

Міністерство освіти і науки України

Національний університет «Львівська політехніка»



Звіт

З лабораторної роботи №5

З дисципліни «Кросплатформенні засоби програмування»

На тему: «Виключення»

Виконав:
ст.гр. КІ-36
Литовко С.Г
Прийняв:
Іванов Ю.С

Львів-2022

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

ТЕОРЕТИЧНІ ВІДОМОСТІ

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

Ієрархія класів виключень

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.

Класи, що спадкуються від *Error* та *RuntimeException*, відносяться до неконтрольованих виключень. Всі інші класи відносяться до контрольованих виключень. Лише контрольовані виключення можуть бути згенеровані програмістом у кодї програми явно за допомогою ключового слова *throw*.

Для всіх контрольованих виключень компілятор перевіряє наявність відповідних обробників.

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

$$6. y = \operatorname{tg}(x) / \sin(2x)$$

Код програми:

App.java

```
App.java ×
1 package KI36.Lytovko.lab05;
2
3 import java.util.Scanner;
4 import java.io.*;
5 import static java.lang.System.out;
6
7 public class App {
8     public static void main(String[] args)
9     {
10         try
11         {
12             out.print("Enter file name: ");
13             Scanner in = new Scanner(System.in);
14             String fName = in.nextLine();
15             PrintWriter fout = new PrintWriter(new File(fName));
16             try
17             {
18                 try
19                 {
20                     Equations eq = new Equations();
21                     out.print("Enter X: ");
22                     fout.print(eq.calculate(in.nextInt()));
23                 }
24                 finally
25                 {
```

```

26         fout.flush();
27         fout.close();
28     }
29     }
30     catch (CalcException ex)
31     {
32         out.print(ex.getMessage());
33     }
34 }
35 catch (FileNotFoundException ex)
36 {
37     out.print("Exception reason: Perhaps wrong file path");
38 }
39 }
40 }
41
42
43 class CalcException extends ArithmeticException
44 {
45     public CalcException() {}
46
47     public CalcException(String cause)
48     {
49         super(cause);
50     }
51 }
52
53
54 class Equations
55 {
56     public double calculate(int x) throws CalcException
57     {
58         double y, rad;
59         rad = x * Math.PI / 180.0;
60         try
61         {
62             y = Math.tan(rad) / Math.sin(2 * rad);
63             if (y==Double.NaN || y==Double.NEGATIVE_INFINITY || y==Double.POSITIVE_INFINITY || x==90 || x== -90)
64                 throw new ArithmeticException();
65         }
66         catch (ArithmeticException ex)
67         {
68             if (rad==Math.PI/2.0 || rad==Math.PI/2.0)
69                 throw new CalcException("Exception reason: Illegal value of X for tangent calculation");
70
71             else if (x==0)
72                 throw new CalcException("Exception reason: X = 0");
73             else
74                 throw new CalcException("Unknown reason of the exception during exception calculation");
75         }
76     }
77     return y;
78 }
79
80 }

```

Результат програми:

```

<terminated> App (7) [Java Application] C:\
Enter file name: lab_05
Enter X: 2

```

lab_05: Блокнот

Файл Редагувати Переглянути

0.5006097300709479

Висновок: Зробивши цю лабораторну роботу, я оволоділа навиками використання механізму виключень при написанні програм мовою Java.