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

Хван Д.В.

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 23:**

****

**Текст програми:**

|  |
| --- |
| /\*\*   \* package   \*/  import java.util.Scanner;  import java.io.\*;  import static java.lang.System.out;  /\*\*   \* Class <code>App</code> Implements driver for Equation class   \*   \* @author Khvan Danylo   \* @version 1.0   \*/  public class App {      /\*\*       \* @param args       \*/      public static void main(String[] args) {          try{          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                  {                      Equation eq = new Equation();                      out.print("Enter X: ");                      fout.print(eq.calculator(in.nextDouble()));                  }                  finally                  {                      fout.flush();                      fout.close();                  }              }              catch (Exceptions e)              {                  out.print(e.getMessage());              }          }          catch (FileNotFoundException ex)          {              out.print("Exception reason: Perhaps wrong file path");          }      } finally {          var exception = ExceptionVal.exceptionMeaning.get(ExceptionVal.divideByZero);          throw new ExceptionR(exception);      }      }  }  /\*\*   \* package   \*/  /\*\*   \* Class <code>Equation</code> implements method for (ctg(8\*x) / x) expression   \* calculation   \* @author Khvan Danylo   \* @version 1.0   \*/  public class Equation {      /\*\*       \* Method calculates the (ctg(8\*x) / x) expression       \* @param X Angle in degrees       \* @throws Exceptions       \* @return (ctg(8\*x) / x)       \*/      public double calculator(double X) throws Exceptions {          double y, radian;          radian = Math.toRadians(8\*X);          try{              y= 1.0 / Math.tan(radian);              if(y==Double.NaN || y==Double.NEGATIVE\_INFINITY || y==Double.POSITIVE\_INFINITY || X\*8==90 || X\*8== -90 || X==0){                  throw new ArithmeticException();              }              y = y / X;          }catch  (ArithmeticException e){                var exception = ExceptionVal.exceptionMeaning.get(ExceptionVal.unknownException);              if(radian == Math.PI/2.0 || radian == -Math.PI/2.0){                  exception = ExceptionVal.exceptionMeaning.get(ExceptionVal.illegalValue);                  throw new Exceptions(exception);              }else if(X == 0){                  exception = ExceptionVal.exceptionMeaning.get(ExceptionVal.divideByZero);                  throw new Exceptions(exception);              }                    throw new Exceptions(exception);          }          System.out.println(radian);          System.out.println( Math.tan(radian)/X);          return y;      }    }  public class ExceptionR extends ArithmeticException {        /\*\*       \* Constructor       \*/      public ExceptionR() {      }      /\*\*       \* Constructor with string       \*       \* @param error       \*/      public ExceptionR(String error) {          super(error);          }    }  /\*\*   \* Class <code>Exceptions</code> more precises ArithmeticException   \*   \* @author EOM Stuff   \* @version 1.0   \*/  public class Exceptions extends ArithmeticException {        /\*\*       \* Constructor       \*/      public Exceptions() {      }      /\*\*       \* Constructor with string       \*       \* @param error       \*/      public Exceptions(String error) {          super(error);          }    }  import java.util.HashMap;  public enum ExceptionVal {      illegalValue,      divideByZero,      unknownException;      public static HashMap<ExceptionVal, String> exceptionMeaning = new HashMap<>() {          {              put(ExceptionVal.illegalValue, "Assertion failed: Illegal value of X for cotangent calculation");              put(ExceptionVal.divideByZero, "Assertion failed: X is equal 0, it's not possible to divide by 0");              put(ExceptionVal.unknownException,                      "Assertion failed: Unknown reason of the exception during exception calculation");          }      };  } |

**Результат роботи:**

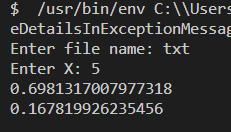
****

Рис.1.1 Результат роботи програми

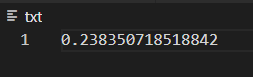
****

Рис.1.2 Результат роботи програми у файлі

**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.