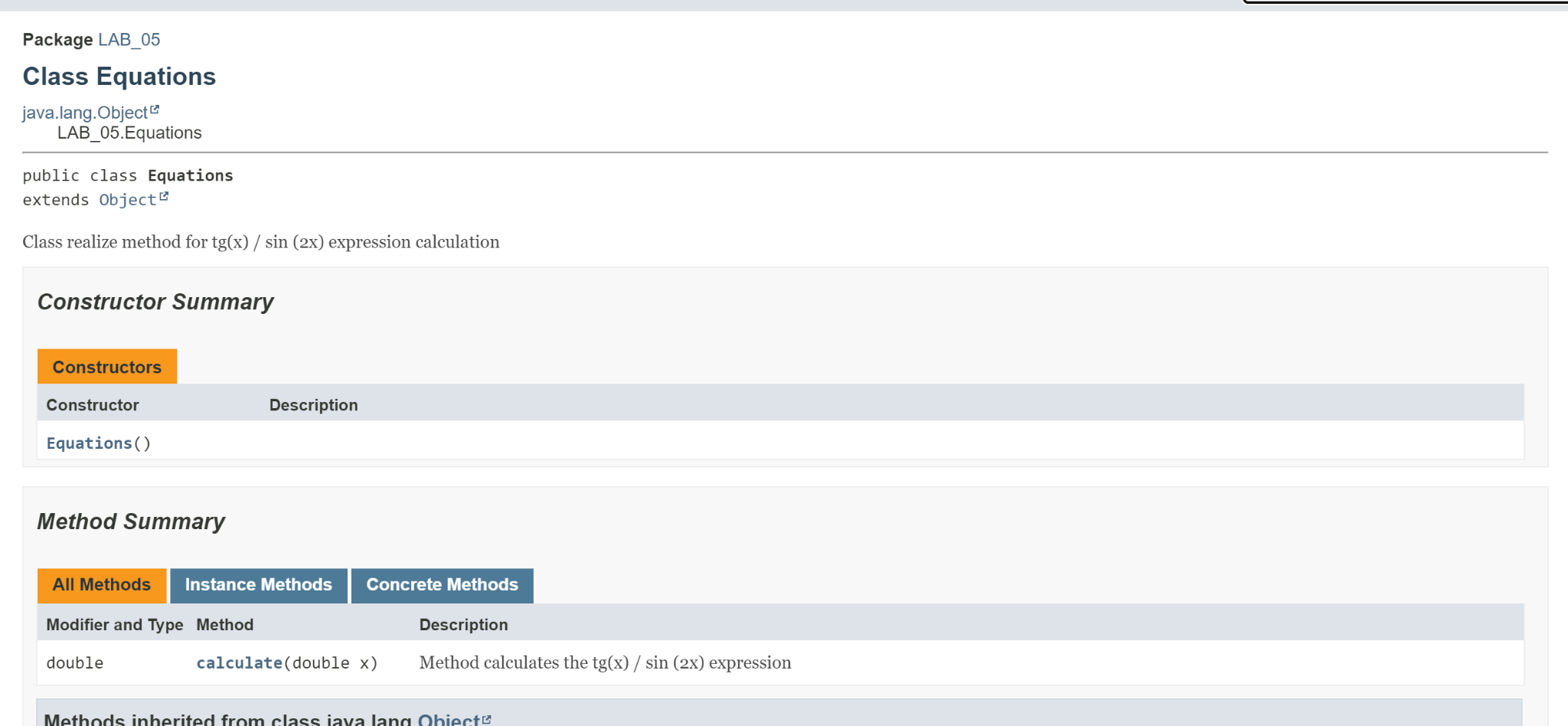
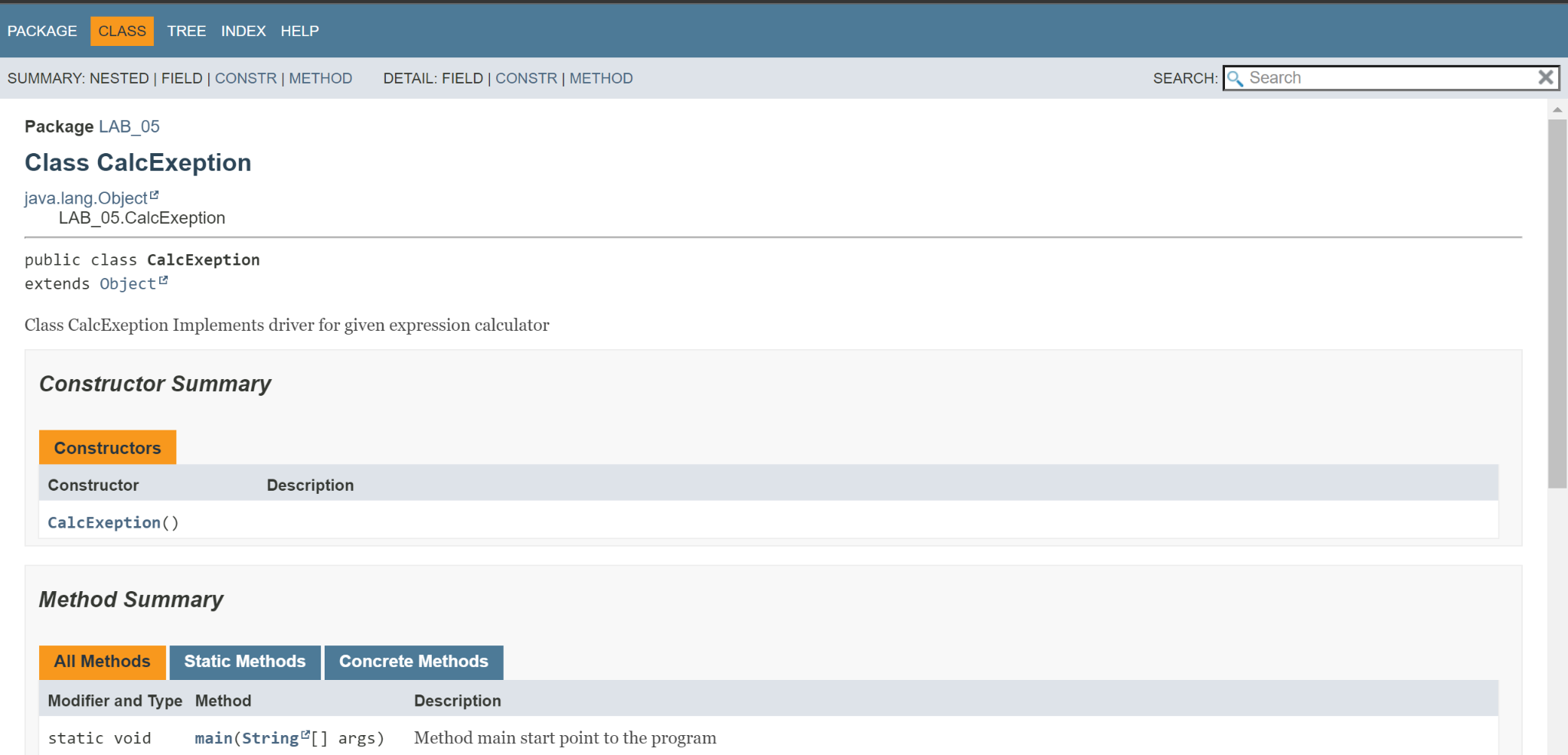
**Результати роботи програми:**

****

****

**Фрагмент згенерованої документації**

****

****

**Відповіді на контрольні запитання**

1. Для створення власного класу контрольованих виключень необхідно обов’язково успадкувати один з існуючих класів контрольованих виключень та розширити його новою функціональністю. Найчастіше власні класи оснащують конструктором по замовчуванню та конструктором, що приймає детальний опис ситуації, яка призвела до генерації виключення. Для відображення опису помилкової ситуації можна використати метод toString() класу Throwable. Для цього необхідно викликати відповідний конструктор класу, що розширяється. Після цього створений клас можна застосовувати для генерації виключень.
2. Виключення можуть генеруватися лише методами. Якщо метод може генерувати виключення певного класу, то назву цього класу слід вказати в заголовку методу після ключового слова throws. Якщо метод може генерувати кілька видів виключень, то всі вони перелічуються через кому. Приклад оголошення методу, що може генерувати виключення:

public int loadData(String fName) throws EOFException, MalformedURLException

{

…

}

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