

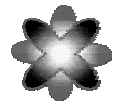
第六章 程式重複執行的方法

重點整理

除了第五章所學會的控制程式的執行流程之外，Java還有其他控制程式執行流程的方法-稱為「迴圈」(loop)。

。透過迴圈的控制，您可以讓同一個處理不斷重複進行，接下來就讓我們一起來認識迴圈吧！

- ▶ 關於迴圈
- ▶ for 敘述
- ▶ while 敘述
- ▶ do ~ while 敘述
- ▶ 巢狀迴圈
- ▶ break 指令
- ▶ continue 指令



for 敘述

Java中有專門用來處理這種重複事件的指令-稱為迴圈敘述(loop statement)，其中最主要的有for敘述、while敘述、do while敘述3種。

for迴圈和if敘述一樣，內部都可以撰寫多行程式敘述句。

語法

for (①起始值； ②判斷是否要重複執行的條件式； ③遞增或遞減運算)

程式敘述句 ;

內部的程式敘述句會重複執行

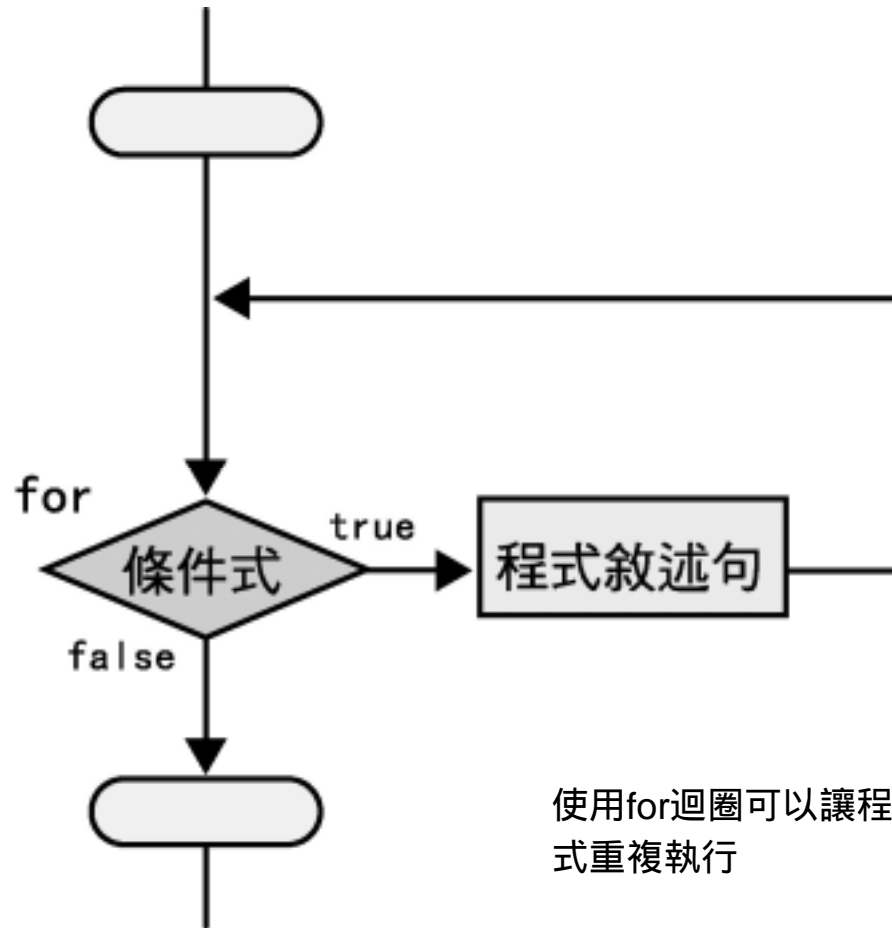
在for迴圈中利用大括號圍起的範圍稱為程式區塊(block)，如果for後面的條件式成立(true)的話，程式區塊內的程式敘述句1、程式敘述句2...等會重複執行。

語法

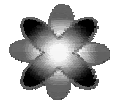
```

for ( ①起始值; ②判斷是否要重複執行的條件式;
      ③遞增或遞減運算) {
    程式敘述句1 ;
    程式敘述句2 ;
    ...
}
    
```

大括號內的程式會重複執行



使用for迴圈可以讓程式重複執行



while 敘述

另一個讓Java程式重複執行的迴圈敘述是while敘述（while statement，或稱為while迴圈）。

只要while迴圈後面的條件式為真(true)，那麼while迴圈下方程式區塊的內容就會不斷重複執行。

語法

```
while (條件式) {
```

當條件式為true時...

```
    程式敘述 ;
```

程式區塊內的程式碼就會重複執行

```
    ...
```

```
}
```

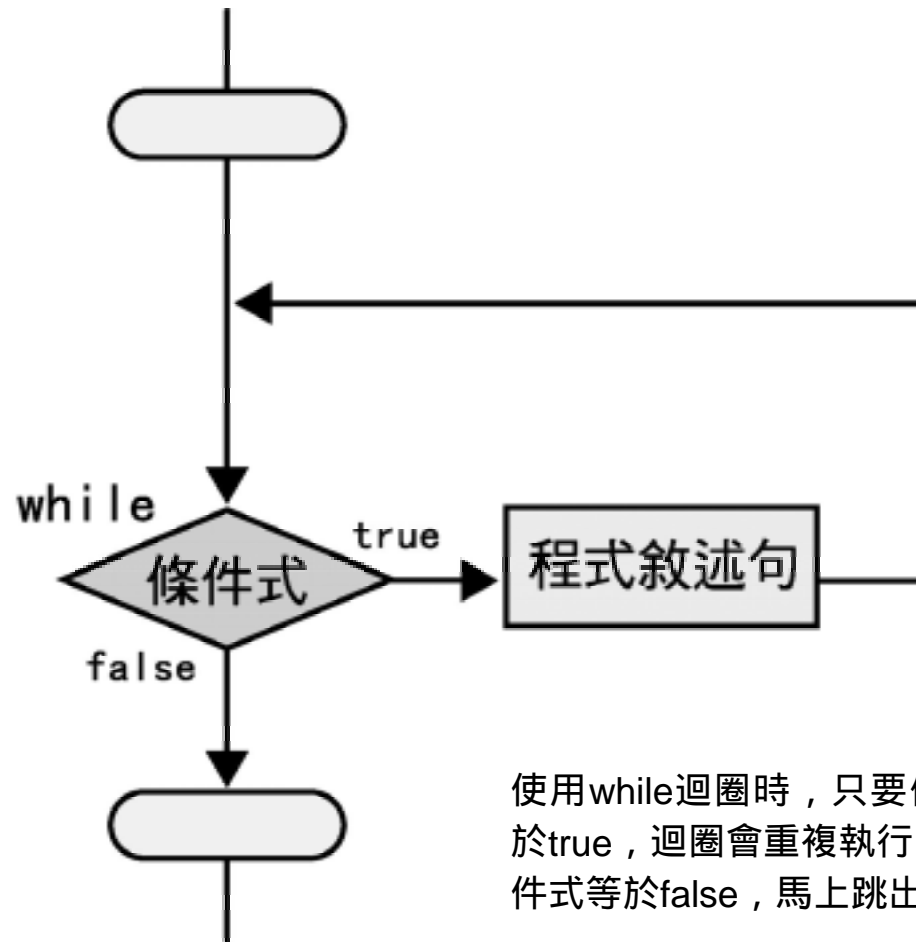
範例

```
class Sample5
{
    public static void main(String args[])
    {
        int i = 1;
        while(i <= 5){
            System.out.println("第" + i + "次的迴圈。");
            i++;
        }
        System.out.println("迴圈結束。");
    }
}
```

當條件式等於true時...

變數i會持續累加，直到i<=5
為false時就跳離迴圈

會在程式區塊內依序
重複執行



使用while迴圈時，只要條件式等於true，迴圈會重複執行；一旦條件式等於false，馬上跳出迴圈。

✿ do ~ while 敘述

只要while後面括號內的條件式為真(true)的話，就會不斷執行中間的程式區塊內容。

語法

```
do {
```

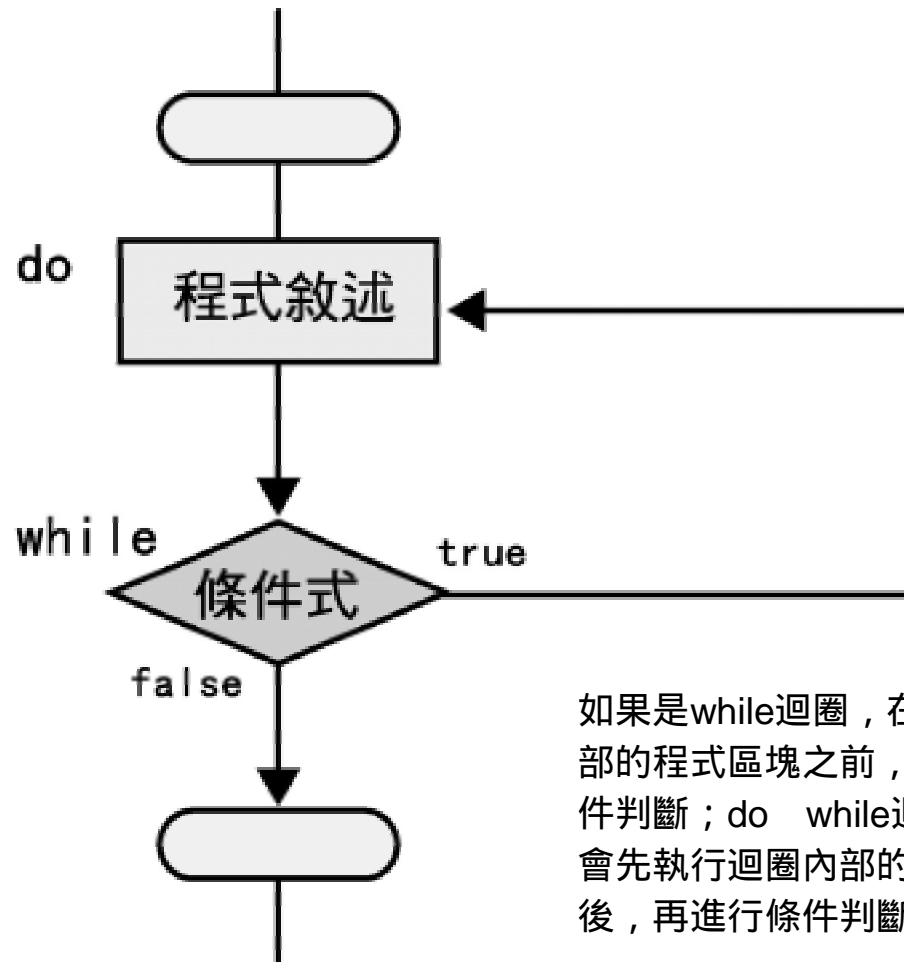
```
    程式敘述1 ;
```

會不斷執行這一段程式區塊的內容

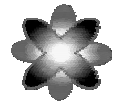
```
    ...
```

```
}while (條件式) ;
```

只要條件式為false時，馬上結束迴圈的執行



如果是while迴圈，在執行迴圈內部的程式區塊之前，會先進行條件判斷；do while迴圈的話，則會先執行迴圈內部的程式區塊之後，再進行條件判斷。



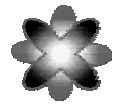
for敘述與巢狀迴圈

如果迴圈之內還有其他迴圈，就稱為「巢狀迴圈」。就像以下的範例當中，for迴圈內部還有另一個for迴圈就屬於巢狀迴圈的一種。

語法

```
for (式子1-1; 式子1-2; 式子1-3) {  
    ...  
    for (式子2-1; 式子2-2; 式子2-3) {  
        ...  
    }  
}
```

2個for敘述可以形成巢狀迴圈



改變程式執行的流程

設計Java程式時如果遇到必須變更執行的流程狀況，必須使用break和continue這2個指令。

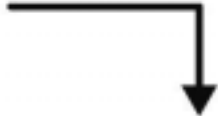
■ break指令

使用break指令，可強程式從目前執行中的迴圈跳離。

語法

```
break ;
```

```
for(int i=1; i<=10; i++){  
    if(i==res)  
        break;  
}
```



■ continue指令

continue指令可以「讓目前執行中的迴圈暫時停住不往下執行，而是回到迴圈繼續下一個執行」。

語法

```
continue;
```

```
for(int i=1; i<=10; i++){
```

```
    if(i==res)
```

```
        continue;
```

```
}
```

