



習題

13.1 例外的基本觀念

1. 請先閱讀下面的程式碼，嘗試瞭解其中每一行的意義，並試著回答接續的問題：

```
01 // hw13_1, 例外訊息的擷取
02 public class hw13_1
03 {
04     public static void main(String args[])
05     {
06         int num=12,den=0;
07         int ans=num/den;
08         System.out.println("end of main()!!");
09     }
10 }
```

- (a) 在編譯此程式時，會不會有錯誤訊息產生？
- (b) 執行 hw13_1 後，系統會拋出如下的例外：

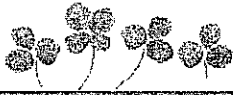
Exception in thread "main" java.lang.ArithmeticException: / by zero

試說明這個例外的涵義，並指出為什麼會產生這個例外。

- (c) 於本例中，程式碼的第 8 行是否會被執行？試說明會或不會的原因。
2. 試修改習題 1 的程式碼，加入 if-else 的判斷式，若 den 為 0，則跳過除法的運算，並印出 "除數為 0" 的字串，若 den 不為 0，則進行除法運算。
 3. 接續習題 1，試加入 try-catch-finally 區塊，使得 catch() 可以捕捉 ArithmeticException 例外。

13.2 例外類別的繼承架構

4. 下面的程式碼裡存在有兩個錯誤，第一個是除數為 0，第二個是陣列索引值超出了範圍。試在此程式碼內加入 if-else 敘述，使得當這兩種錯誤有任何一種發生時，程式便會停止執行，並印出 "程式執行有誤" 字串。



```
01 // hw13_4
02 public class hw13_4
03 {
04     public static void main(String args[])
05     {
06         int num=12;
07         int den[]={12,0,3,0,0,4};
08
09         for(int i=0;i<10;i++)
10             System.out.println("ans="+num/den[i]);
11     }
12 }
```

5. 試修改習題 4 的程式碼，使用 try-catch 區塊來捕捉由錯誤而產生的例外，其中 catch 區塊應可同時捕捉 "除數為 0" 與 "陣列索引值超出了範圍" 這兩種錯誤所造成的例外。
6. 試修改習題 4 的程式碼，使用 try-catch 區塊來捕捉由錯誤而產生的例外，其中 catch() 可捕捉所有可能由系統拋出的例外，並印出 "捕捉到例外了" 字串（提示：由 catch() 來捕獲 Exception 例外類別的物件）。

13.3 拋出例外

7. 試修改習題 1，將第 6~7 行的程式碼寫在一個可拋出 ArithmeticException 的 test() 函數內，並在 main() 內撰寫程式碼來捕捉由 test() 所拋出的例外（請參考 app13_6 的寫法）。
8. 接續習題 7，試將習題 7 所撰寫的 test() 函數改寫在一個獨立的 Ctest 類別內，使得 ArithmeticException 例外是由 Ctest 類別內的 test() 所拋出（請參考 app13_7 的寫法）。
9. 如果把 app13_7 的 aaa() 改為「實例函數」，而非「類別函數」，試修改 main() 使得其輸出結果與 app13_7 相同。

13.4 自己撰寫例外類別

10. 試修改 app13_8，使得當半徑大於 100 時，會拋出 RadiusTooLarge 例外，且半徑小於 0 時，拋出 RadiusIsNegative 例外。



11. 試修改 app13_8，在 CCircleException 類別內加入 toString() 函數，使得程式碼的 35 行若修改成下面的敘述時

```
System.out.println(e);
```

可印出 "CCircleException 例外被拋出了" 字串。

12. 三角形的三個邊為 a、b、c。當 $a=b=c$ 時，即構成正三角形。下面的程式，可以判斷是否為正三角形：

```
01 // hw13_12,
02 public class hw13_12
03 {
04     public static void main(String args[])
05     {
06         int a=3;
07         int b=3;
08         int c=3;
09
10         if((a+b)<c || (a+c)<b || (b+c)<a)
11             System.out.println("不構成三角形");
12         else if(a==b && a==c && b==c)
13             System.out.println("這是正三角形");
14         else
15             System.out.println("這不是正三角形");
16     }
17 }
```

請將判別三角形的程式碼撰寫到 void triangle(int a,int b,int c) 函數，並利用 try-catch 區塊捕捉自訂的例外，這些例外由 triangle() 拋出。請完成下面各題的要求：

- (a) 若此三邊不能構成一個三角形，則拋出自訂的 NotTriangle，並印出 "不構成三角形" 字串。
- (b) 若此三角形為直角三角形，則拋出自訂的 EquilateralTriangle，並印出 "這是正三角形" 字串。
- (c) 若此三角形不為直角三角形，則拋出自訂的 NotEquilateralTriangle，並印出 "這不是正三角形" 字串。



13.5 拋出輸出/輸入的例外類別

13. 試撰寫一程式，可以由鍵盤輸入一字串。若字串之值為 "520"，則拋出 `Exception520` 這個例外物件，並印出下面的字串：

"這是由字串 520 所引起的例外"

如果輸入的字串不為 "520"，則印出原來輸入的字串。

14. 試撰寫一程式，由鍵盤輸入一整數，再依序完成題目的要求：

- (a) 若整數之值小於 10，則拋出 `IntegerTooSmall` 例外，並印出下面的字串：

"您輸入的整數的值太小"

- (b) 若整數之值大於 70，則拋出 `IntegerTooLarge` 例外，並印出下面的字串：

"您輸入的整數的值太大"

- (c) 若整數之值介於 10 和 70 之間，印出原來的數值。

- (d) 若輸入的不是整數，即拋出 `InputMismatchException` 例外，並印出下面的字串：

"您輸入的不是整數"

15. 試設計 `void mySqrt(int n)` 函數，計算 `n` 的開根號值。請於 `mySqrt()` 中判別 `n` 是否大於 0；若 `n` 小於 0，則拋出自訂的 `ArgumentOutOfBound` 例外，並印出字串 "`n` 小於 0"，其中 `n` 為實際的變數值。請利用 `BufferedReader` 輸入 `n`，`IOException` 例外請直接由 `main()` 拋出。

16. 接續習題 15，但 `IOException` 例外請由 `try-catch` 區塊來捕捉。