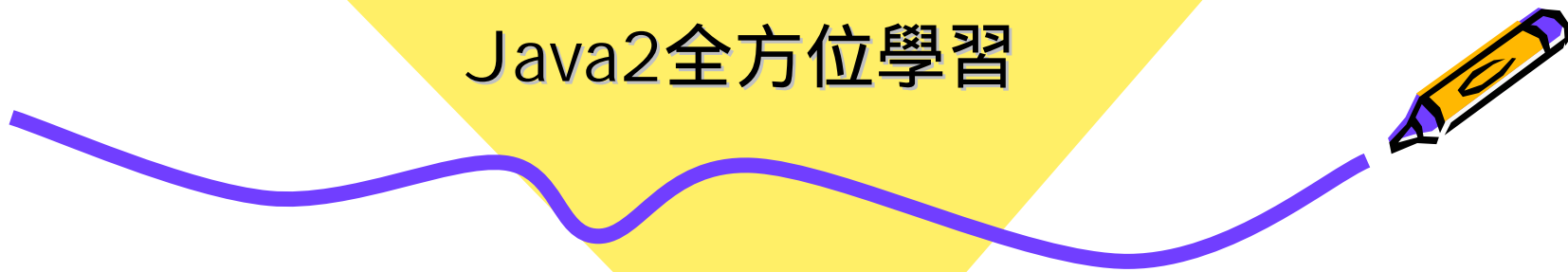


Java的陳述式

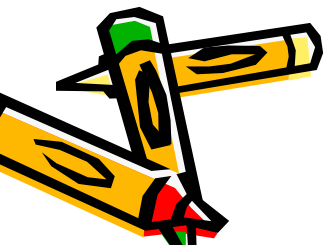
第六章

Java2全方位學習



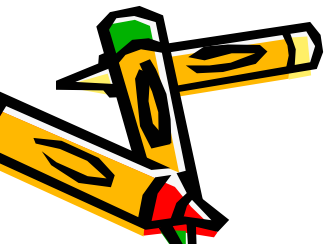
大綱

- 陳述式的組成與規則
- 一般陳述式的使用
- 宣告陳述式的使用
- 條件流程控制陳述式的使用
- 迴圈控制陳述式的使用



陳述式的組成與規則

- 陳述式通常以分號(;)作為結束
- 忽略中間的空格、定位符號和換行
 - 不限制只寫成一行

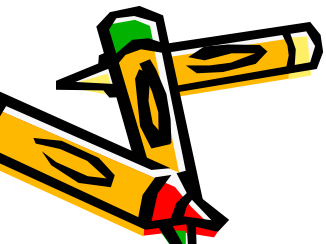


區塊陳述式

- 大括號{ }
- 做為某些陳述式的結束符號
- 將某段要一起執行的程式碼，寫在一個區塊中

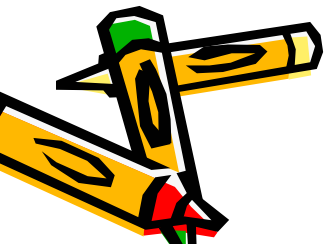
陳述式的種類

- 一般陳述式
- 宣告陳述式
- 條件流程控制陳述式
- 迴圈控憲陳述式



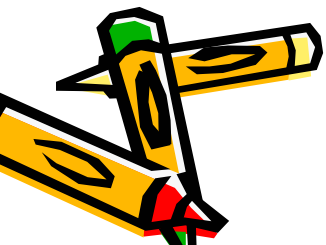
一般陳述式

- 註解
- 指定陳述式
- 物件的使用



註解

- 程式碼說明
- 多行註解
 - `/* ... */`
- 單行註解
 - `//`
- JavaDoc



指定陳述式

- `<變數> <指定運算子> <運算式>;`

物件的使用

- “.”運算子
- <物件/類別名稱>.<方法名稱>([參數])/<屬性名稱>[[<參數>S]];

宣告陳述式

- 基本資料型態宣告

- [**<修飾子>**S] **<資料型態名稱>** **<變數名稱>**[=**<初始值>**];
- 隱含宣告
 - 方法呼叫參數

- 物件的宣告

- [**<修飾子>**S] **<物件類別型態>** **<物件名稱>**[=**產生物件**];

宣告陳述式(二)

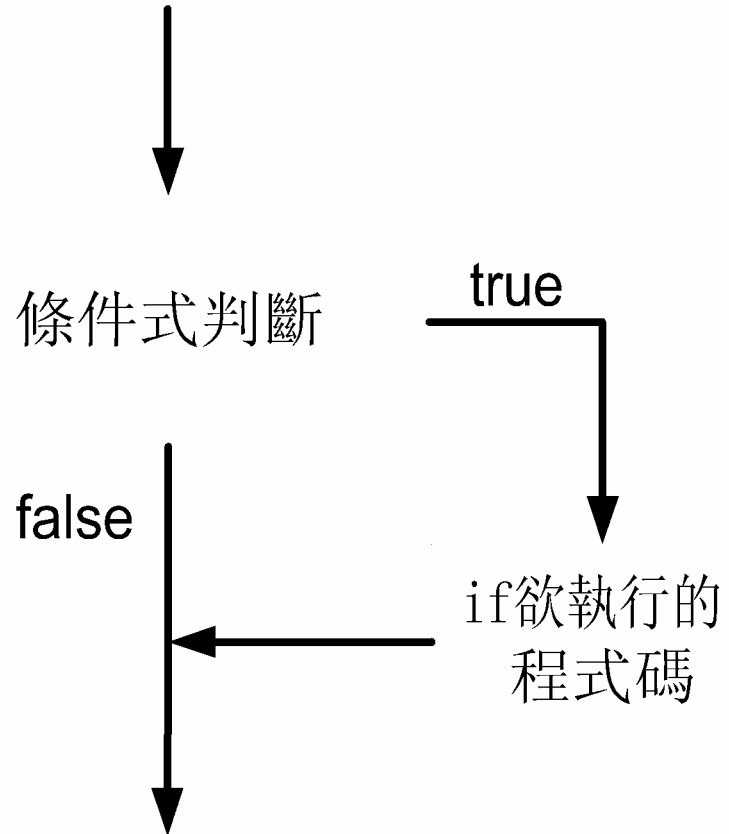
- 類別的宣告
 - [`<修飾子>s`] `class` `<類別名稱>` [`extends` `<類別名稱>`] [`implements` `<介面名稱>s`]
- 方法的宣告
 - [`<修飾子>s`] `<傳回值型態>` `<方法名稱>` (`[<參數>s]`)

條件控制陳述式

- 讓程式依不同的條件，執行不同的程式碼
- if-else
- switch

if陳述式

- if (<條件判斷>)
{ <程式碼> }

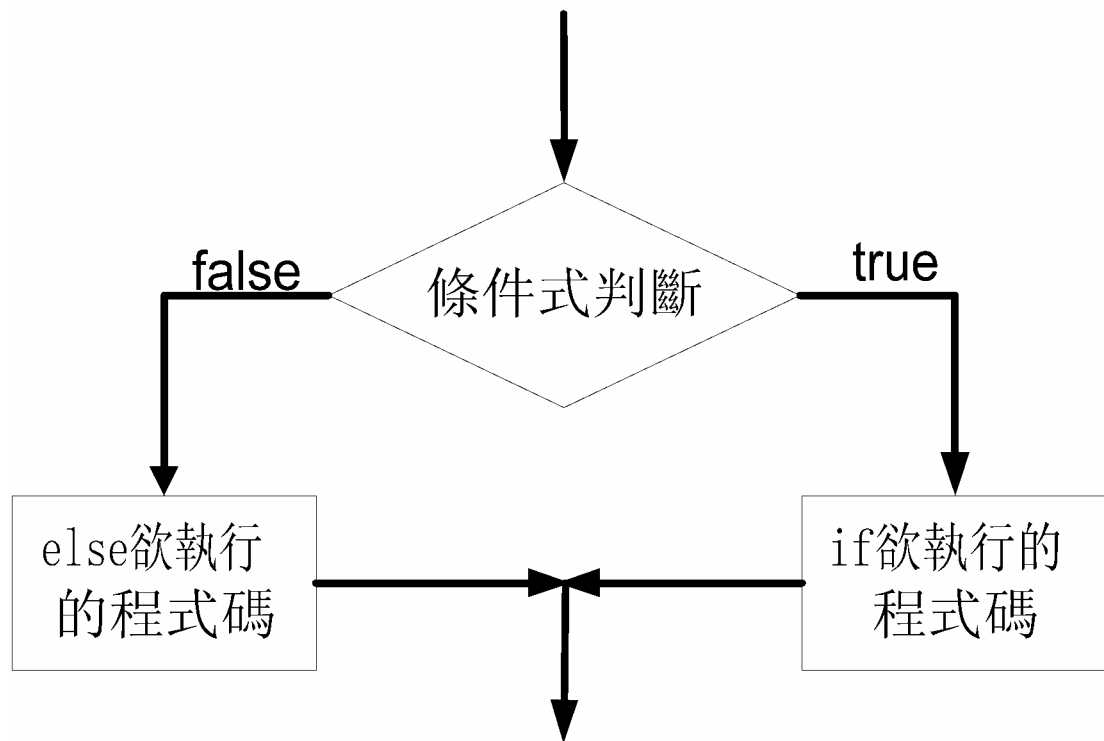


條件判斷

- 結果需為boolean型態
- 比較運算
- 邏輯運算

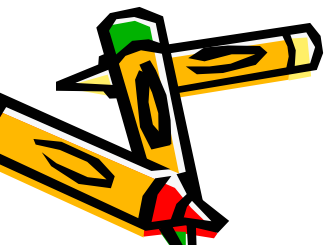
else

```
if (<條件判斷>
{ <程式碼> }
else
{ <程式碼> }
```



if-else進階使用

- 多個if-else
- 判斷的流程



?:運算子

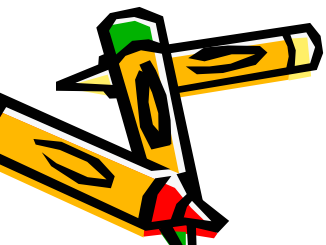
- 三元運算子
- $\langle \text{變數} \rangle = \langle \text{條件判斷式} \rangle ? \langle \text{條件判斷為 true 時之運算式} \rangle : \langle \text{條件判斷為 false 時之運算式} \rangle$

switch陳述式

```
switch (<byte、 short、 int或char變數>
{
    case <算術運算式或數值>:
        <程式碼>s;
        [break;]
    case ...
    ...
    [default:]
}
```

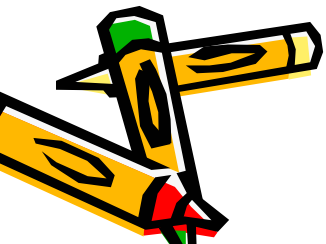
switch陳述式(二)

- break關鍵字的使用時機
- default標籤的使用時機



迴圈控制陳述式

- 重覆執行某段程式碼
- for迴圈
- while迴圈
- do-while迴圈

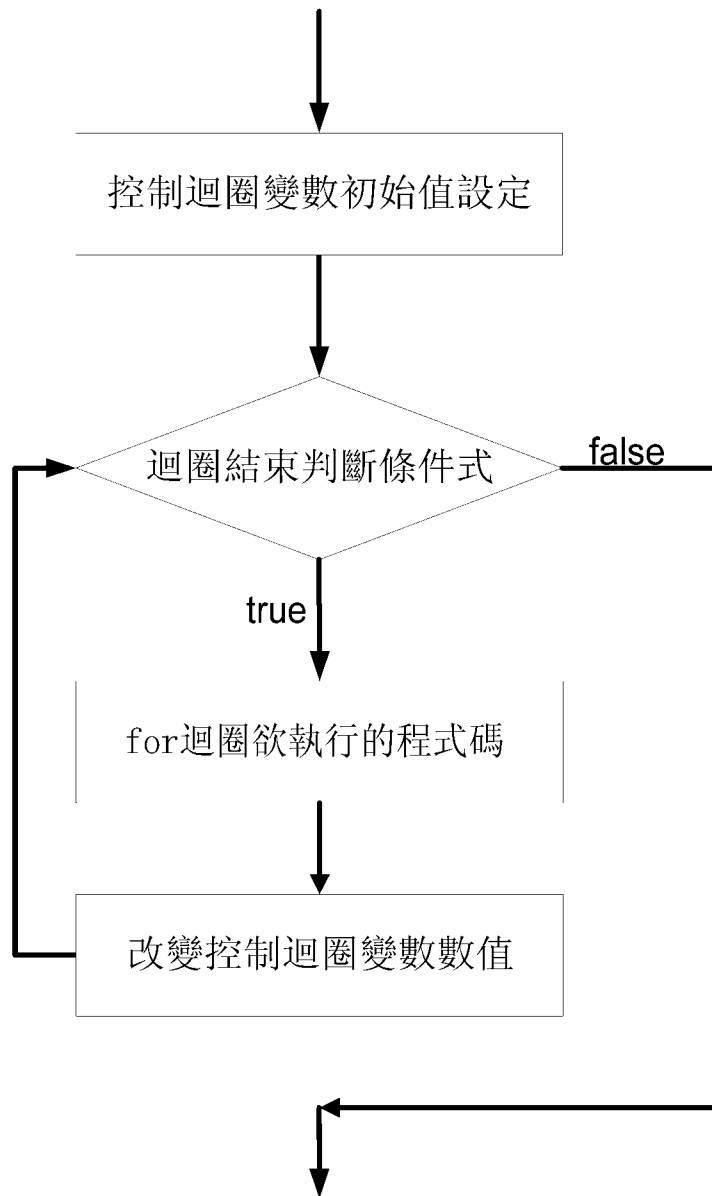


for迴圈

- 可預知執行次數

```
for (<控制迴圈變數初始值設定>;  
    <迴圈結束判斷條件式>;  
    <控制迴圈變數值改變方式>)  
{  
    <程式碼>s;  
}
```

for迴圈流程圖



for迴圈使用注意事項

- 不使用浮點數當控制變數
- Simple is good
- 無窮迴圈



while迴圈

- 未知執行次數

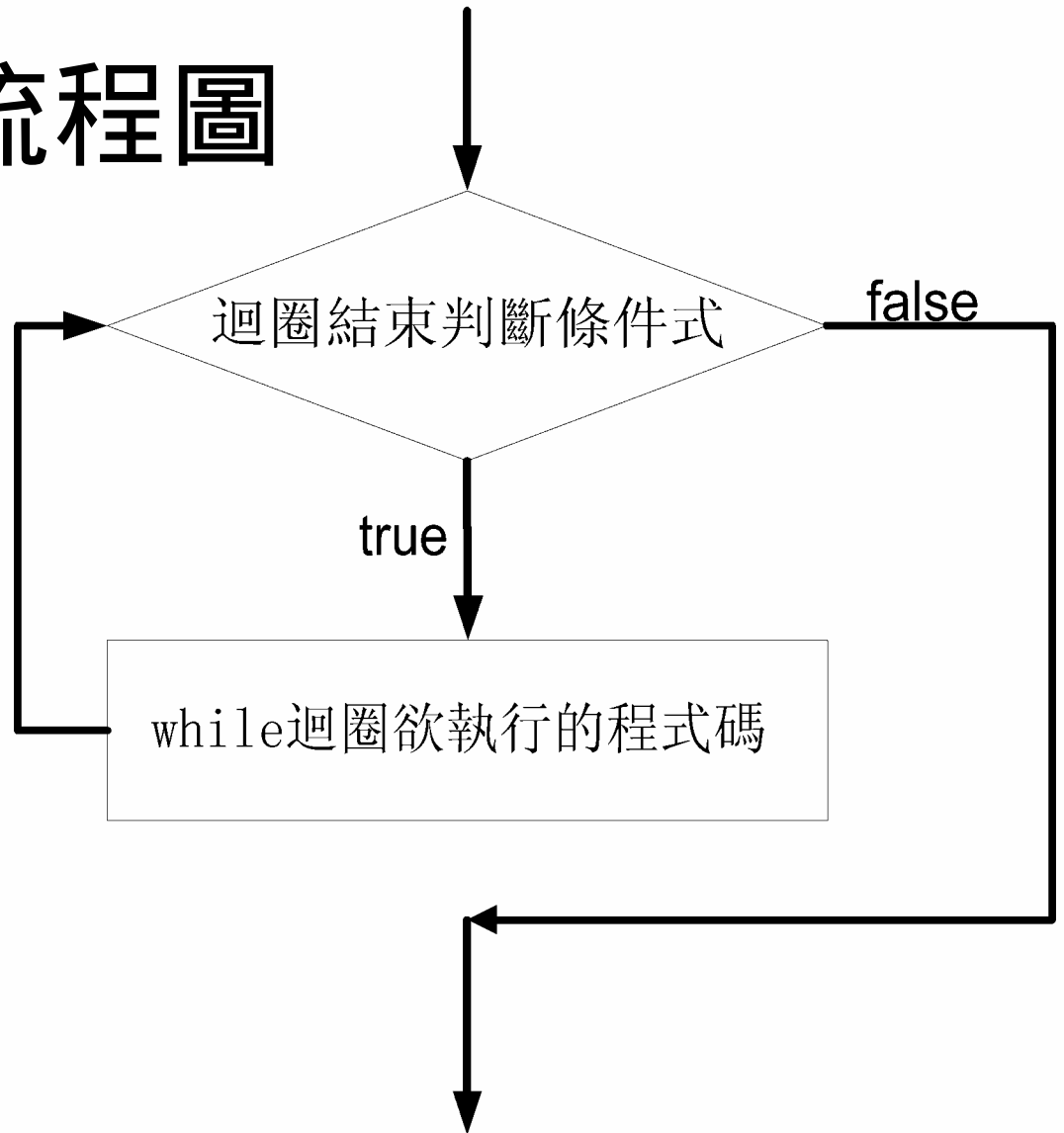
while (<迴圈結束判斷條件式>)

{

 <程式碼>S

}

while迴圈流程圖



do-while迴圈

- 至少執行一次

```
do
```

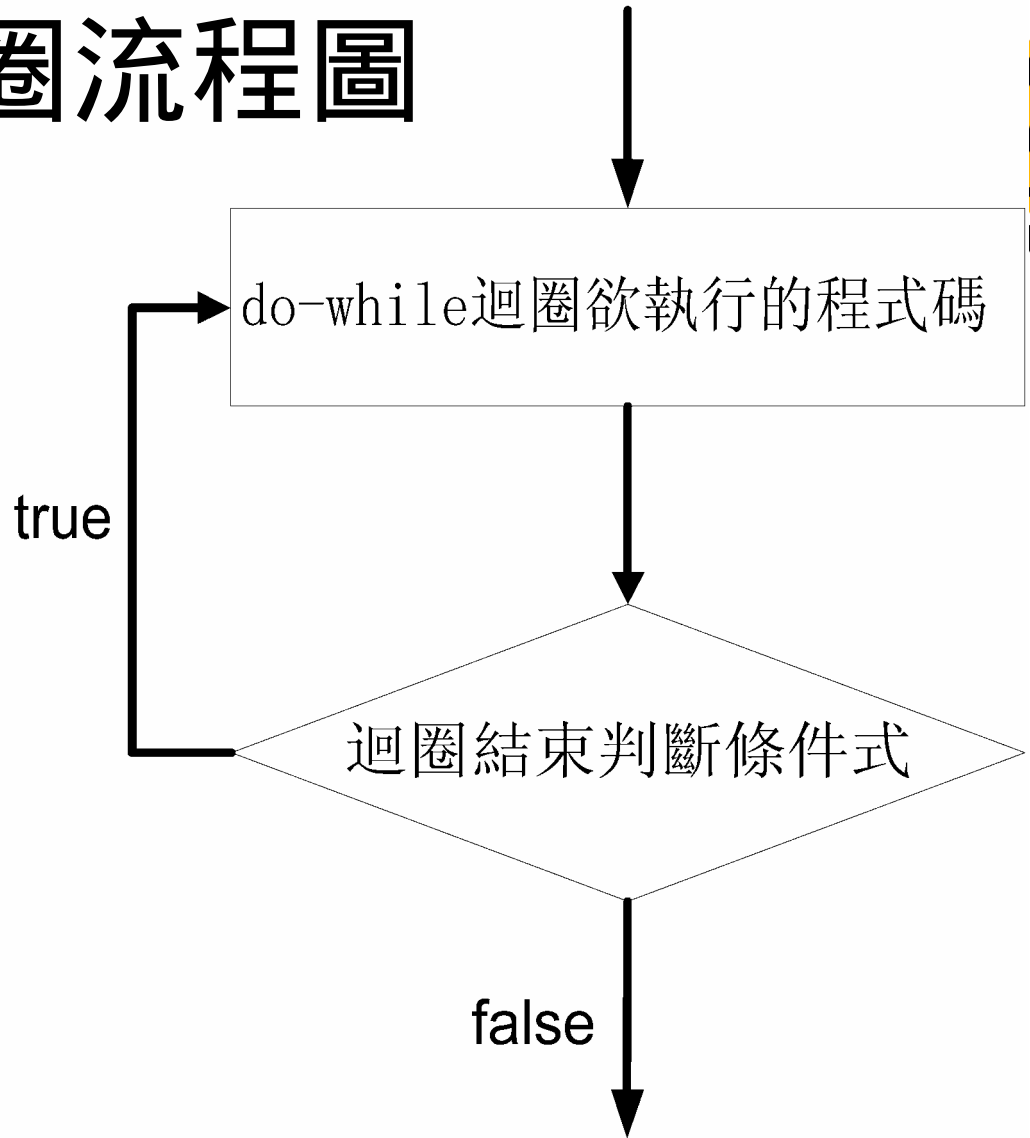
```
{
```

```
    <程式碼>S
```

```
} while (<迴圈結束判斷條件式>);
```

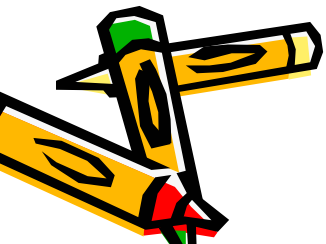


do-while迴圈流程圖



進階迴圈控制

- 巢狀迴圈
- break關鍵字
 - 跳離目前所在的迴圈
- continue關鍵字
 - 執行下一次迴圈



Q&A

