Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 5

з дисципліни: «Кросплатформні засоби програмування»

на тему: «Виключення»

**Виконав:**

студент групи КІ-35

Скіпар І.В.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2022

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

**Завдання (варіант № 10)**

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:
5. Дайте визначення терміну «виключення».
6. Розкрийте призначення та особливості роботи блоку finally.

**Текст програми**

Equations.java

package KI35.Skipar.Lab5;  
  
import static java.lang.Math.\*;  
  
*/\*\*  
 \* Class <code>Equations</code> implements method for (tg(x)^2) expression  
 \* calculation  
 \** ***@author*** *Illia  
 \** ***@version*** *1.0  
 \*/*public class Equations {  
 */\*\*  
 \* Method calculate the (tg(x)^2) expression  
 \** ***@param*** *deg angle in degrees  
 \** ***@return*** *expression result  
 \** ***@throws*** *ArithmeticException check correct input  
 \*/* public double calculate(double deg) throws ArithmeticException  
 {  
 double res = 0, rad;  
 rad = *toRadians*(deg);  
  
 try {  
 res = *tan*(rad)\* *tan*(rad);  
 if ((int)(deg/90)%2!=0) throw new ArithmeticException();  
 }  
 catch (ArithmeticException e)  
 {  
 throw new ArithmeticException("Invalid X");  
 }  
  
 return res;  
 }  
}

EquationsApp.java

import KI35.Skipar.Lab5.\*;  
  
import java.io.FileNotFoundException;  
import java.io.PrintWriter;  
import java.util.Scanner;  
  
import static java.lang.System.*out*;  
  
public class EquationsApp {  
  
 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(fName);  
 try {  
 try {  
 Equations equations = new Equations();  
 *out*.print("Enter X: ");  
 fout.print(equations.calculate(in.nextDouble()));  
 } finally {  
 fout.flush();  
 fout.close();  
 *out*.print("End program\n");  
 }  
 }  
 catch (ArithmeticException e) {  
 *out*.print(e.getMessage());  
 }  
 }  
 catch (FileNotFoundException e) {  
 throw new RuntimeException(e);  
 }  
 }  
}

**Результат виконання програми**

Консоль

Enter file name: log.txt

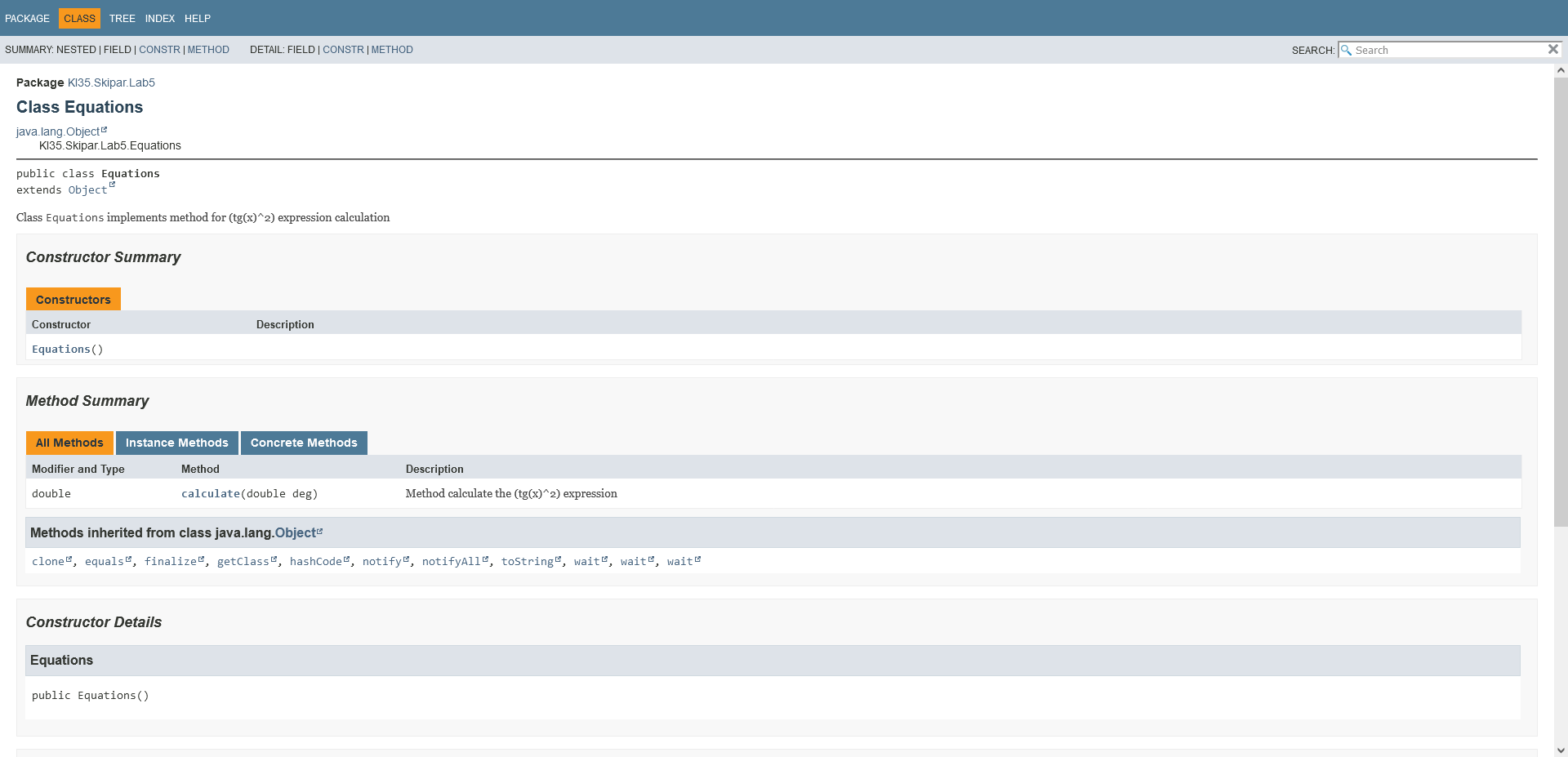
Enter X: 45

End program

Log.txt

0.9999999999999998

**Фрагмент згенерованої документації**



**Відповіді на контрольні запитання**

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

2. Особливість блоку finally полягає в тому щоб виконати дію в будь-якому випадку, навіть коли виконується помилка. Призначений цей блок здебільшого для закривання файлів.

**Висновок**

В результаті виконання лабораторної роботи я навчився працювати з обробленням помилок в мові Java.