

ใบงานการทดลองที่ 12

เรื่อง การใช้คำสั่ง try catch และ throw exception

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการใช้วัตถุ การทำงานหลายๆอย่างพร้อมกัน และการติดต่อระหว่างงาน
- 1.2. รู้และเข้าใจการจัดการกับความผิดปกติในการเขียนโปรแกรมเชิงวัตถุ

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

- 3.1. Java Exception คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เป็นวิธีการจัดการข้อผิดพลาดที่เกิดขึ้นในขณะที่โปรแกรมทำงาน

```
1 import java.util.Scanner;
2 public class TestException1 {
3     public static void main (String[] args) {
4         Scanner reader = new Scanner(System.in);
5         int x;
6
7         System.out.print("Enter number: ");
8         x = reader.nextInt();
9         System.out.println("Your number is " + x);
10    }
11 }
```

- 3.2. คำสั่ง try มีลักษณะการทำงานอย่างไร?

```
Try {
    Jorhvsosd
    Int answer = j / s ;
}
```

- 3.3. คำสั่ง catch มีลักษณะการทำงานอย่างไร?

```
catch (InputMismatchException ex) {
    System.out.println("Exception occurred: " + ex);
}
```

- 3.4. คำสั่ง finally มีลักษณะการทำงานอย่างไร?

เป็นคำสั่งจบของภาษา Java ซึ่งเราสามารถใช้ในการประกาศตัวแปร, method และ class ได้ด้วย โดยที่มันจะมีความหมายแตกต่างกันออกไปขึ้นอยู่กับว่าเราไปใช้ในการประกาศอะไร

```
1 finally {
2     System.out.println("bar's finally");
3 }
```

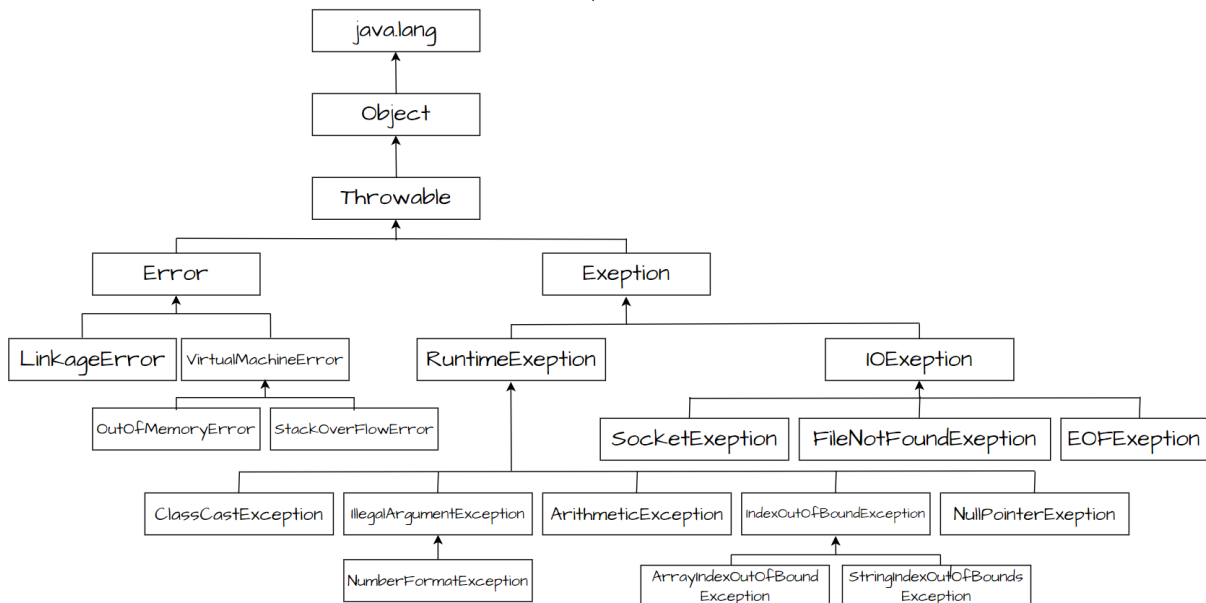
- 3.5. ลักษณะโครงสร้างของคำสั่ง try catch เป็นอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

เป็นคำสั่งที่ใช้สำหรับกำหนดบล็อกเพื่อตรวจสอบและจัดการกับข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม

```
1 package com.java.myapp;
2 public class MyClass {
3     public static void main(String[] args) {
4         try {
5             int x = 200 ;
6             int y = 0 ;
7             int z = x / y;
8             System.out.println(" x / y = " + z) ;
9         } catch(Exception e) {
10             System.out.println("Error : " + e.getMessage());
11         }
12     }
13 }
```

4. ลำดับขั้นการปฏิบัติการ

- 4.1. จากผังงานต่อไปนี้ จงเขียนโค้ดโปรแกรมเพื่อแสดงตัวอย่างการจัดการความผิดปกติของคลาสการจัดการสิ่งผิดปกติจนครบทุกคลาส (เน้นเฉพาะส่วนของ Error และ Exception)



ตัวอย่างโค้ดโปรแกรมการจัดการสิ่งผิดปกติในส่วนของ Error

```
1  try {
2      int[] array = new int[ 1000*1000*1000 ] ;
3
4  }catch( OutOfMemoryError e ) {
5      System.out.println( " Array Size too large " );
6  }//end catch
7  try {
8      Stack<Integer> st = new Stack<>();
9  }catch( StackOverflowError e ) {
10     System.out.println( " Stack Overflow " );
11 }//end catch
```

ตัวอย่างโค้ดโปรแกรมการจัดการสิ่งผิดปกติในส่วนของ Exception

```

1  try {
2      String[] strArray = new String[] { "John", "Snow" };
3      ArrayList<String> strList = (ArrayList<String>) Arrays.asList(strArray);
4      System.out.println("String list: " + strList);
5      }catch( ClassCastException e ) {
6          System.out.println( " ClassCastException " );
7      }
8      try {
9          int a = Integer.parseInt(null);
10         }catch( NumberFormatException e ) {
11             System.out.println( " NumberFormatException " );
12         }
13         try {
14             int a = 5 / 0 ;
15         }catch( ArithmeticException e ) {
16             System.out.println( " ArithmeticException " );
17         }
18         try {
19
20             int[] a = 5 / 0 ;
21
22         }catch( ArithmeticException e ) {
23
24             System.out.println( " ArithmeticException " );
25
26         }
27         try {
28
29             int[] a = { 2 , 3 , 4 } ;
30             System.out.println( a[10] );
31         }catch( ArrayIndexOutOfBoundsException e ) {
32
33             System.out.println( " ArrayIndexOutOfBoundsException " );
34
35         }
36         }
37         try {
38
39             String str = "Java Code Geeks!";
40             CharSequence seq = str.subSequence(10, 20);
41
42         }catch( StringIndexOutOfBoundsException e ) {
43
44             System.out.println( " StringIndexOutOfBoundsException " );
45
46         }
47         try {
48
49             String a = null;
50             System.out.println(a.toString());
51
52         }catch( NullPointerException e ) {
53
54             System.out.println( " NullPointerException " );
55
56         }
57         try {
58
59             client.sendMessage("hi");
60             client.sendMessage("hi again");
61
62         }catch( SocketException e ) {
63
64             System.out.println( " SocketException Connection Lost " );
65
66         }
67         }
68         try {
69             FileReader reader = new FileReader("file.txt");
70
71         }catch( FileNotFoundException e ) {
72
73             System.out.println( " SocketException " );
74
75         }
76         try {
77
78             DataInputStream dis = new DataInputStream(new
79             FileInputStream("D:\\data.txt"));
80             while(true) {
81                 char ch;
82                 ch = dis.readChar() ;
83                 System.out.print(ch);
84             }
85         }catch( EOFException e ) {
86
87             System.out.println( " EOFException " );
88
89         }

```

5. สรุปผลการปฏิบัติการ

การใช้ try เป็นส่วนหนึ่งที่ทำให้โปรแกรม exception ขึ้นได้ และสามารถทำ exception ให้ตรวจจับข้อมูลที่มีส่วนผิดพลาดได้

6. คำถามท้ายการทดลอง

6.1. เพราะเหตุใดการใช้ `catch(Exception e)` ; จึงไม่เหมาะกับการจัดการสิ่งผิดปกติที่ดีที่สุด
เพราะตัวโปรแกรมที่สร้างมามีแนวโน้ม อาจ ตรวจจับส่วนของ error ได้ ว่ามันจะ error ตรงไหน

6.2. การจัดการสิ่งผิดปกติจากการตัวเลขต่างๆ ด้วยเลขศูนย์ ควรเลือกใช้วิธีใด?
`Catch(ArithmeticException e){ }`

6.3. การจัดการสิ่งผิดปกติจากการเรียกใช้งาน Element เกินขนาดของอาร์เรย์ ควรเลือกใช้วิธีใด?
`Catch(ArrayIndexOutOfBoundsException e){ }`
