



第八章

物件導向程式設計

前言



物件導向程式設計(Object-Oriented Programming ;OOP)是目前最流行的程式設計模式，它更符合人類的思考模式。以前我們所學的C語言雖然提供有效率的功能，但並沒有提供完整物件導向的技術，例如C語言未提供封裝性與繼承性等特性。因此，在Java改善C語言的缺失，而提供完整物件導向的特性。



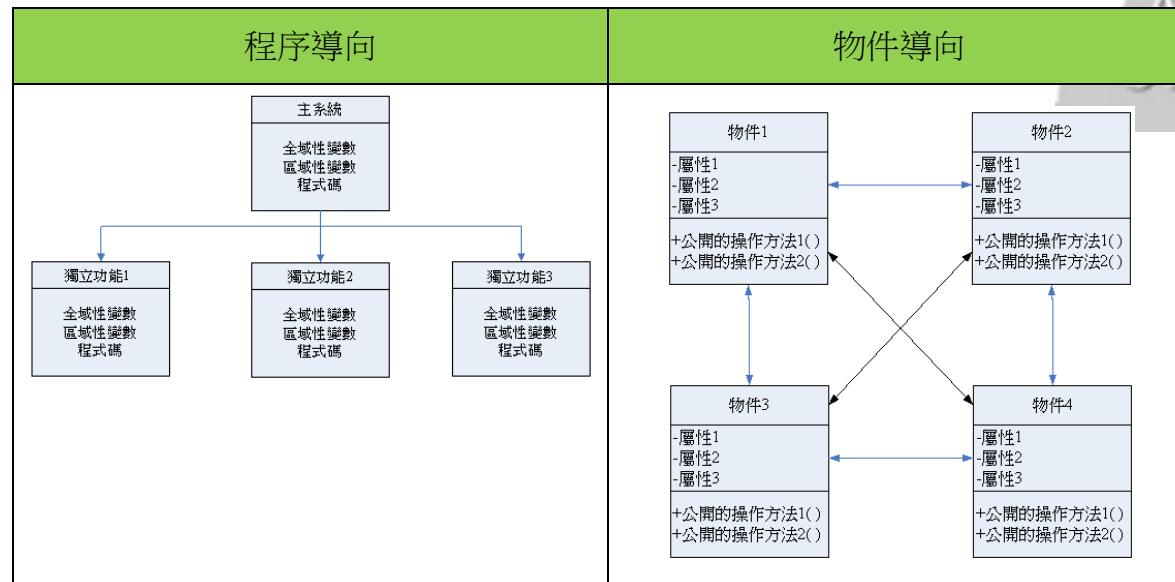
8-1 物件導向程式設計概論

在我們前面的章節中，都是利用傳統的「程序導向」設計模式，亦即所謂的結構化程式設計。但是，在人類的真實生活上都是以「物件導向」的方式來思考。並且目前大部份的程式語言都具有物件導向的功能，例如：Java, C++, C#等。

【程式撰寫的方法】

1. 程序導向：是屬於傳統的設計模式，亦即所謂的結構化程式設計方式。
2. 物件導向：它是利用「物件」方式來撰寫程式。

【示意圖】



【物件導向的特性】主要是由四個基礎概念所構成：

1. 封裝性(Encapsulation) ==> 本章介紹。
2. 繼承性(Inheritance)
3. 多載性(Polymorphism)
4. 抽象化(Abstraction)

【註】其中後三項，會在第九章詳細介紹。
真理大學 資訊工程學系

8-1.1 程序導向

【定義】是指以「功能」需求為主，程式只要能達到所要求的功能即可。

【作法】

在模組化設計過程中，是以「功能」來做為分解問題的依據，分解後的每一個小模組被視為一個獨立功能的子系統，然後再由下往上逐步將各功能模組合併，最後完成整個系統。

【示意圖】

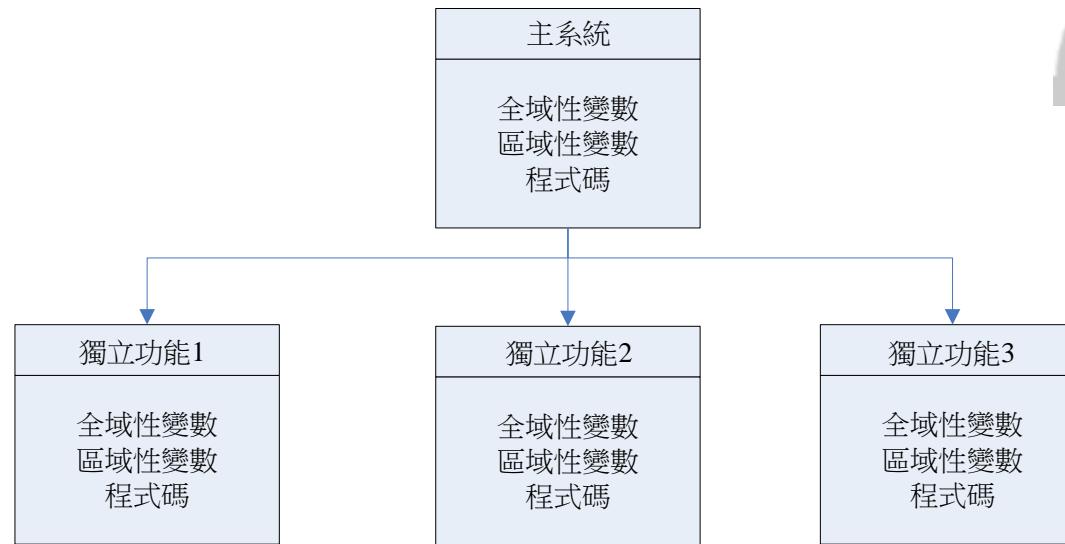


圖 8-1 程序導向示意圖

【缺點】缺少物件導向的四大特性。



8-1.2 物件導向

【定義】是指以「物件」為主。每一個物件模組就是一個定義好的類別。

【觀念】將一些常用的程式碼(屬性及方法)整合在起來，以便組成一個可被重複使用的程式碼零件。因此，我們可以稱這些程式碼零件，就叫做「類別」。如果該類別被實體化時，就會產生了「物件」。

【作法】

先規劃整個系統需要那些物件，再分析這些物件有什麼特性(屬性)、如何操作(方法)，以及物件與物件之間的關聯性，然後開始撰寫各個物件模組，最後再依照系統的需求，將各個封裝良好的物件模組由下向上架構出整個系統。

【示意圖】

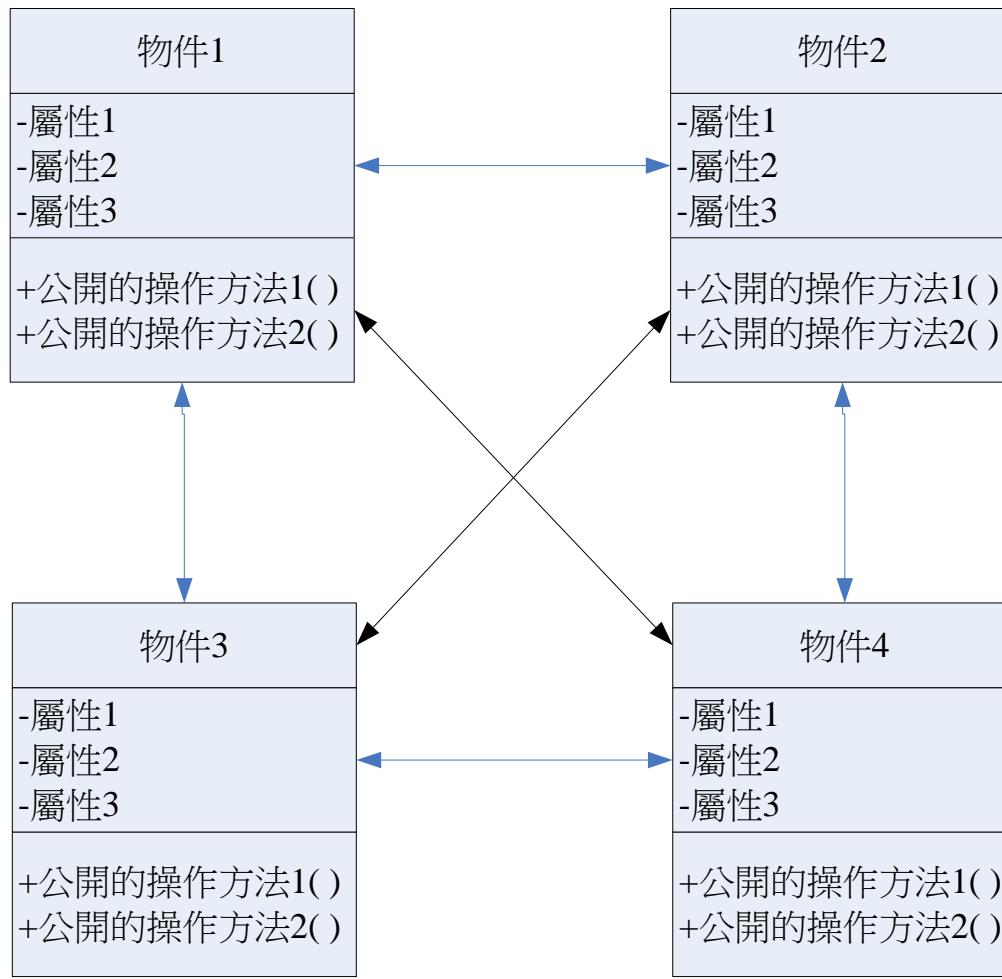


圖 8-2 物件導向示意圖



【設計步驟】

步驟一：宣告類別

透過 `class` 保留字來定義「類別名稱」。

```
修飾字 class 類別名稱
```

```
{
```

```
    程式區塊;
```

```
}
```

【說明】在每一支Java程式碼當中，至少會存在一個含以上的類別。

步驟二：宣告「資料成員」及定義「方法成員」

將相關的「資料成員(包含欄位或屬性)」和「方法成員(包含函式或行為)」包裝成一個「類別」。

```
修飾字 class 類別名稱
{
    資料成員(包含屬性或欄位) //宣告變數來記錄狀態值
    .....
    方法成員(包含函式或行為) //定義函式來完成某一功能
    .....
}
```

【說明】每一個「類別」當中，利用宣告變數來記錄狀態值及定義函式來完成某一功能。

步驟三：建立物件再來「存取」類別中的「資料成員中的屬性」及「呼叫」類別中的「方法成員」

ALETHEIA
UNIVERSITY

【語法 1】先宣告，再建立物件

類別名稱 物件變數名稱; //宣告某個物件變數為某種類別

物件變數名稱=new 類別名稱; //使用 new 建立物件實體

【語法 2】宣告物件的同時也建立物件

類別名稱 物件變數名稱=new 類別名稱();

【說明】利用已經宣告完成的「類別」，我就可以建立許多「物件」。

8-2 類別(Class)？

【定義】「類別」是指將具有相同性質的「物件」之集合。亦即將物件做「分門別類」。

【例如】人、狗、貓、車、書等。

【物件與類別的關係】

1. 「類別」視為「物件」的藍圖或模板。

類別	物件
人類	你、我、小明、大華

2. 「類別」只是用來描述「物件」的屬性、方法。



類別
人類的屬性：姓名、身高、體重...
人類的方法：走路、跑、跳...

【註】「人」這個類別只能定義物件的屬性與方法，卻無法執行方法。因為類別只是一種描述而已，實際上是無法存取資料及執行方法。

3. 「類別」本身並不是實際的「物件」，必須要透過「物件」來達成。

物件
小明的身高 170, 體重 70，每天上課走 1000 公尺
大華的身高 160, 體重 45，每天上課走 500 公尺

8-2.1 定義類別(Class)

Java是一種物件導向語言，在Eclipse整合開發環境中，必須要先「新增專案」名稱，其目的用來管理Java各項資源，但是，要讓Java可以真正執行時，則必須要在專案中，再「新增類別」名稱，並在類別中來撰寫程式碼。因此，每一支Java程式碼當中，至少會存在一個含以上的類別。

【定義】使用class關鍵字來定義類別。

【語法】

```
修飾字 class 類別名稱
{
    資料成員;
    方法成員();
}
```

【設計步驟】

步驟一：定義類別名稱

```
修飾字 class 類別名稱
```

```
{
```

```
    程式區塊;
```

```
}
```

【說明】

1. 「修飾字」有三種型式來宣告類別：

public	abstract	final
公共類別	抽象類別	最終類別

- (1) **public(公共類別)**：此類別沒有存取的限制。
- (2) **abstract(抽象類別)**：此類別無法被建立新的實體物件。
- (3) **final(最終類別)**：此類別無法被繼承。

- 
2. 「class」是用來定義「類別名稱」的保留字。
 3. 「類別名稱」的命名規則與變數相同。參考第三章。
 4. 「{...}」將定義類別的「資料成員及方法成員」放到左、右大括號中。

步驟二：宣告「資料成員」及定義「方法成員」

```
修飾字 class 類別名稱
{
    修飾字 資料成員(包含屬性或欄位) //宣告變數來記錄狀態值
    .....
    修飾字 方法成員(包含函式或行為) //定義函式來完成某一功能
    .....
}
```

【說明】

每一個「類別」當中，利用宣告變數來記錄狀態值的「資料成員」及定義函式來完成某一特定功能的「方法成員」包裝成一個「類別」。其中，「資料成員」及「方法成員」的「修飾字」有四種型式：

private	public	protected	default
私有成員	公共成員	保護成員	預設(不須宣告)

【範例】

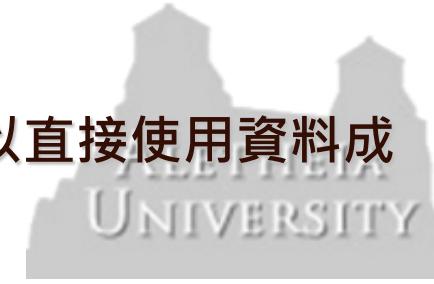
請建立一個求BMI值的「類別」，其資料成員有四個欄位及一個求BMI方法成員。

其中BMI值的判定比對如表所示：

BMI<20	20<=BMI<=25	BMI>25
多吃一點吧！	身材保持很好！	有空多運動！

【撰寫程式】

程式檔案名稱	ch8-2/src/Person.java
01	//定義與建立人類的類別
02	public class PersonClass {
03	//資料成員
04	public float Height; //宣告Height欄位
05	public float Weight; //宣告Weight欄位
06	public float BMI; //宣告BMW欄位
07	public String Result; //宣告Result欄位
08	//方法成員
09	public void BMI() //定義BMI方法
10	{
11	Height=Height/100;
12	BMI=Weight/(float)Math.pow(Height, 2);
13	if (BMI <20)
14	Result="多吃一點吧！";
15	else if (BMI>= 20 && BMI<=25)
16	Result="身材保持很好！";
17	else
18	Result="有空多運動！";
19	}
20	}



【註1】

在同一類別之中，如果要存取資料成員(變數)時，可以直接使用資料成員名稱(變數名稱)即可。

【註2】

如果主程式類別的main()方法內所定義的變數，不能直接呼叫資料成員。



8-2.2 定義物件(Object)

由於類別是一種「無法直接被存取」的形式類別，它必須要先建立對應的物件實體，而在Java中，使用「new關鍵字」來建立類別的物件實體。並且在程式碼中，我們可以利用既有的類別來產生數個物件。

【語法1】先宣告，再建立物件

在Java中，物件是一種類別型別的變數，在產生物件之前，要先宣告此物件所屬類別，進而建立物件。

類別名稱 物件變數名稱;	//宣告某個物件變數為某種類別
物件變數名稱=new 類別名稱;	//使用 new 建立物件實體



【範例】

```
PersonClass John;  
  
John = new PersonClass();
```

【語法2】宣告物件的同時也建立物件

```
類別名稱 物件變數名稱=new 類別名稱();
```

【範例】

```
PersonClass John = new PersonClass();
```

【撰寫程式】

程式檔案名稱	ch8-2/src/MainMethod.java
01	<pre>public class MainMethod {</pre>
02	<pre> public static void main(String[] args) {</pre>
03	<pre> //類別的宣告與建立物件</pre>
04	<pre> //建立John物件屬於PersonClass類別</pre>
05	<pre> PersonClass John = new PersonClass();</pre>
06	<pre> John.Height = 170; //指定John身高為「170」</pre>
07	<pre> John.Weight = 70; //指定John體重為「70」</pre>
08	<pre> John.BMI(); //John呼叫BMI方法</pre>
09	<pre> System.out.println("BMI=" + John.BMI);</pre>
10	<pre> System.out.println(John.Result);</pre>
11	<pre> }</pre>
12	<pre>}</pre>

【說明】

當我們要利用程式來存取類別時，首先必須要先建立對應的實體，也就是建立對應的物件。接下來，我們就可以指定或讀取該物件的資料成員，也可以呼叫物件的方法。

【作法】在「資料成員(或稱屬性)」或「方法成員(或稱方法)」名稱之前要加上物件名稱，並用點『 . 』隔離。



【範例】

(1) 指定物件的屬性 : **物件.屬性 = 屬性值**

例如：行號05中的John.Height = 170;

(2) 呼叫物件的方法 : **物件.方法(對應參數)**

例如：行號07中的John.BMI();

【執行結果】

BMI=24.221453

身材保持很好!

【隨堂練習】

承上一題利用類別來求BMI值，請再修改此程式，可以讓使用者隨時可以輸入「身高」及「體重」來求BMI值。

【撰寫程式】

程式檔案名稱	ch8-2EX1/src/MainMethod.java
	<pre>01 import java.util.Scanner; //載入Scanner類別套件 02 public class MainMethod { 03 public static void main(String[] args) { 04 //類別的宣告與建立物件 05 PersonClass John = new PersonClass(); //建立John物件屬於Person類別 06 Scanner inData=new Scanner(System.in); 07 System.out.print("請輸入身高："); 08 String strHeight=inData.next(); //輸入身高 09 System.out.print("請輸入體重："); 10 String strWeight=inData.next(); //輸入體重 11 John.Height=Integer.parseInt(strHeight); //指定John身高 12 John.Weight=Integer.parseInt(strWeight); //指定John體重 13 John.BMI(); //John呼叫BMI方法 14 System.out.println("BMI=" + John.BMI); 15 System.out.println(John.Result); 16 } 17 }</pre>

【執行結果】



請輸入身高：170

請輸入體重：70

BMI=24.221453

身材保持很好!