Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра МКС



**Звіт**

З лабораторної роботи №5

З дисципліни: «Кросплатформні засоби програмування»

На тему:

«Виключення»

Виконав: ст.гр. КІ-35

Куденчук Владислав

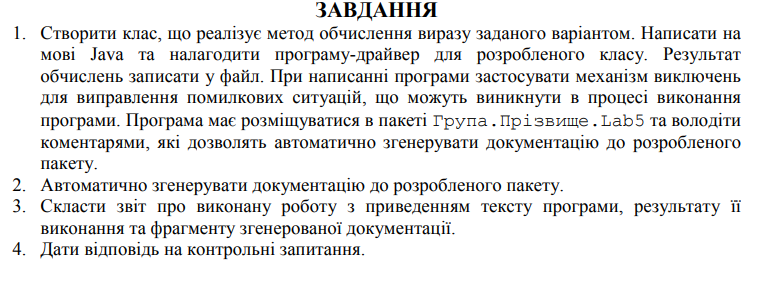
Прийняв:

Іванов Ю. С.

**Львів 2022**

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

**Завдання**





Виконання:

Код:

Class EquationsApp

package KI\_35.Kudenchuk.Lab5;  
import java.util.Scanner;  
import java.io.\*;  
import static java.lang.System.*out*;  
*/\*\*  
 \* Class <code>EquationsApp</code> Implements driver for Equations class  
 \** ***@author*** *Kudenchuk Vladyslav  
 \** ***@version*** *1.0  
 \*/*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 eq = new Equations();  
 *out*.print("Enter X: ");  
 fout.print(eq.calculate(in.nextDouble()));  
 }  
 finally  
 {  
 fout.flush();  
 fout.close();  
 }  
 }  
 catch (CalcException ex)  
 {  
 *out*.print(ex.getMessage());  
 }  
 finally  
 {  
 *out*.print("finally block");  
 }  
 }  
 catch (FileNotFoundException ex)  
 {  
 *out*.print("Exception reason: Perhaps wrong file path");  
 }  
 }  
}

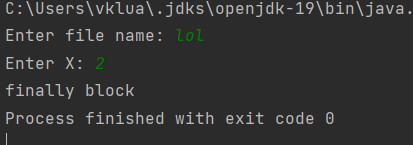
Class CalcExeption

package KI\_35.Kudenchuk.Lab5;  
  
*/\*\*  
 \* Class <code>CalcException</code> more precises ArithmeticException  
 \** ***@author*** *EOM Stuff  
 \** ***@version*** *1.0  
 \*/*class CalcException extends ArithmeticException  
{  
 public CalcException(){}  
  
 public CalcException(String cause)  
 {  
 super(cause);  
 }  
}

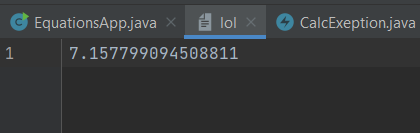
Class Equation

package KI\_35.Kudenchuk.Lab5;  
  
*/\*\*  
 \* Class <code>Equations</code> implements method for (x-4)/sin(3x-1) expression  
 \* calculation  
 \** ***@author*** *Kudenchuk Vladyslav  
 \** ***@version*** *1.0  
 \*/*class Equations  
{  
 public double calculate(double x) throws CalcException  
 {  
 double y, sin\_x;  
 sin\_x = 3 \* x \* Math.*PI* / 180.0f;  
 try  
 {  
 y = (x - 4.0f) / Math.*sin*(3 \* x);  
 if (Double.*isNaN*(y) || y==Double.*NEGATIVE\_INFINITY* ||  
 y==Double.*POSITIVE\_INFINITY* || sin\_x==0)  
 throw new ArithmeticException();  
 }  
 catch (ArithmeticException ex)  
 {  
 if (sin\_x==0)  
 throw new CalcException("Exception reason: division by zero (sin(3 \* x - 1) == 0)");  
 else  
 throw new CalcException("Unknown reason of the exception during exception calculation");  
 }  
 return y;  
 }  
}

Console:



Text file:



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