

Chapter 15

網路程式語言 — JAVA

(15-1) JAVA語言簡介

- 一、Java為Sun Microsystems發展出來，是一套新的個體導向程式語言，這新的語言很像一般的C、C++及Object C，如在流程控制for迴圈的語法，四種語言都是一樣的，但是它並不是以這些語言為基礎開發完成，也沒有希望一定要與這些語言相容。
- 二、Java語言能允許一段Java程式，透過網路傳送到另一部機器上執行，而且對於程式的來源機器，使用者不用對它做合法性的確認，也可以相信這段程式不會危害到我們的系統。為了達到這個目的，Java刪掉了一些一般C與C++都有的性質，如Java語言沒有指標（pointer）型態，如此可以避免利用指標存取不合法的記憶體位置。

(15-2) JAVA的特性

一、簡單（Simple）

Java強調它與現今流行的C++的語法很相似，但是更容易學習。Java保留C++的基本功能，而對某些可能造成問題的功能則以其他的方式來完成，或者直接刪掉此項功能。例如，Java沒有

Operator overloading, Header file, Preprocessor, Pointer, Structure, Union, Template, Implicit type conversion等。事實上，所列的功能中，有些是可以別的方式來完成，如Structure直接用class就可以了，而有些則沒有也不會有太大的傷害，如Operator overloading就是一例。

二、個體導向 (Object-Oriented)

(一)Java以類別 (class) 來組織程式，沒有一般的獨立函數 (function)，程式中只有類別。每一個類別會包含一堆成員方法 (member method)，而這些成員方法可用來定義個體的行為，一個類別的成員方法可以由繼承另一個類別而得到。在Java中類別的繼承架構都是以Object這類別為根，也就是，無論你有沒有繼承某個類別，Object類別一定會成為你類別的祖先 (ancestor)。

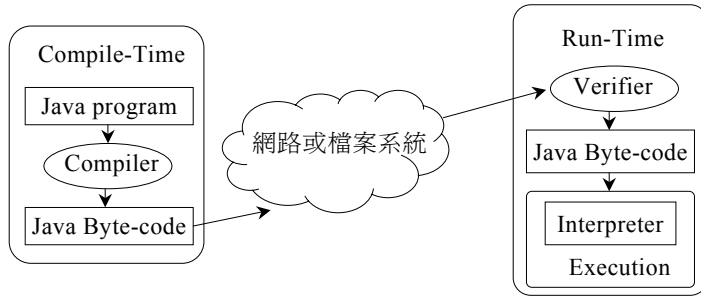
```
【例】 public class HelloTheWorld{  
        public static void main(String argv[ ]){  
            System.out.println("Hello world!!");  
        }  
    }
```

此程式的執行結果是輸出Hello world!!

(二)在Java中，只允許單一繼承 (single inheritance)，也就是一次只能繼承一個類別，並不允許多重繼承 (multiple inheritance)。

三、直譯式語言 (Interpreted Language)

執行Java程式時，需要有一個Java直譯器來加以執行，可是它並不像一般的直譯式語言，如Basic、TCL等，寫好程式之後就可以利用它們的直譯程式來執行，並馬上看到執行的結果。Java程式在執行之前，必須先編譯 (compile)，將Java程式轉變成另一種中間碼，稱為Byte-code，之後這Byte-code再拿給Java直譯程式執行。



四、跨平台

Java的程式經編譯後會產生一種稱為Byte-code之中間碼，而因為Byte-code是針對一虛擬機器（virtual machine）設計的，並非針對某種特定的機器，因此具有架構中立性（architecture neutral）。一個被編譯過的Java程式，可經由網路傳送至任何機器上，並利用Java直譯程式來執行，不需要再重新編譯，因此Java的程式可具有跨平台的特性。

五、多線執行（Multithreaded Execution）

在Java程式中允許有多個Thread一起執行。Thread（「線」）是指一個執行單元或一個指令執行流程，而Multithread則表示可以使一個程式在執行時，有多個執行單元。例如可以利用一「線」播放聲音，而另一「線」則執行動畫，使程式能有更大的彈性，而此種並行處理的功能，在一般的C++語言中並沒有提供。

六、垃圾收集（Garbage Collection）

在Java系統中，程式設計者不用去擔心記憶體的管理問題，因Java有一個內建的程式叫Garbage Collector，它會自動去分析記憶體，釋放不再使用的記憶體，這讓程式設計者可以專心於程式的設計，不必理會內部記憶體是圓的還是扁的。

七、強固性 (Robust)

對於系統而言，Java會小心的檢查每一筆記憶體的存取，保證它的合法性，不會讓Java程式去破壞電腦系統。在傳統的程式中，我們可以利用更改指標，而存取到不合法的位置；但由於在Java中並沒有提供指標，使得Java程式的存取範圍被有效地控制。另外，Java亦提供了例外處理 (exception handling) 的機制，使得程式本身對可能發生的錯誤能做有效的控制。

八、安全性 (Security)

Java並沒有提供指標，而且程式在執行之前是先將程式編譯成Byte-code，所以在執行時，可以利用Byte-code中所包含的資訊來查驗此程式的合法性（如程式是否違反了語言的規則，在執行過程中是否會存取到不被允許的資料等），以確保程式可以安全地執行，而不會危害到使用者的系統。這個查驗步驟通常可由瀏覽器來完成。

15-3 JAVA程式的分類及發展步驟

一、JAVA程式可分為兩類，一為Application，另一為Applet。

(一)Application是一個獨立 (standalone) 的Java程式，不需藉助具有Java能力的瀏覽器即可執行，且一個Application所受的限制也比Applet來得少，如Applet並不允許在客戶端做檔案的寫入動作（這主要是系統安全上的考量），但Application可以。我們能允許一個Application在自己的機器上執行，基本上就是假設這個程式並不會危害到我們的機器，所以允許它有較大的能力。

(二)Applet基本上與Application類似，但是並不能單獨執行，需要透過具有Java能力的瀏覽器來執行，如利用IE、Netscape等瀏覽器。通常一個Applet的能力是被限制的，這是因為一般的Applet都是來自網路上的其他機器，而且我們不能保證所傳過來的Applet都可以完全信任，所以必須對它的能力加以限制。