“使用者亂搞一下，你的程式就死掉”

“學了 try catch 你的程式就 robust 了!”

Exception 異常(例外)處理

你的程式寫得非常完美，使用者怎麼亂搞都不會出錯嗎?

經常看到的出錯情況:

- 規定輸入整數，卻輸入有小數點的數
- 規定輸入整數，卻輸入字串
- 分母被 0 除
- 陣列索引出界
- 開檔案存檔案失敗

這些情況就是程式發生了異常 Exception 例外!!

例外(Exception)是指程式發生異常(錯誤),導致程式無法繼續執行!

如何捕捉異常(例外)狀況?

try{...}catch(){...} 可以搞定一切!

範例:相除運算之異常

```
int n1 = 10;
int n2 = 0;
int ans;

ans = number1 / number2 ;
```

發生了何種例外?

程式設計師可以用 if 避免產生異常

```
int n1 = 10;
int n2 = 0;
int ans;

if (n2 != 0)
{
   ans = n1 / n2 ;
}
else
{
   System.out.println( "小學老師教過分母不可為 0" );
}
```

程式設計師可以用 try 避免產生異常

```
int n1 = 10;
int n2 = 0;
int ans;

try
{
   ans = n1 / n2 ;
}
catch(ArithmeticException ex)    //Exception ex
{
   System.out.println( "小學老師教過分母不可為 0" );
}
```

範例 2 陣列操作之異常

```
int[] array = new int[3];

(1):
System.out.println(array[3]);

發生了何種例外?
ArrayIndexOutOfBoundsException: 3

(2):
System.out.println("請輸入一個整數:");
array[2] = input.nextInt();
//使用者不小心輸入小數點，發生了何種例外?
```

```
int[] array = new int[3];
try {
        System.out.println(array[3]);
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("陣列索引出界了!");
        e.printStackTrace();
    }
```

```
public class MyException {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        int[] array = new int[3];
```

```
        try {
        System.out.println(array[3]);
        System.out.println("請輸入一個整數:");
        array[2] = input.nextInt();
        //使用者不小心輸入小數點，這個異常 可以被捕捉嗎?

        } catch( ArrayIndexOutOfBoundsException    e    )
        {
            System.out.println("陣列索引出界了!");
            e.printStackTrace();
        } catch ( Exception e) //在這裡可以捕捉到輸入小數點的異常
        {
            System.out.println("發生其他異常了!");
             System.out.println(e.toString());
        }
        finally{
            System.out.println("程式已執行完畢");
        }
}
```

try{...}區塊中打上我們要執行的程式碼，
catch()可以設定要處理的 Exception 類型，
catch(){...}區塊則是異常時要執行的動作，

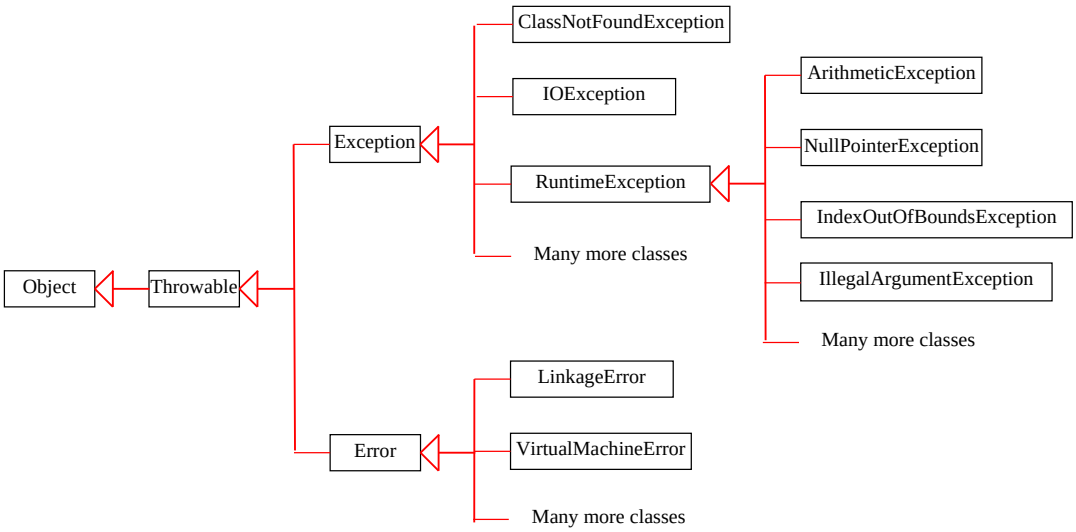
不管有沒有發生異常，最後都能執行某一段程式，就會搭配 finally{...}

```
        try {
                System.out.println(array[3]);
        } catch (Exception e) {
                System.out.println("發生異常!!");
        } finally {
                System.out.println("程式已執行完畢");
        }
```

不管程式執行是否發生異常，最後都會執行到 finally{...}區塊內的動作，常被用在資源釋放，例如在程式最後關閉資料庫連線或刪除暫存檔案等。

有哪些例外類別?

一大堆，多到你記不得!!



來源:教科書

如何自行定義拋出異常(例外)?

前面的實例都是系統自動拋異常，你可以設計自己的拋出異常。

throws Exception 拋出異常

如果在該寫 try catch 包住程式碼時，你沒寫，把責任都給外面的人(程式)去處理。用 throws 拋出例外即可。(通常用在方法的名稱後面)

```
public class MyThrowException {
```

```
    public static double sqrt2(int n) throws Exception {
        return Math.sqrt(n);
    }
    public static double sqrt(int n) throws Exception {
        if (n >= 0) {
            return Math.sqrt(n);
        } else {
            //System.out.println("throw 異常!"); //可以在此顯示異常訊息
            throw new Exception("不可小於零!");
        }
    }

    public static void main(String[] args) {
    //呼叫該方法時，要寫捕捉 try catch
        try {
            System.out.println(sqrt(4));
            System.out.println(sqrt(-4));
        } catch (Exception ex) {
            System.out.println("捕抓到異常!");
            System.out.println(ex.toString());
            System.out.println(ex.getMessage());
        }
    }
}
```

範例 3 檔案存取之異常

詳情請見檔案存取講義

寫法 1:

```
public static void main(String[] args)
{
```

```
    try {
        Formatter out = new Formatter("doc.txt");
        out.format("%s %s\n", "編號", "數量");
        out.format("%s %d\n", "A10", 25);
        out.close();
    catch (FileNotFoundException e) {
        System.out.println("存檔出了問題!");
    }
}
```

寫法 2:

```
        public static void main(String[] args) throws FileNotFoundException {
            Formatter out = new Formatter("doc.txt");
            String msg = "大家好，歡迎來到 Java 的世界\n ";
            out.format("%s", msg);   //準備好 文字內容
            out.flush();     //馬桶沖水!   將緩衝區的資料串流 stream 沖出去
            out.close();     //蓋上馬桶   將資料串流 stream 關閉
        }
```

複雜些的寫法，自己寫的方法:

```
public static void readFile() throws FileNotFoundException {
    Formatter out = new Formatter("doc.txt");
    out.format("%s %s\n", "編號", "數量");
    out.format("%s %d\n", "A10", 25);
    out.close();
 }

public static void main(String[] args)
{
     try{
         readFile();
     }
```

```
    catch ( FileNotFoundException e ){
       System.out.println("存檔出了問題!");
    }
}
```