

- 使用 foreach 語法取得陣列元素

從 JDK 5.0 開始新增了 `foreach` 語法,又稱之為 `Enhanced for loop`,您可以將這個語法使用於陣列的循序存取上,語法如下：

```
1 for(資料型態 變數名稱 : 陣列名稱) {  
2     迴圈內容  
3 }
```

建構式

建構式(`Constructor`)是與類別名稱具有相同名稱的特殊方法,建構式會在您使用關鍵字 `new` 產生一個實例時執行,至於執行哪一個建構式,視您實例化時給定哪一種型態的引數而決定。沒有任何參數的建構式稱之為預設建構式 (`Default constructor`)。

重點提示：如果您定義了有參數的建構式,即使您不使用預設建構式,在設計上仍建議撰寫一個本體為空的(不包括任何陳述句)的預設建構式。

繼承

```
class 子類別 extends 父類別
```

不定長度引數(Variable-Length Argument)

只要在宣告參數時,於型態指定後加上...即可：讓不知道有多少引數的人可以使用。

This 語法

每一次物件呼叫共用的方法時,隱含的都會傳遞給方法一個 **this** 參考名稱,這個名稱會參考至物件本身 假設 **hello1** 呼叫 **hello()**方法,則 **this** 所參考的物件就是 **hello1** 所參考的物件,因而知道所使用的 **name** 成員就是 **hello1** 所參考物件的 **name** 成員;同樣的,如果 **hello2** 呼叫 **hello()** 方法,this 所參考的物件就是 **hello2** 所參考的物件,所以也可以正確的存取到 **hello2** 的 **name** 成員。

finalize()方法

在 JVM 回收物件之前,會執行物件的 **finalize()**方法,您可以在這個方法中撰寫一些物件被回收前的善後工作。

其他語法

回收資源：**System.gc();**

instanceof 運算子測試兩個物件是否為同一個類別的實例,如果不是就直接返回 **false**。