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

Дзьобань М. Б.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2022

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

**Завдання (варіант № 3)**

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.

**Варіант 3:** y=sin(x)/cos(x)

1. Автоматично згенерувати документацію до розробленого пакету.
2. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
3. Дати відповідь на контрольні запитання

**Текст програми**

**Main.java**

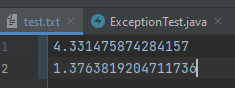
import KI\_35.Dzoban.Lab5.\*;  
  
  
import java.util.Scanner;  
import java.io.\*;  
import static java.lang.System.*out*;  
  
*/\*\*  
 \* Class Main implements driver for Calculation class  
 \** ***@author*** *Mykola Dzoban'  
 \** ***@version*** *1.0.0  
 \*/*public class Main {  
 */\*\*  
 \** ***@param*** *args Main class argument  
 \*/* 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  
 {  
 Calculation calc = new Calculation();  
 *out*.print("Enter X: ");  
 fout.print(calc.calculate(in.nextInt())+"\n");  
 *out*.print("Enter X for test: ");  
 fout.print(calc.calculate(in.nextInt()));  
 }  
 finally  
 {  
 fout.flush();  
 fout.close();  
 }  
 }  
 catch (CalcException | ExceptionTest ex)  
 {  
 *out*.print(ex.getMessage());  
 }  
  
 }  
 catch (FileNotFoundException ex)  
 {  
 *out*.print("Exception reason: Perhaps wrong file path");  
 }  
 }  
}

**Результат виконання програми**

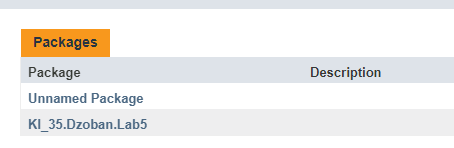
Enter file name: test.txt

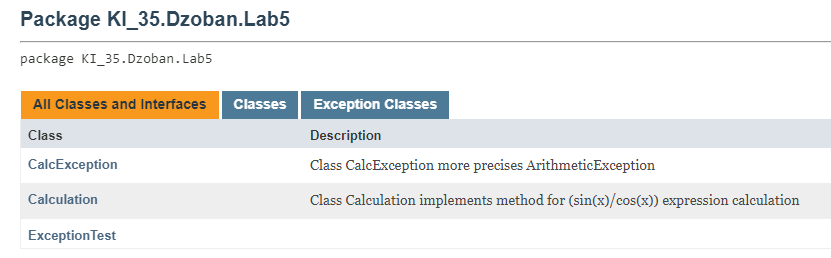
Enter X: 77

Enter X for test: 54



**Фрагмент згенерованої документації**





**Відповіді на контрольні запитання**

* 1. Дайте визначення терміну «виключення»

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

1. Яка ієрархія виключень використовується у мові Java?

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. Ієрархія класів, що спадкує клас Error, описує внутрішні помилки і ситуації, що пов’язані з браком ресурсів у системі підтримки виконання програм. Жоден об’єкт цього типу самостійно згенерувати неможна. При виникненні внутрішньої помилки можна лише відобразити повідомлення користувачу та спробувати коректно завершити виконання програми. Такі помилки є нечастими. Ієрархія класів, що спадкує клас Exception поділяється на клас RuntimeException та інші. Виключення типу RuntimeException виникають внаслідок помилок програмування. Всі інші помилки є наслідком непередбачених подій, що виникають під час виконання коректної програми, наприклад, помилок вводу/виводу.

1. Як створити власний клас виключень?

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.

**Висновок**

Під час виконання лабораторної, я оволодів навиками використання механізму виключень при написанні програм мовою Java.

**Додаток**

**Calculation.java**

package KI\_35.Dzoban.Lab5;  
  
*/\*\*  
 \* Class Calculation implements method for (sin(x)/cos(x)) expression calculation  
 \** ***@author*** *Mykola Dzoban  
 \** ***@version*** *1.0.0  
 \*/*public class Calculation {  
 */\*\*  
 \* Method calculate the (sin(x)/cos(x)) expression  
 \** ***@param*** *x Angel in degrees  
 \** ***@throws*** *CalcException  
 \*/* public double calculate(int x) throws CalcException, ExceptionTest {  
 double y, rad;  
 rad = (x \* Math.*PI*) / 180.0;  
  
 try {  
 y = Math.*sin*(rad) / Math.*cos*(rad);  
  
 if (y == Double.*NaN* || y == Double.*NEGATIVE\_INFINITY* || y == Double.*POSITIVE\_INFINITY* || x == 90 || x == -90 || x==30)  
 throw new ArithmeticException();  
 } catch (ArithmeticException ex) {  
 if (rad == Math.*PI* / 2.0 || rad == -Math.*PI* / 2.0)  
 throw new CalcException("Exception reason: Illegal value of X for tangent calculation");  
 else if( x==30 ){  
 throw new ExceptionTest("Test new exception");  
 }  
 else  
 throw new CalcException("Unknown reason of the exception during exception calculation");  
 }  
  
 return y;  
 }  
}

**CalcExeption.java**

package KI\_35.Dzoban.Lab5;  
  
*/\*\*  
 \* Class CalcException more precises ArithmeticException  
 \** ***@author*** *Mykola Dzoban  
 \** ***@version*** *1.0.0  
 \*/*public class CalcException extends ArithmeticException {  
 public CalcException(){}  
 public CalcException(String message)  
 {  
 super(message);  
 }  
}