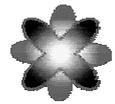


## 第五章 條件處理

### 重點整理

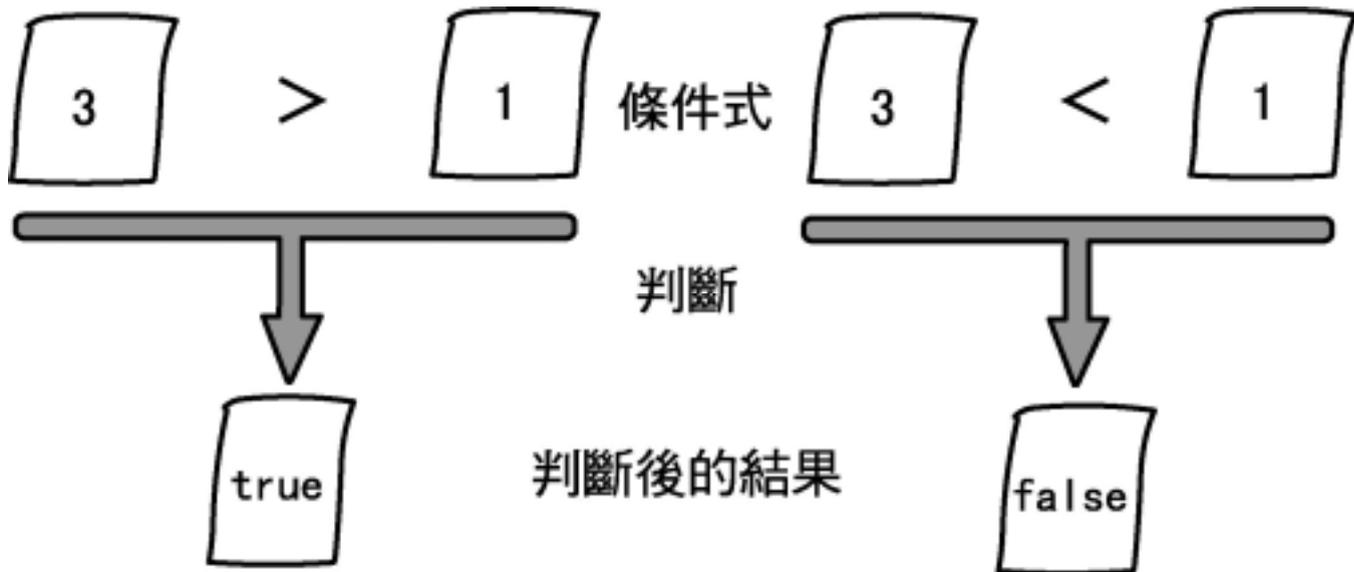
Java語言和其他優秀的程式語言一樣，也是有一套控制程式執行的方法，可以讓程式根據問題發生的實際狀況產生適當的反應，本章的內容就是要為您介紹這些優異的程式控制方法。

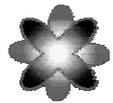
- ▶ 條件式
- ▶ 關係運算子
- ▶ 條件判斷句
- ▶ if 敘述
- ▶ if~else 敘述
- ▶ if~else if~else
- ▶ switch 敘述
- ▶ 邏輯運算子



# 條件式的表示法

經過條件判斷最後得到true或false的程式，在Java中被稱之為「條件式」。true代表條件式的結果為真、false代表條件式的結果為假。

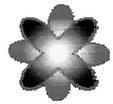




# 關係運算子

為了要比較大小關係而使用的大於(>)、小於(<)等符號，被稱為關係運算子(relational operator)。

關係運算子	條件式的結果要等於true，前提是...
==	右邊的值等於左邊的值
!=	右邊的值不等於左邊的值
>	左邊的值大於右邊的值
>=	左邊的值大於或等於右邊的值
<	左邊的值小於右邊的值
<=	左邊的值小於或等於右邊的值



# if 敘述

當括號內的條件式為真(true)時，才會執行if內部的其他程式；如果為假(false)，則不會進入執行。

語法

if (條件式)

其他敘述句 ;

條件式為真(true)時才會進入執行其他敘述句

範例

```
import java.io.*;
```

```
class Sample1
```

```
{
```

```
    public static void main(String args[]) throws IOException
```

```
{
```

```
        System.out.println("請輸入整數。");
```

接下頁程式



接前頁程式

```
BufferedReader br =  
    new BufferedReader(new  
InputStreamReader(System.in));
```

```
String str = br.readLine();  
int res = Integer.parseInt(str);
```

多 使用者從鍵盤輸入的  
值，再指定給變數res

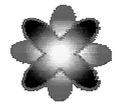
```
if(res == 1) {  
    System.out.println("輸入的是1。");
```

```
System.out.println("結束處理。");
```

多 如果輸入值等於1，則此處  
條件式結果為true

如 於是就會進來執行  
if內部的敘述

```
}
```



# if 敘述內部有二行以的程式敘述

## 句時

在if敘述中處理2行以上的的程式敘述句，必須在if敘述的內部用{ }將所有的程式敘述句框住。

語法

```
if (條件式) {  
    程式敘述句1 ;  
    程式敘述句2 ;  
  
    ...  
}
```

✿ if敘述內部有二行以的程式敘述句時...



範例

```
import java.io.*;
```

```
class Sample2
```

```
⋮
```

```
if(res == 1){
```

屆時如果輸入1的話...條件  
就會吻合 (結果為true)

```
    System.out.println("輸入1。");
```

```
    System.out.println("選擇1。");
```

會依序執行程式區塊  
內的所有程式碼

```
}
```

```
System.out.println("結束處理。");
```

```
}
```

```
}
```

# ✿ if ~ else 敘述

使用if else敘述時，條件式的結果不管是true還是false，都有各自不同的程式執行方向。內部的程式區塊也可以撰寫多行程式碼

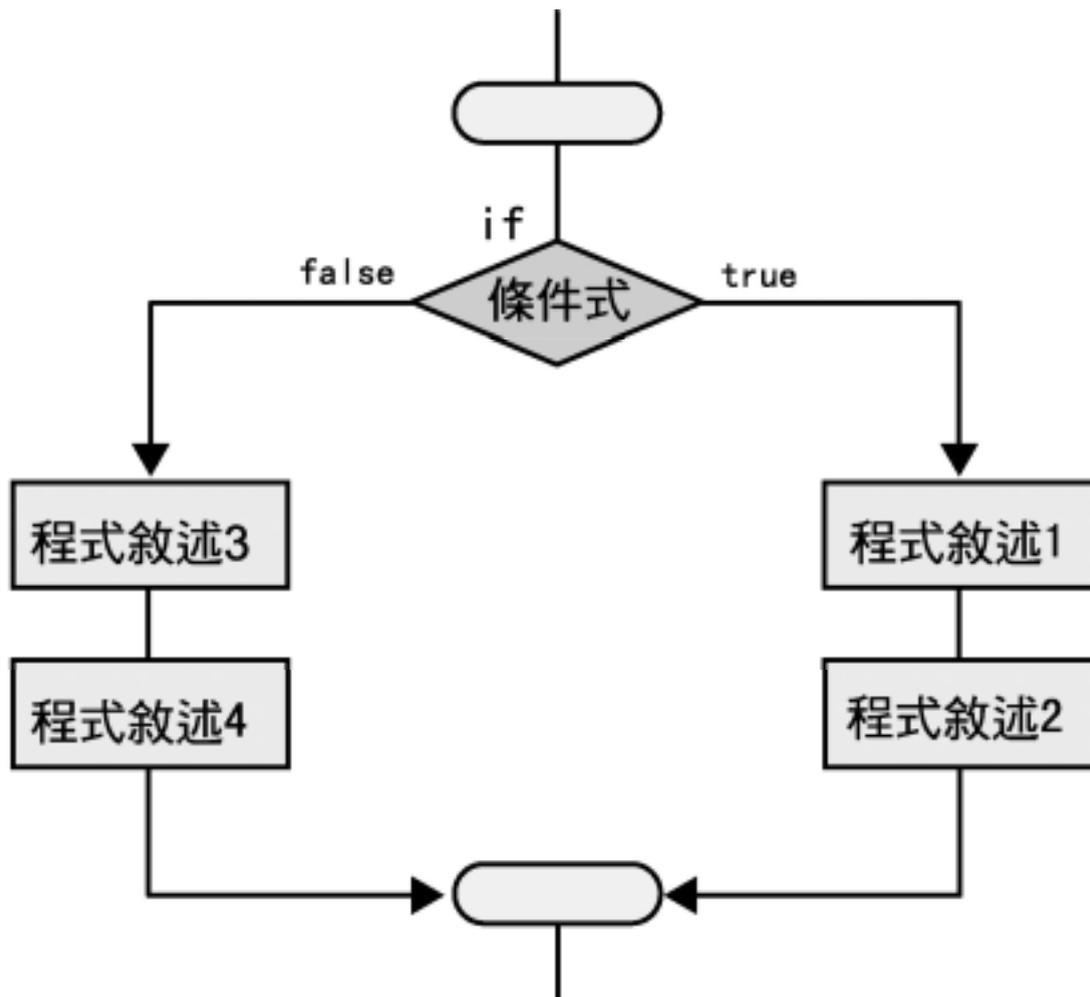
語法

```
if (條件式) {  
    程式敘述1 ;  
    程式敘述2 ;  
    ...  
else  
    程式敘述3 ;  
    程式敘述4 ;  
    ...  
}
```

上述的條件式為true時，會依序執行這個部份

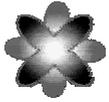
上述的條件式為false時，會依序執行這個部份

✿ if ~ else 敘述



# 判斷多個條件式 else敘述

if else if



語法

```
if (條件式1) {  
    程式敘述1 ;  
    程式敘述2 ;  
    ...  
}
```

條件式1為true時，會依序執行這個部份

```
else if (條件式2) {  
    程式敘述3 ;  
    程式敘述4 ;  
    ...  
}
```

條件式1為false並且條件式2為true時，會依序執行這個部份

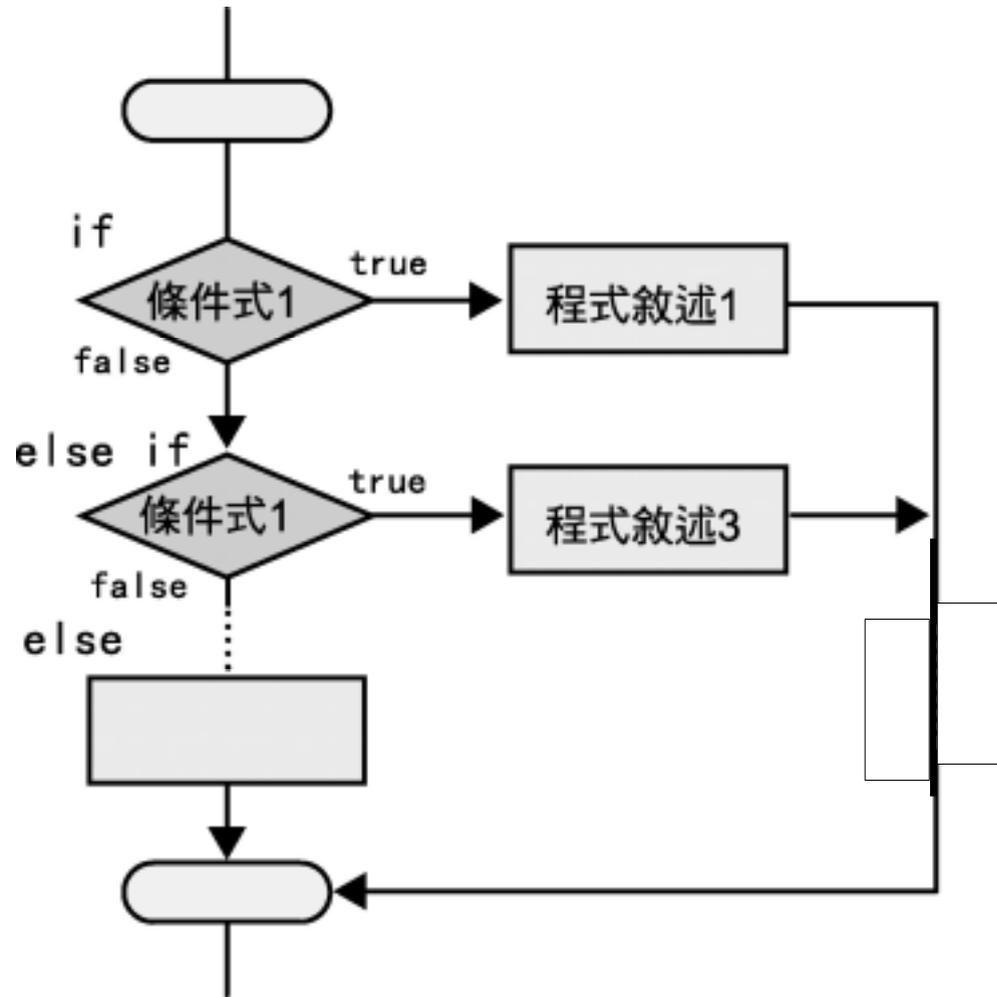
```
else if (條件式3) {  
    ...  
}
```

當條件式1、2均為false並且條件式3為true時，會執行這個部份

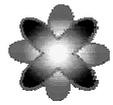
```
else {  
    ...  
}
```

上述所有條件式均為false時會執行這個部份

✿ if else if else敘述



if else if else敘述  
可以一次處理多個條件式



# switch 敘述

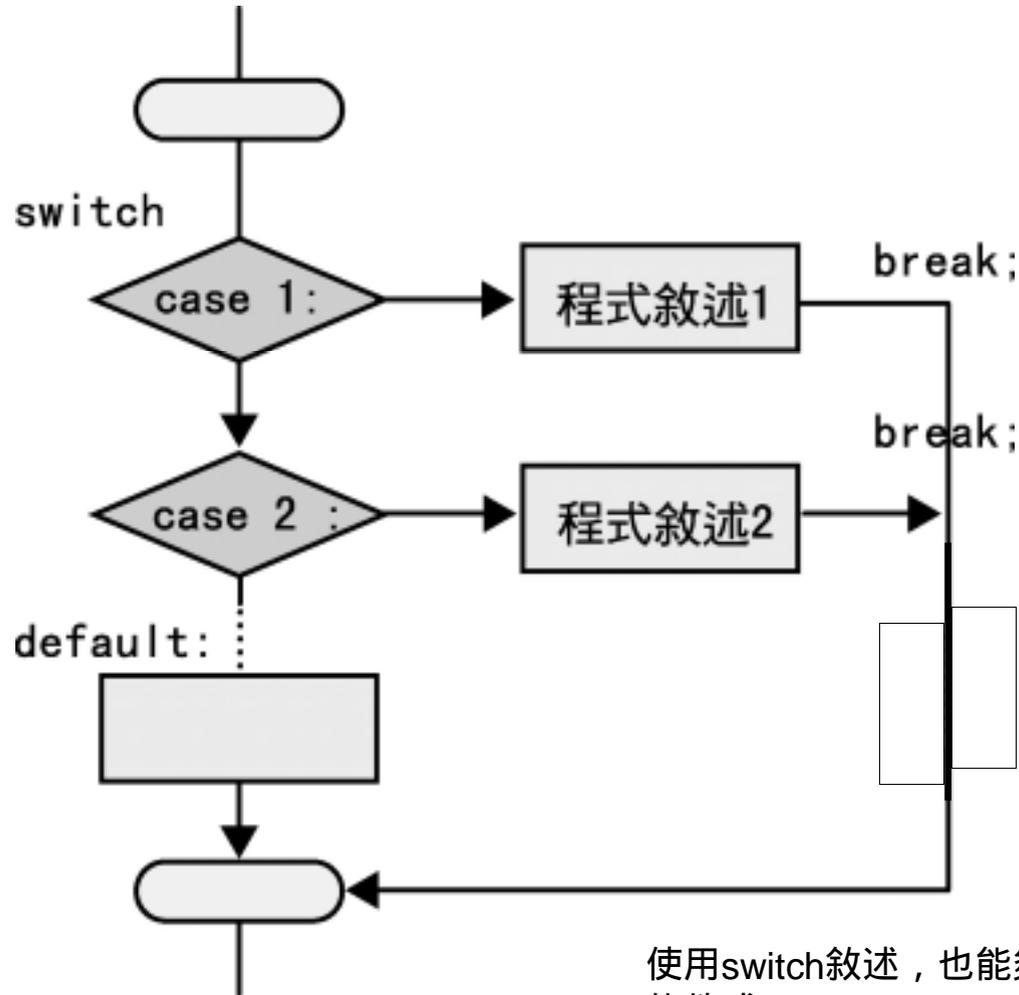
語法

```
switch (運算式) {  
  case 值1:  
    程式敘述1 ;  
    ...  
    break ;  
  case 值2:  
    程式敘述2 ;  
    ...  
    break ;  
  ...  
  default :  
    程式敘述d ;  
    ...  
    break ;  
}
```

運算式計算後的值如果等於值1會執行這一部份

運算式計算後的值如果等於值2會執行這一部份

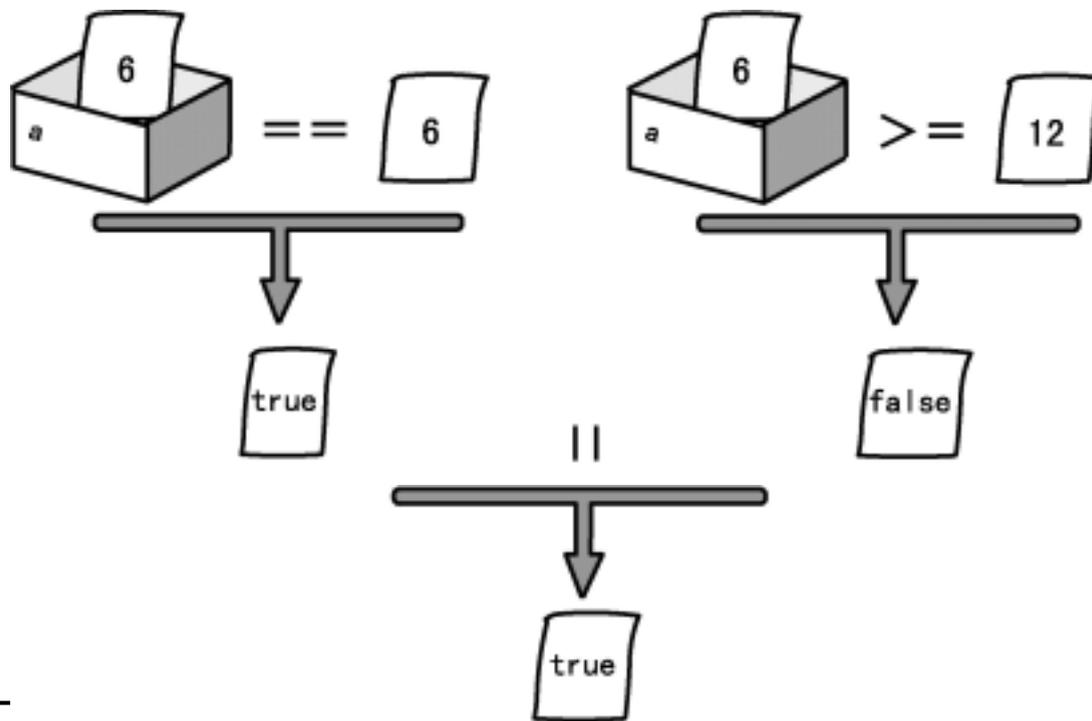
運算式計算後的值如果不等於值1和值2則執行這一部份



使用switch敘述，也能夠一次處理多個條件式

# ✿ 邏輯運算子

邏輯運算子的用意其實就是進行2個條件式的 true/false 在運算。



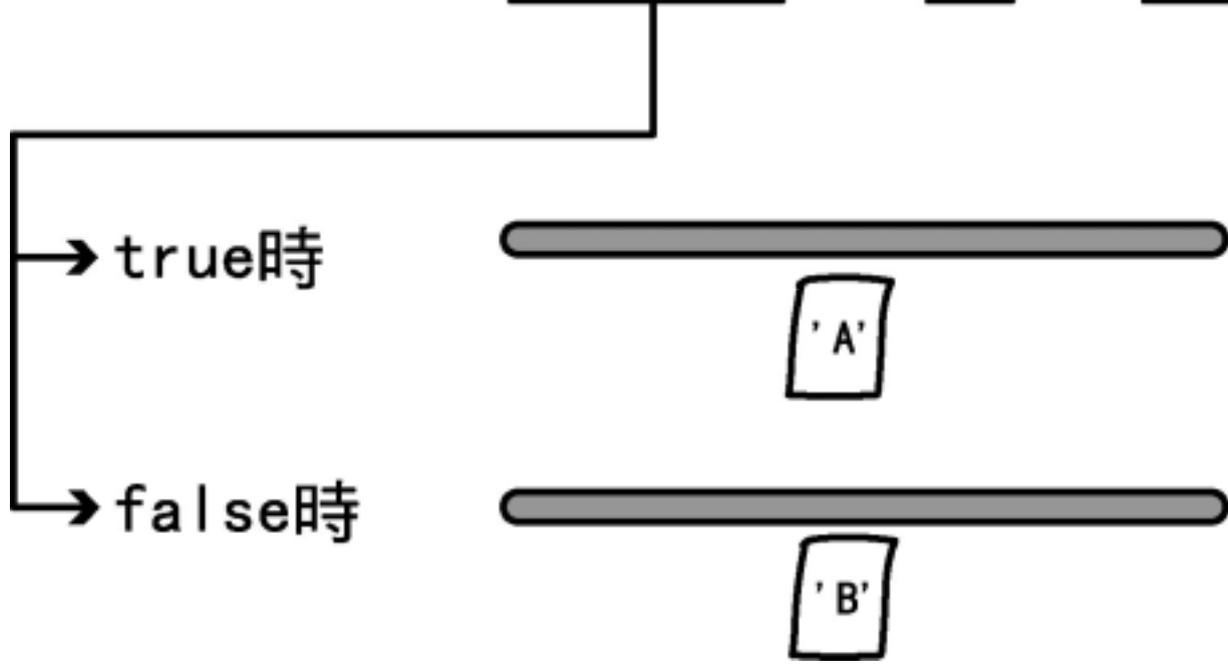
# 條件運算子

如果只是簡單的條件判斷，除了有if敘述可以使用之外，其他還有條件運算子（conditional operator）的「?:」指令可以使用

語法

條件式 ? true的時候為式子1 : false的時候為式子2

ans = (res==1) ? 'A' : 'B'



條件運算子會根據條件式當中設定的值  
(此處是res==1)判斷結果到底是true還是false，再決定應該執行哪一個程式碼