Міністерство освіти та науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



**ЗВІТ**

**з лабораторної роботи № 5**

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

**на тему:** «ВИКЛЮЧЕННЯ»

Виконав: ст. гр. КІ-35

Гудима О.П.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю.С

Львів-2022

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

**ЗАВДАННЯ**

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.

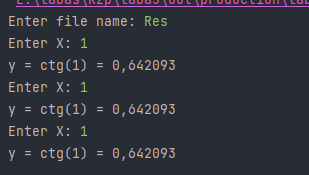
**Варіант 7:**

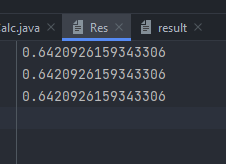
****

**Текст програми:**

|  |
| --- |
| ***Лістинг Calc***  *import static* java.lang.System.out;  *import* java.io.File; *import* java.io.FileNotFoundException; *import* java.io.PrintWriter; *import* java.util.Scanner; *public class* Calc {   *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*{  *for*(*int* i = 0; i<3;i++) {  Equations eq = *new* Equations();  out.print("Enter X: ");  fout.print(eq.calculate(in.nextInt())+ "\n");  }   }  *finally*{  fout.flush();  fout.close();  }  }  *catch* (CalcException | NewException ex){  out.print(ex.getMessage());  fout.print(ex.getMessage());  }  }  *catch* (FileNotFoundException ex){  out.print("Exception reason: Probably wrong file path");  }  } } *class* NewException *extends* ClassCastException {   } *class* CalcException *extends* ArithmeticException {  *public* CalcException(){}   *public* CalcException(String cause){  *super*(cause);  } } *class* Equations{  *public double* calculate(*int* x) *throws* CalcException{  *double* y, rad;  rad = x \* Math.PI / 180.0;  *try*{  y = 1/Math.tan(x);  System.out.printf("y = ctg(%d) = %f \n", x , 1/Math.tan(x));  *if* (y==Double.NaN || y==Double.NEGATIVE\_INFINITY || y==Double.POSITIVE\_INFINITY || x==90 || x== -90)  *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==0)  *throw new* CalcException("Exception reason: X = 0");  *else  throw new* CalcException("Unknown reason of the exception during exception calculation");  }   *return* y;  }  } |

**Результат роботи:**

****

****

**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 виникненні уникаючи процесу розкручування стеку.

1. У яких ситуаціях використання виключень є виправданим?

Генерація виключень застосовується при:

* + помилках введення, наприклад, при введенні назви неіснуючого файлу або Інтернет адреси з подальшим зверненням до цих ресурсів, що призводить до генерації помилки системним програмним забезпеченням;
  + збоях обладнання;
  + помилках, що пов’язані з фізичними обмеженнями комп’ютерної системи, наприклад, при заповненні оперативної пам’яті або жорсткого диску;
  + помилках програмування, наприклад, при некоректній роботі методу, читанні елементів порожнього стеку, виходу за межі масиву тощо.

**Висновок:**

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