

# 第一章 認識 Java 與程式開發

## 1-1 何謂程式語言？

### 1-1-1 編寫劇本的工具

何謂『程式』( Program )？打個比方，就是程式設計師 ( Programmer ) 扮演著『編劇者』的角色，製作了一套『劇本』，讓電腦依照劇本內容演出，如果將『程式』改稱為『演出表』，也許會更貼切。

我們需要一套規則來製作『劇本』，設計師利用此規則製作劇本；電腦也按照相同的規則執行劇本內容，如此一來，電腦與設計師之間的期望值才不至於發生落差。如果將製作劇本的規格工具化，任何人利用該工具製作劇本，所製作出來的劇本大多數『電腦』也都能明瞭其意思，此工具即是『程式語言』( Program language )。另外，利用程式語言編寫劇本，必須符合該語言的『文法』( grammar )，『文法』即是程式的規則，如圖 1-1 所示。

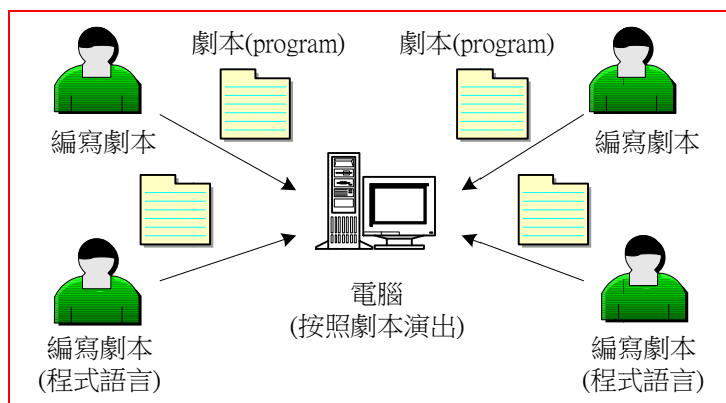


圖 1-1 程式語言與程式

由此可見，程式語言就如同『英文』一樣，我們利用正確的『單字』(如，關鍵字)，與正確的文法(如，程式規則)，將若干個單字組合成一個『句字』(如，敘述句)。也依照文法規範，將若干個句字組合成一篇文章(如，程式)。只要都合乎語言規則，任何人都可以依照文法規則閱讀該篇文章。同理可證，只要利用標準語言規則所編寫的程式，電腦應該都能執行才是。

到底電腦是一部機械裝置，它的處理能力是由人們所編寫軟體所規劃著。到目前為止，電腦還未具備思考及哲學推斷能力，只能依照程式軟體的步驟進行著。也就說，它僅能依照『劇本』演

出，自己並無法判斷劇本的對錯；假如劇本有疏忽或不完整的地方，導致『演出』亂無章法或不能滿足預先所期望的結果，這完全不可歸咎於電腦，而是編寫『劇本』的軟體工程師的疏忽。

既然電腦是一部機器，沒有理解『人』自然語言的能力，也無法讀取一般文字（如中文、英文）所編寫的劇本；吾人並不能直接指揮電腦從事某些工作，必須透過某一種工具編寫劇本才行。另一方面，真實環境裡存在著多種表演型態（如歌仔戲、布袋戲、西洋歌劇、現代舞...等），每一種演出都有其特殊的地方，也需要其專屬的工具來編寫劇本。程式設計也如同此道理，需要許多專屬工具來製作各式各樣的劇本。隨著時代變遷與科技的進步，程式語言也不斷的演進並滿足時勢所需，如目前常見的 C、Java、Fortran、VB 等等。每種程式語言都有其專門屬性，及特殊運用領域，無法直接比較之間的優劣點。雖然各種程式語言的運用領域不同，但編寫程式的技巧幾乎是相同的；也就是說，雖然有那麼多程式語言，利用他們將真實環境程式化的技巧及邏輯思維大致上是一樣的。

## 1-1-2 高階語言與機械語言

為了滿足各種環境需求，所以需要發展出許多『程式語言』來應對，這種想法，完全是從『人』的立場思考，純粹是方便『人』編寫程式，對『電腦』（演出者）來說，根本不是這麼一回事。因為到目前為止，電腦大多是由各種數位 IC 組成的機械，僅能辨識『0』與『1』，更清楚的說，電腦只能了解利用 0 與 1 的『機械語言』所編寫的程式，不管是早期的 Pascal、Fortran 到現在還有不少愛用者的 C、VB，以及本書介紹的 Java，如果沒有經過適當的轉換，電腦都看不懂。只是因為要人們利用機械語言編寫程式，非一般人所能及，所以需要使用 C、Java...這些比較靠近人類自然語言的**程式語言**編寫程式，然後利用稱之為『**編譯器**』（**Compiler**）的專屬軟體，將其翻譯成電腦所認識的『機械語言』，如圖 1-2 所示。

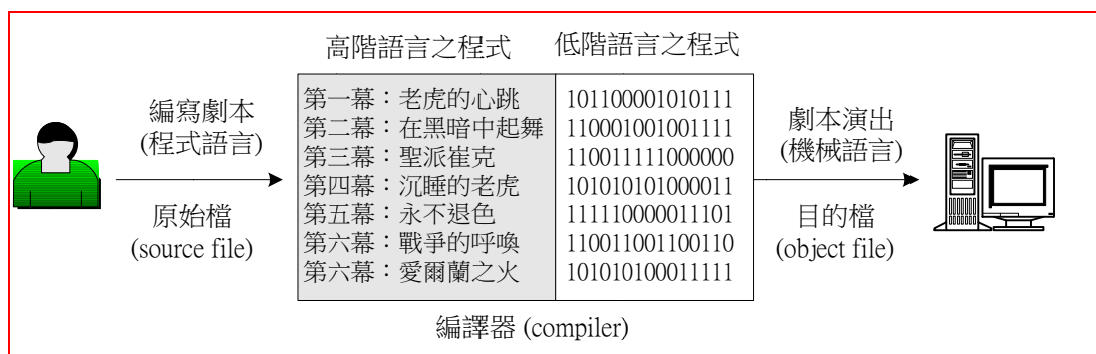


圖 1-2 程式語言的編譯器

我們將『人』所使用的語言工具，稱之為『高階語言』，所編寫出來程式檔案稱為『原始檔』；電腦所認識的語言稱為『低階語言』或『機械語言』，其所構成的檔案稱為『目的檔』或『機械碼』；而將原始檔轉換成目的檔的工具稱為『編譯器』。

不同系列的『中央處理機』(CPU)都有各自的『機械語言』，之間是很難相容的；譬如 Motorola 68000 系列與 Intel Pentium 系列之間的機械碼，則完全不能相容。也就是說，同樣語言編寫出來的程式，利用 Pentium 主機編譯後得到的機械碼，無法移植到 Alpha 主機上執行，因為兩者之間的 CPU 型態並不相同；即使是相同 CPU 型態的主機，若使用不同的作業系統，其機械碼不一定可以相容。如果要機械碼相容的話，不但必須使用相同系列的 CPU (如 Pentium 系列)，也要採用相同系列的作業系統才可以(如 Windows 系列)。由此可見，機械語言對系統的相依性非常高，同樣的程式語言，在不同的主機電腦上，都有各自的『編譯器』。

## 1-2 Java 語言的特性

### 1-2-1 程式移動性高

有個單純的想法，如果將程式語言(如 C 語言)標準化，而一般資訊系統也具備標準化的編譯器(如 C compiler)，則利用標準化所編寫的軟體套件，應該很容易的移植到其他系統才對。如圖 1-3 所示，程式設計師在 Windows 系統上，利用某種程式語言編寫出一套應用系統，如果將其移植到 Linux 或 Mac 系統上，只需使用該種程式語言的編譯器重新編譯，就可在該系統上執行，目前有許多廣為流傳的軟體套件(如 Apache 或 MySQL)就是使用這種概念。事實上並非總是如此容易，每套資訊系統都有其特殊的屬性，利用標準語言所編寫的程式，許多地方不能滿足其他系統環境所需，而需要特殊的更改。針對比較大的軟體套件，依照系統更改某些程式內容，還能滿足經濟效益；如果對某些較小的軟體程式，可能乾脆直接重寫還比較省事。

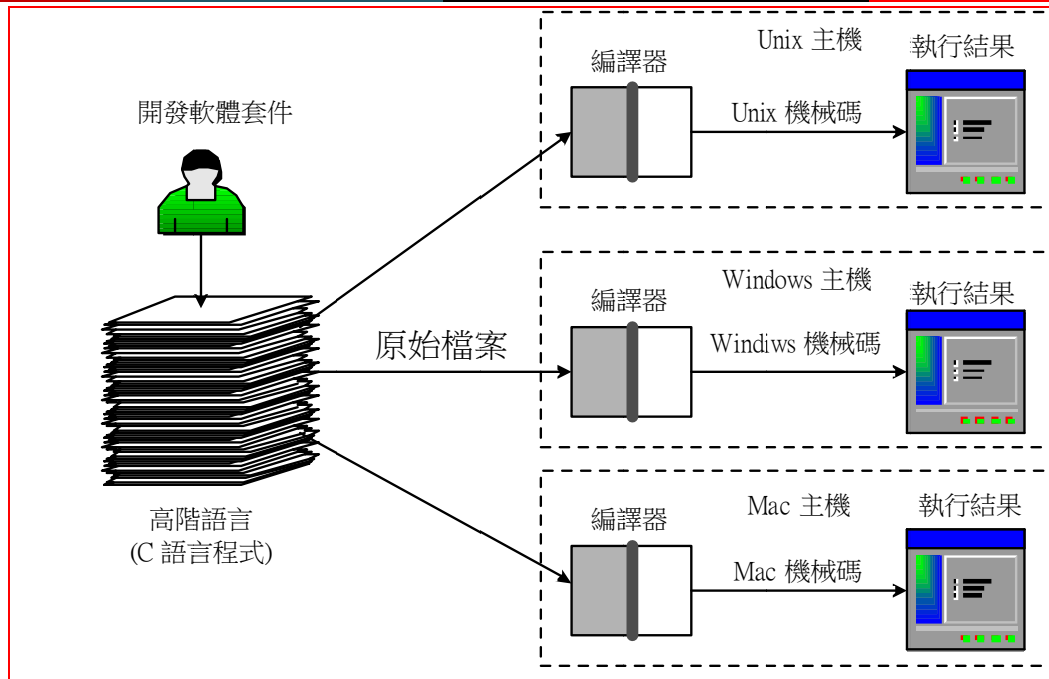


圖 1-3 傳統語言的軟體套件移植

會造成傳統軟體套件移植困難的主要癥結，是『編譯器』與『執行環境』之間的相依性；各種執行環境必須有特殊的編譯器配合，才能產生適合它執行的目的檔。Java 為了解決此癥結，乾脆將編譯環境與執行環境分開，成為兩套工作環境，如下：(如圖 1-4 所示)

- (1) **Java 軟體發展環境 (Java Software Development Key, JSDK)**：提供 Java 程式編譯與連結相關軟體套件，是一套完整的 Java 程式開發系統；主要工具是 Javac 編譯器，以及 Java.\* 套件。Java 程式經由 javac 編譯後，將會產生一個稱之為 Bytecode(中介碼)的檔案，並以 class 作為檔案型態。
- (2) **Java 虛擬機械(Java Virtual Machine, JVM)**：提供執行 Bytecode 的環境。主要是執行 java 編譯完成後的機械碼。我們可以在各種系統上發展與 JVM 相容的工作環境，如此便能執行其他系統所發展的 Java 軟體，如圖 1-4 所示。

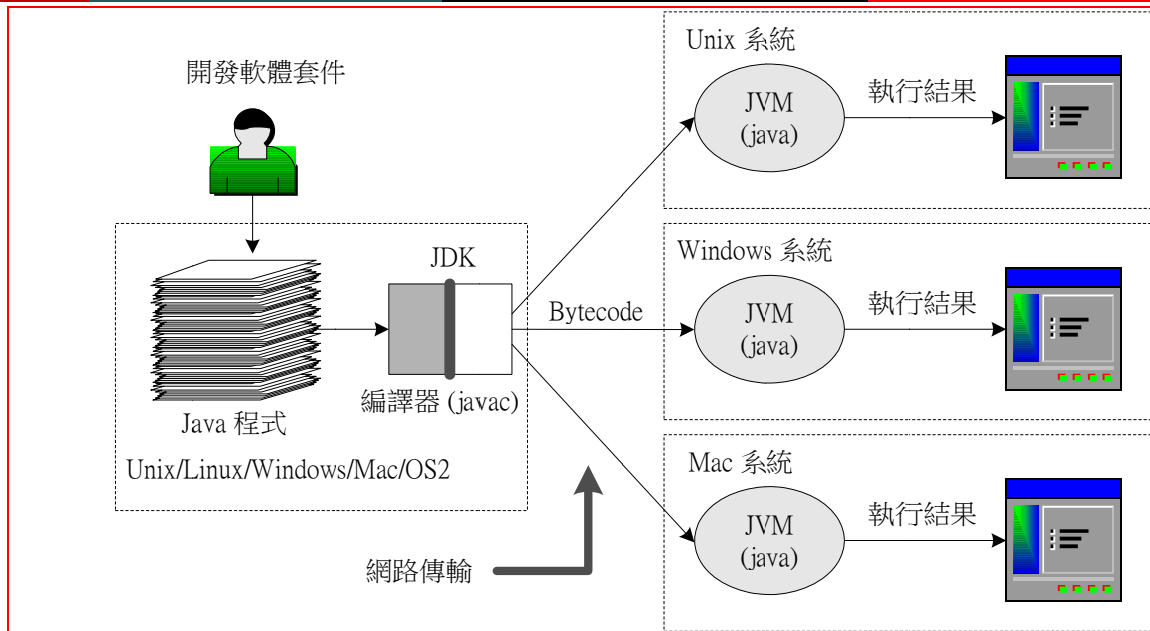


圖 1-4 Java 程式開發與執行環境

Java 的優勢在於，各種電腦系統都可使用『Java 虛擬機械』(JVM)執行由任何主機發展出來的 **Bytecode** 程式，漂亮地能解決機械碼的相依性問題。其實 JVM 的功能很像『全球資訊網』(WWW)的瀏覽器功能，不論哪種主機電腦都可安裝瀏覽器(如 IE)，顯示由各式各樣不同的電腦系統、應用軟體所製作的網頁。Java 的概念跟瀏覽器相似，將開發環境與執行環境分開，減低兩者之間的相依性，使得任何系統所開發的 Java 程式都可容易的移植到其他系統上執行。

## 1-2-2 豐富程式套件

執行環境與發展環境分開之後，開發軟體時則可不用考慮真實執行環境如何，如此一來，開發及連結導入軟體套件則更加容易。按照 Java 標準語法發展出的套件，將可容易的被導入其他 Java 軟體程式內使用。雖然各種傳統語言大多會提供一些較常用的『庫存函數』，但如果牽涉到較複雜的應用環境時(如視窗環境)，庫存函數大多與執行環境有關。因此，需針對系統(如 Windows 系統)的屬性，開發出專屬的庫存函數(如 Visual 函數)。Java 則不然，有關工作環境都由 JVM 環境處理，與開發環境無關，開發軟體套件時大多不用去理會真實執行環境；因此，許多人願意去開發軟體套件，也可容易的使用他人所發展的套件，如圖 1-5 所示。

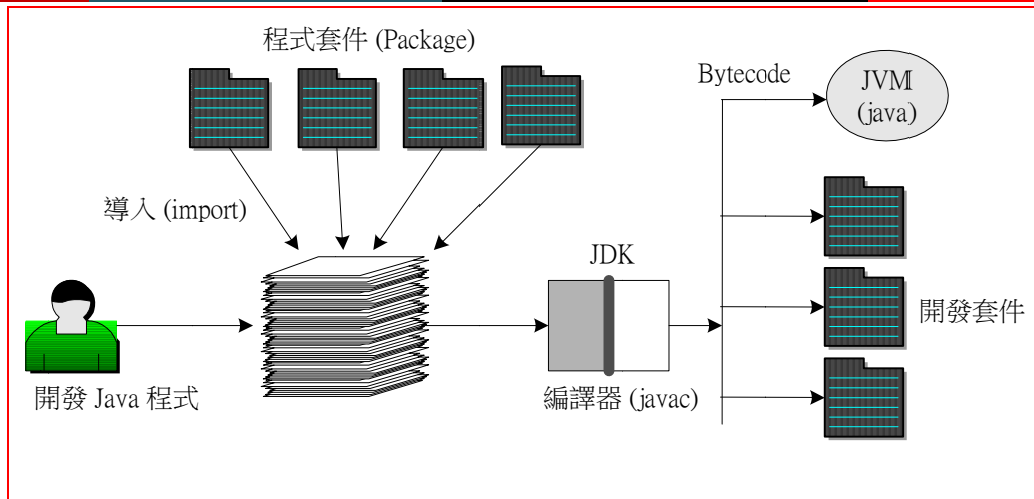


圖 1-5 Java 的軟體 IC 功能

既然開發出來的程式套件可應用於任何系統，我們就可以針對某些特殊功能製作成程式套件，以供他人使用。這種現象就如同硬體的 IC 晶片一般，每一只 IC 晶片都提供有某一特殊功能，將多個晶片組合在一起，就成一個專屬裝置設備；同樣的，組合多個 Java 程式套件，即可成為一個特定應用的軟體套件，這就是所謂『軟體 IC』(Software IC)的特性。

### 1-2-3 物件導向特性

早期談到 Java 語言大多離不開物件導向的功能，可說是 Java 最為特殊的特質。但以目前軟體領域的發展，幾乎所有發展軟體工具都具有物件導向功能，簡單說，物件導向不再是 Java 的專屬特性。利用物件導向發展工具所建立的軟體，大多由若干個類別(Class) 所構成，每一個類別實現某一個特殊功能，類別可能是由程式設計師自己編寫而成，也可能來自其他套件得到(如此則稱為軟體 IC)。所謂物件導向的特性就是敘述類別的特質如何，一般來講有三大特性，如下：

- **封裝性(Encapsulation)**：每一類別都具有封裝性，類別僅提供輸入/輸出介面，讓使用者呼叫引用，類別內的程式碼與變數與外界程式具有隔離性，亦是，無法直接呼叫或存取變數。具有此特性，類別才允許任意移值使用，不會受到其他類別程式限制，如此才能具有軟體 IC 的功能。
- **繼承性(Extends)**：類別可以再擴充產生新的類別，新類別除了具有原父類別功能外，還可增加其他功能，這就是繼承性。
- **多型性(Polymorphism)**：一般程式語言的函數呼叫最大限制就是資料型態，物件導向所建構



的類別可以隨需要自動改變相對應的資料型態，這就是多型性。譬如，我們建構一個加法器 (add(x, y))，當引數 x 與 y 是整數時，則回傳結果是整數；呼叫同樣類別，當 x 與 y 是浮點數時，則回傳是浮點數。如果其他傳統程式設計的話，則必須分開製作一個回傳整數，另一個回傳浮點數。

很可惜，本書並沒有敘述到如何處理物件導向的相關技巧，如有興趣的話，可繼續閱讀本書的延伸『Java 程式設計(二)含物件導向』。

## 1-3 建立 Java 程式開發環境

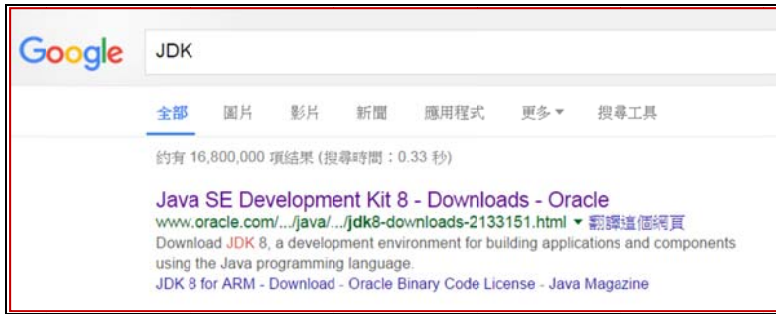
Java 是由 sun 公司發展出來，語言規範也大多由該公司所制定，是目前解決跨平台軟體設計的最佳工具。Java 將程式發展環境 (JDK, javac) 與執行環境 (JVM, java) 分開，任何系統上所發展出來的軟體程式，可容易的移植到任何系統上執行，只要系統上有 **JVM (Java Virtual Machine)** 即可。Sun 公司將 Java 軟體開發環境掛載於網路上 (java.sun.com)，完全免費讓任何人下載使用；更難得可貴的，也隨時發展許多功能強大的軟體套件 (如 Swing)，以供使用者發展應用系統。

由此可見，Sun 公司除了免費提供 Java 語言的學習工具外，更提供許多強大的發展工具。這方面與其他語言不同，學習其他程式語言大多需要購買該語言的發展工具 (如 Visual Basic)，或則僅能使用試用版。不然的話，採用一些免費的發展工具 (如 Dev C++)，大致上都沒有特殊套件或庫存函數可供學習，就算有也得另外花錢買。簡單的說，學習 Java 語言不但不須購買發展工具，而且它的發展工具功能非常的強。

### 1-3-1 Java JDK 套件安裝

Java 包含兩只軟體套件，一者為開發程式的 J2SDK；另一者為執行 Java 程式的 JVM。應注意的是，兩套軟體都必須安裝，其步驟如下：

- **步驟 1**：Google 上搜尋 JDK，則會搜尋到 Oracle 官方網站，如下：



進入官方網站後，點選 Windows x86 版本，如下：

Java SE Development Kit 8u71		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.71 MB	jdk-8u71-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.65 MB	jdk-8u71-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.75 MB	jdk-8u71-linux-i586.rpm
Linux x86	174.91 MB	jdk-8u71-linux-i586.tar.gz
Linux x64	152.74 MB	jdk-8u71-linux-x64.rpm
Linux x64	172.9 MB	jdk-8u71-linux-x64.tar.gz
Mac OS X	227.24 MB	jdk-8u71-macosx-x64.dmg
Solaris SPARC 64-bit	139.78 MB	jdk-8u71-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.05 MB	jdk-8u71-solaris-sparcv9.tar.gz
Solaris x64	139.98 MB	jdk-8u71-solaris-x64.tar.Z
Solaris x64	96.19 MB	jdk-8u71-solaris-x64.tar.gz
Windows x86	181.21 MB	jdk-8u71-windows-i586.exe
Windows x64	186.55 MB	jdk-8u71-windows-x64.exe

由下載目錄下，可觀察到此檔案：

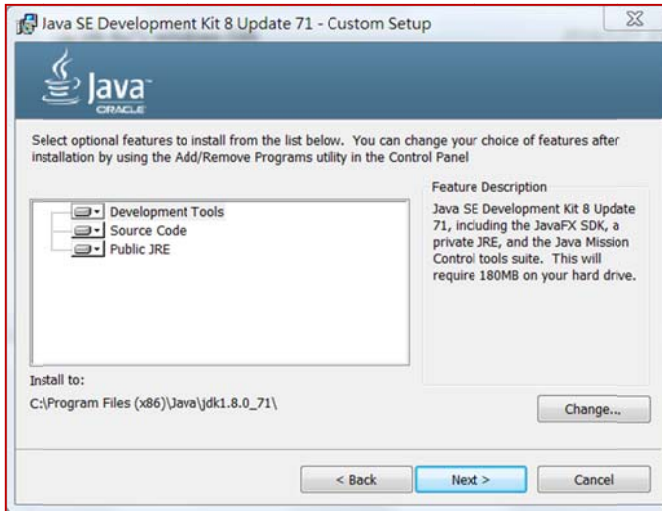
名稱	修改日期
jdk-8u71-windows-i586	2016/1/27 上午 09:39

## ■ 步驟 2: 安裝 Java SDK 軟體: J2SE Development Kit 5.0 及 J2SE Runtime Environment。

內定值的安裝路徑: c:\Program Files\java

1. 按兩下【jdk-8u71-windows-i586】檔案，則自動進入軟體安裝程序，並出現歡迎安裝的精靈畫面。
2. 首先出現 JDK 授權同意書的對話視窗，並選擇『I accept the terms in the license agreement』，再點選【Next】。
3. 出現選擇套件視窗，與安裝路徑（內定值為 c:\Program Files(86)\Java\jdk1.8.0\_71\；也可改變到其他目錄）；選擇「Development Tools」與【Next】。





4. 再出現選擇軟體套件，選擇「J2SE Runtime Environment」的目錄，預設值是 c:\program Files(x86)\java\jre1.8.0\_71(不用變更)，直接按【下一步】。



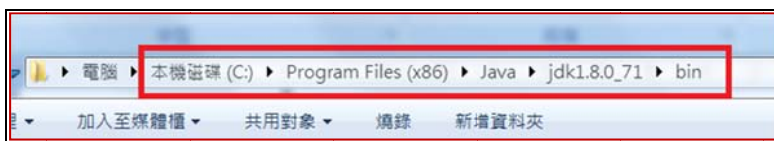
進入安裝狀況：



5. 安裝完成，按入【Finish】。

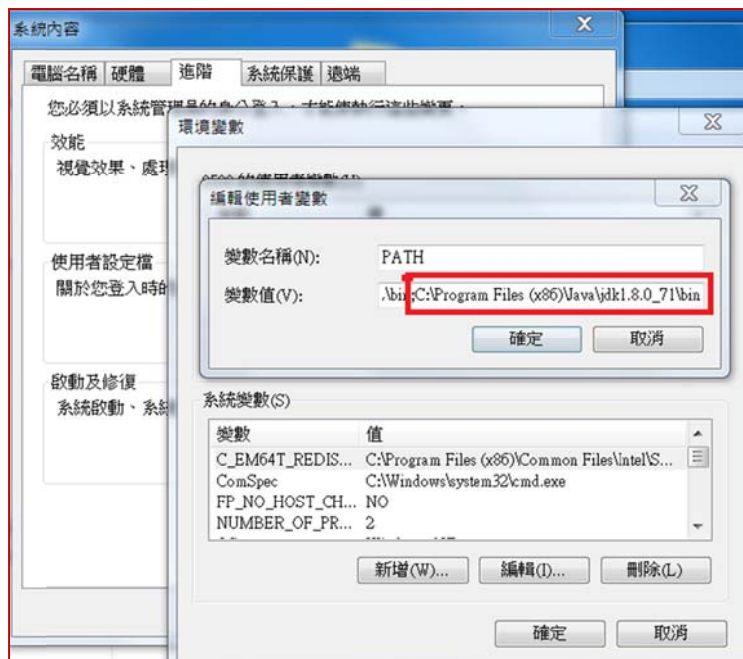


- **步驟 3**：利用『檔案總管』觀察 jdk 安裝在甚麼地方，將此位置複製下來，如下：



- **步驟 4**：設定命令路徑：`C:\Program Files(86)\java\jdk1.8.0_71\bin`

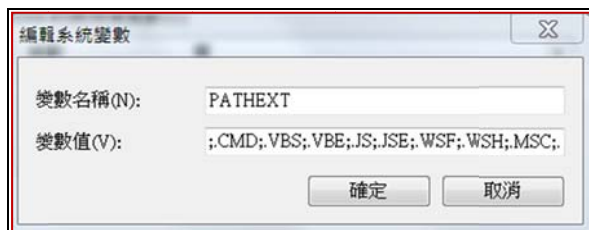
1. 選擇「開始」→「控制台」→「系統」→「進階設定」→「環境變數」→「Path」→「編輯」；視窗如下：



2. 將「變數值」的內容最後面增加【`;C:\Program Files(86)\java\jdk1.8.0_71\bin`】(可利用複製 - 貼上)。如果安裝於其他位址也相同方法設定，譬如，安裝於 `C:\j2sdk1.4.2`，則加入【`;C:\j2sdk1.4.2\bin`】。
3. 再由系統變數上選擇『PATHEXT』系統變數，按入編輯，則出現對話框如下：



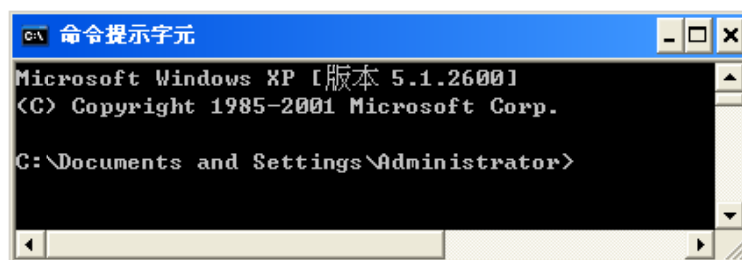
4. 將『變數值』最後加入『;.』，表示在目前目錄(.)下即可找到 class 類別程式。



## 1-3-2 測試 JDK 執行環境

### (A) 步驟 1：啟動『命令提示字元』

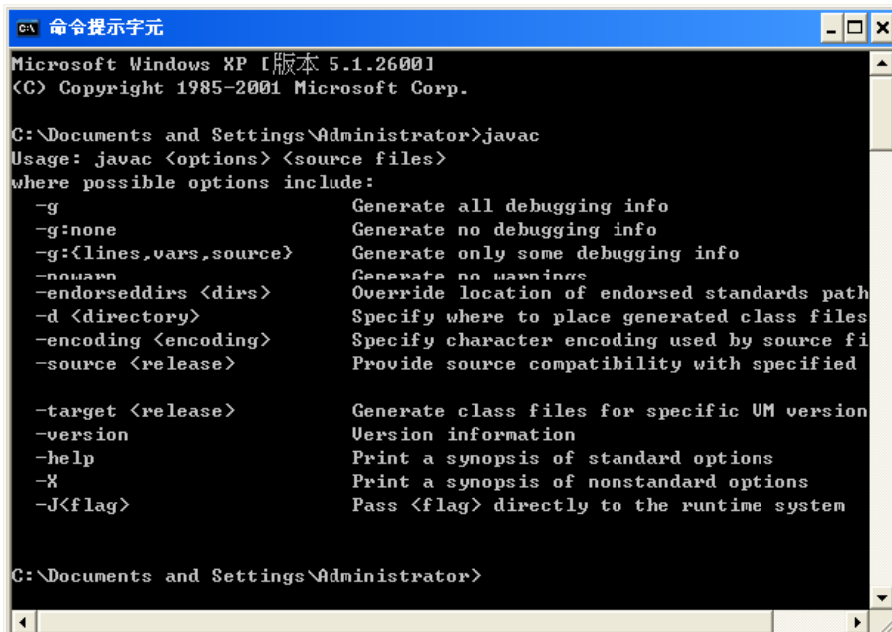
除非利用其他發展工具(如 JavaBult)，否則編譯與執行 Java 程式，大於『命令提示字元』視窗下，以『命令行』(Windows 命令)方式操作。開啟『命令提示字元』視窗如下：由桌面『開始』→『所有程式』→『附屬應用程式』，再點選『命令提示字元』，則可出現下列視窗(可建立捷徑於桌面上)：



### (B) 步驟 2：測試 J2SDK - javac

進入『命令提示字元』後，可立即執行『javac』命令測試 J2SDK 是否安裝正確。如下圖所示，執行 javac 後會出現一大串訊息，表示系統環境正確；如僅出現『javac』不是內部

或外部命令』，則可能是 J2SDK 安裝不正常，或環境變數 Path 沒有設定正確，請回到 1-3-1 (步驟 3)，重新安裝或設定 Path 變數。



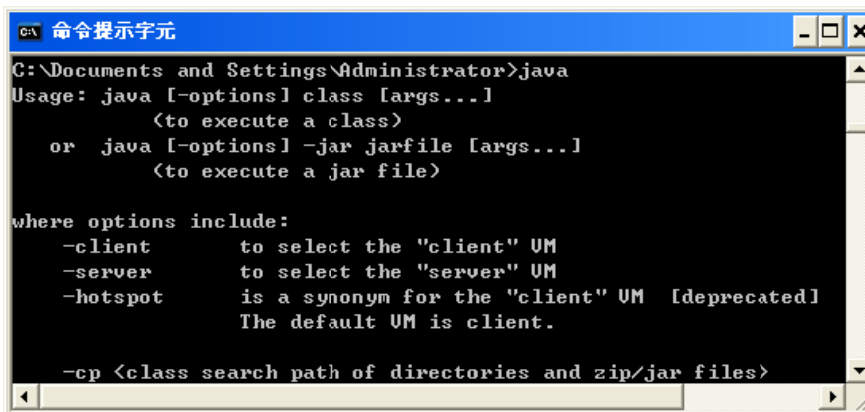
```
命令提示字元
Microsoft Windows XP [版本 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>javac
Usage: javac <options> <source files>
where possible options include:
  -g                Generate all debugging info
  -g:none           Generate no debugging info
  -g:{lines,vars,source}  Generate only some debugging info
  -nowarn           Generate no warnings
  -endorseddirs <dirs>  Override location of endorsed standards path
  -d <directory>     Specify where to place generated class files
  -encoding <encoding> Specify character encoding used by source files
  -source <release>   Provide source compatibility with specified release
  -target <release>   Generate class files for specific VM version
  -version          Version information
  -help            Print a synopsis of standard options
  -X               Print a synopsis of nonstandard options
  -J<flag>         Pass <flag> directly to the runtime system

C:\Documents and Settings\Administrator>
```

### (C) 步驟 3：測試 JRE - java

同樣的，執行 java 命令可以測試 JRE (Java Runtime Environment) 是否安裝正確，如出現下圖樣式，表示正常，否則必須重新安裝 jdk-1\_5\_0 (或其他版本) 套件。



```
命令提示字元
C:\Documents and Settings\Administrator>java
Usage: java [-options] class [args...]
           (to execute a class)
or java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
  -client          to select the "client" VM
  -server          to select the "server" VM
  -hotspot         is a synonym for the "client" VM [deprecated]
                  The default VM is client.

  -cp <class search path of directories and zip/jar files>
```

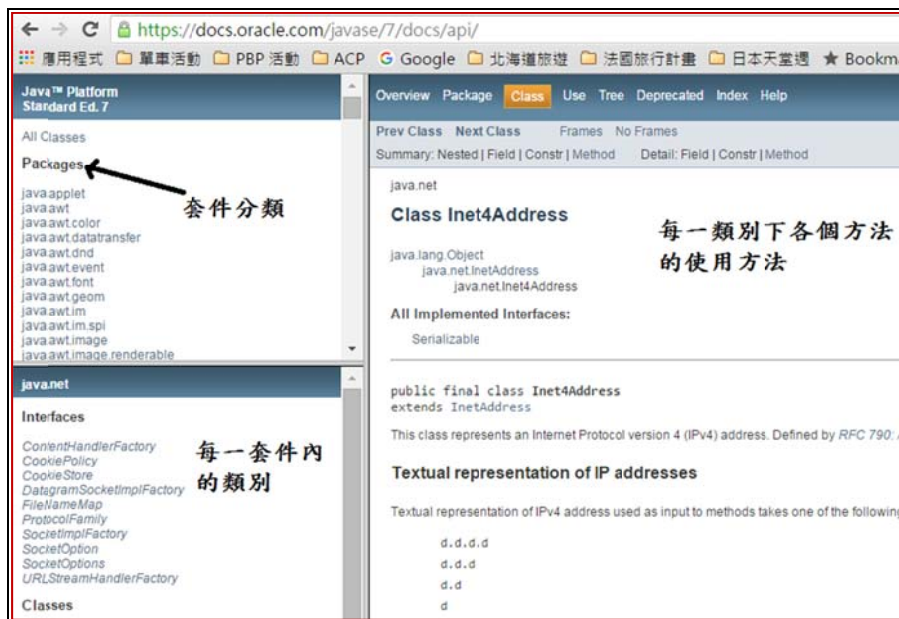
## 1-3-3 Java DOC 文件查詢

近十幾年，利用 Java 發展的軟體越來越廣泛，想得到的地方都有 Java 的蹤影，幾乎是包山包海，譬如：資料庫應用、網路應用、網路通訊、密碼系統、資訊安全、等等，連目前最夯的 App 也不放過。只要您想得到、有心學習，Java 都提供最佳的發展平台。為何 Java 有此廣大功能呢？因為它包含了各種領域應用系統的發展套件，只要您會使用這些套件，

發展任何應用系統都沒問題，然而，Java 將套件的使用方法，很仔細的歸納如下：（很可惜它都是英文，看不懂英文很難成為厲害的程式設計師）



該網站上列出目前有哪些套件(Packages, 左上角視窗)，當您選擇某一套件後，立即歸納出該套件有哪些類別(Classes, 左下角視窗)，每一種類別表現某一特殊功能。當選擇某一類別之後，將會顯示該類別包含哪些方法(method, 右邊視窗)，並顯示所有方法的引用方法，如下圖所示。



## 1-4 Windows 檔案結構與常用命令

基本上，JVM 在 Windows 系統是利用『命令提示字元』視窗操作的（透過其他發展工具，諸如如 JavaBult，就可省略 Windows 命令操作）。我們先認識一些簡單的 Windows 操作命令，對爾後操作電腦會有很大的幫助。

### 1-4-1 Windows 檔案結構

Windows 是利用樹狀結構來紀錄檔案儲存位置，如圖 1-7 所示。每一檔案儲存裝置以單一英文字母標示其名稱，如 A、B、C...等等；習慣上將 A 與 B 識別保留給軟式磁碟機使用，由 C 字母開始為硬式磁碟機、光碟機或隨身碟記號。每一磁碟機（如 C: 磁碟機）各自建立一只如樹根往下延伸的檔案系統，樹頭為所有樹根的起始位置稱之為『根』（root、C:\）；樹根往下延伸，會出現許多節點，如果某一節點允許再往下延伸，則稱為『目錄』（Directory）或『資料夾』；不允許往下延伸的端點則是『檔案』（File）。每一目錄（或資料夾）允許儲存多個檔案，以若干個子目錄。

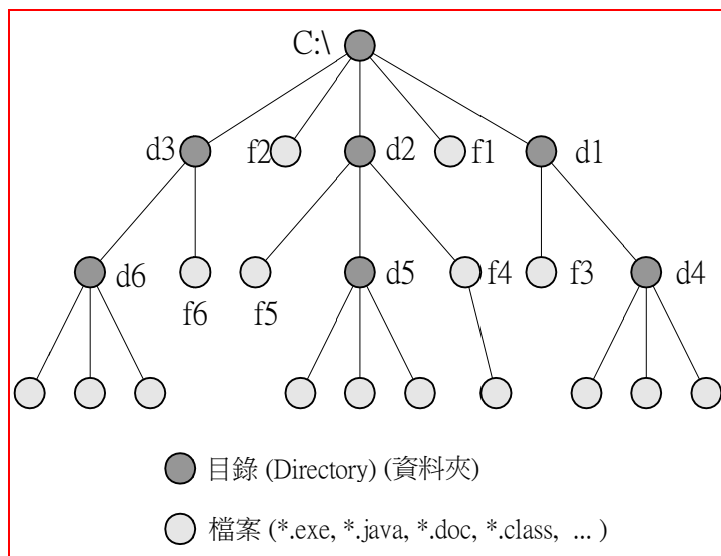


圖 1-7 Windows 檔案結構

每一個檔案儲存裝置（如磁碟機、光碟機、隨身碟...等）都有一隻樹狀檔案系統，圖 1-7 為 C 磁碟機上的檔案結構，檔案名稱格式如下：

- C:\f2：C 磁碟機根目錄下之檔案，檔案名稱為 f2。
- C:\d3\f2：C 磁碟機的 d3 子目錄下之檔案，名稱為 f2。
- C:\d2\d5：C 磁碟機的 d2 子目錄下的子目錄，目錄名稱為 d5。

另外，一般檔案的名稱由兩個元件所構成：檔案名稱.檔案型態，如 Example.java，則檔案名稱為 Example；檔案型態為 java。在 Windows 系統上，使用者開啟某一檔案時，它是由檔案型態來決定啟動哪一個軟體，因此，檔案型態很重要，不可以取錯。

## 1-4-2 Windows 常用命令



## (A) 檔案管理

在『命令提示字元』下，較常用的檔案管理工具有：

C:\> dir (顯示目前工作目錄下的檔案)

C:\> del f1.java (刪除 f1.java 檔案)

C:\> copy f2.java f3.java (複製 f2.java 成為 f3.java)

C:\> copy c:\d2\d4\f6 (複製 C:\d2\d4\f6 檔案到目前工作目錄下)

C:\> ren f2.java f3.java (將 f2.java 名稱改為 f3.java)

C:\> type f2.java (顯示 f2.java 檔案內容)

## (B) 目錄管理

在『命令提示字元』下，較常用的目錄操作工具有：

C:\> d: (切換到 D 磁碟機上操作)

D:\>cd \ (將工作目錄切換到根目錄)

D:\>md java (建立新目錄，名稱為 java)

D:\>cd java (將工作目錄切換到 java 目錄下)

D:\java> cd .. (切換到父目錄)

D:\> rd java (移除掉 java 目錄)

目錄操作上有兩個特殊符號，一個句點『.』表示目前目錄；兩個句點『..』表示父目錄(上一層目錄位置)。如系統顯示提示符號為『D:\java\Ex2\_1>』，則表示於磁碟機 D 下操作，目錄位置是 D 磁碟機根目錄(D:\)下的 java 子目錄(D:\java)，又 java 目錄下的 Ex2\_1 子目錄(D:\java\Ex2\_1)下操作之意思。

## 1-5 Java 程式開發

### 1-5-1 程式編輯工具 - Notepad

我們需要一只工具來編寫程式，最基本工具是『記事本』，但它缺乏程式碼的檢測能力，比較困難編寫程式。之前本書採用 Eclipse，但它主要再編寫 Android 程式使用(新版本也不用了)，產生太多附屬程式，反而讓學生產生困擾。因此，本書新的版本採用 Notepad 來製作程式，他可以檢測許多程式語言，譬如：C++、Java 或 PHP 等等語言，是目前最流行的工具。Notepad 套件是免費軟體，只要在 Google 搜尋與安裝即可。

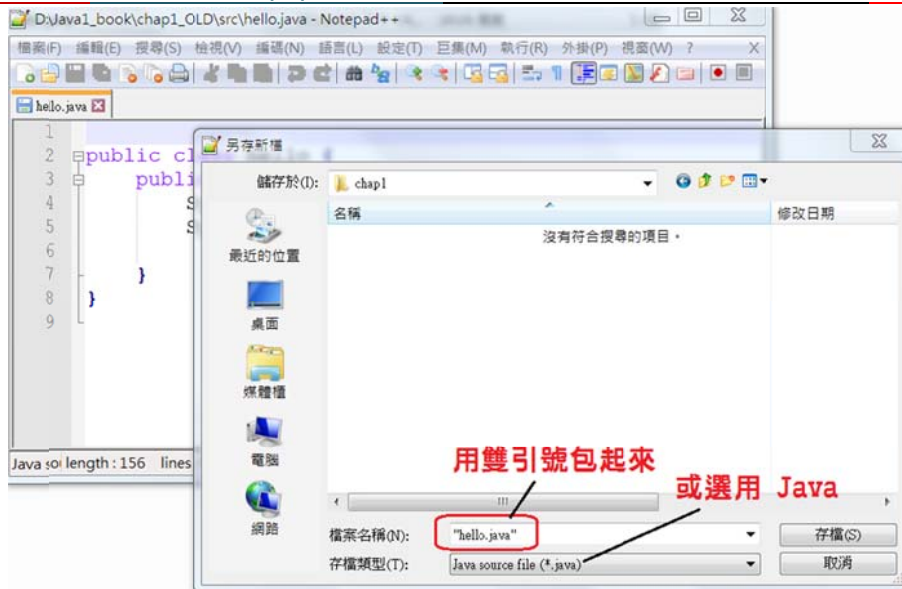
#### (A) 安裝 Notepad

只要在 Google 搜尋 Notepad，再下載安裝即可：(只要按下一步)



#### (B) 執行 Notepad

利用 Notepad 編輯後，指定檔案名稱與存入時，須注意，如輸入檔案型態則須用雙引號包起來，或下拉式選用檔案型態。



### 1-5-2 一般程式開發程序

完成 j2SE 套件安裝程序後，即可開始編寫 java 程式，編寫與執行 java 程式的步驟如圖 1-8 所示。編寫 Java 原始程式需利用沒有排版功能的『記事本』，否則會插入排版符號造成編譯的困擾，簡單說明編譯程序如下：

- (1) 編寫原始程式：**需依照 Java 程式規則，利用『記事本』編寫原始程式。完成後，儲存檔案名稱的副檔名需是 java，如 Demo.java 檔案。
- (2) 編譯原始程式：**完成原始程式製作後（如 Demo.java），則在『命令提示字元』視窗下，利用 javac 程式將原始檔案編譯成 Bytecode 檔（如 D:> javac Demo.java）。如果原始檔案沒有錯誤的話，則會產生中介程式（如 Demo.class）；如果原始程式中有關鍵字錯誤（語法錯誤，如錯別字）或規則錯誤（語意錯誤，如文法不對），則會顯示錯誤位置（行數），需重回『記事本』修改。
- (3) 執行程式：**如果原始檔案沒有任何語意或語法錯誤，經由 javac 編譯後，將產生中介程式（如 Demo.class）。接下來，可利用 java 執行器執行該中介程式（如 D:> java Demo）。程式執行的結果是否能滿足原目的，如果不能滿足的話，表示程式語法都對但邏輯上有問題，需重回『記事本』修改。修改後，也需再經過 javac 編譯，才可讓 java 執行。

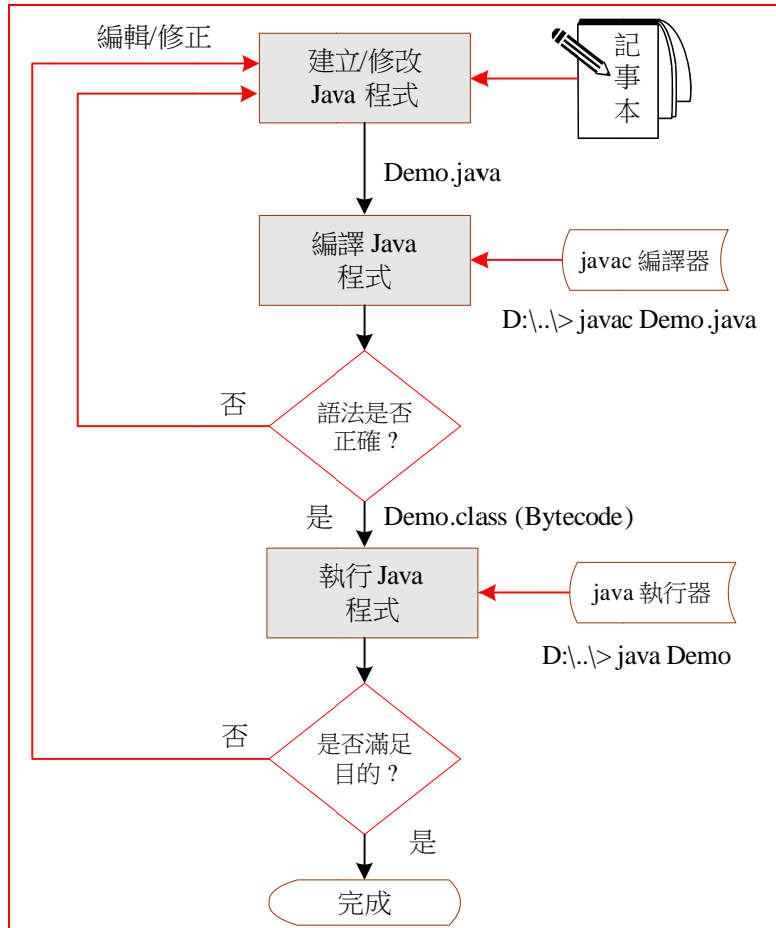


圖 1-8 Java 程式開發步驟

### 1-5-3 Java 程式開發步驟

本書將一個步驟接一個步驟，利用一個簡單範例 ( `Demo.java` )，分別說明各步驟的操作方法，如下：

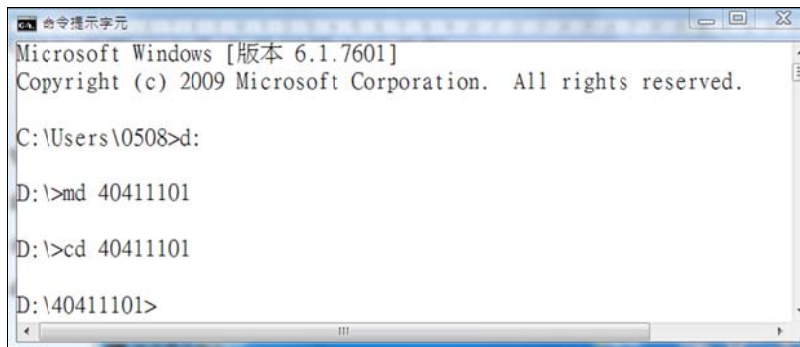
#### (A) 步驟 1：啟動『命令提示字元』

除非利用其他發展工具 ( 如 `JavaBult` )，否則編譯與執行 Java 程式，大於『命令提示字元』視窗下，以『命令行』( Windows 命令 ) 方式操作。開啟『命令提示字元』視窗如下：由桌面『開始』→『所有程式』→『附屬應用程式』，再點選『命令提示字元』，則可出現下列視窗 ( 可建立捷徑於桌面上 )：



## (B)步驟 2：建立並切換工作目錄

習慣上，我們大多將自行建立的檔案儲存於 D 磁碟機（假設 C 磁碟機安裝 Windows 系統）；又編寫 Java 程式時，一般都建議將一只程式放置於各自一個子目錄下（目前不嚴格限制）。假設目前欲練習編寫 Demo.java 程式，則建議在 D 磁碟機上，開啟一個專門存放 Java 程式的目錄，操作步驟如下圖所示。（假設 d:\tsnien\_java 目錄）

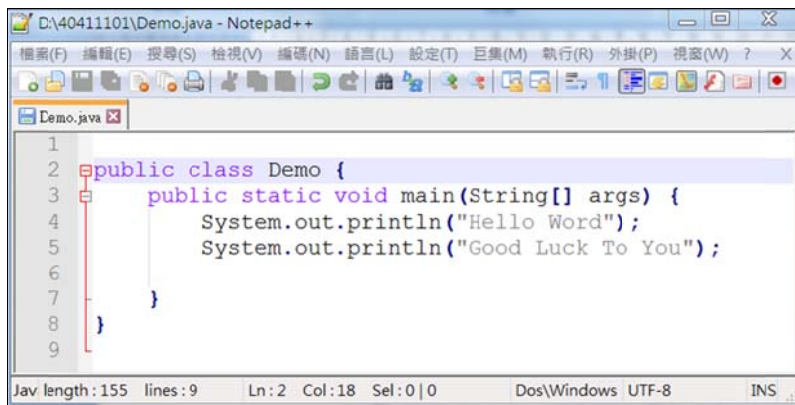


```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\0508>d:
D:\>md 40411101
D:\>cd 40411101
D:\40411101>
```

## (C)步驟 3：『NotePad』編寫程式

接著開始編寫程式，可利用『記事本』或 NotePad 編寫。如利用 NotePad 輸入 Demo.java 程式範例，如下圖所示。



```
D:\40411101\Demo.java - Notepad++
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 編碼(N) 語言(L) 設定(T) 巨集(M) 執行(R) 外掛(P) 視窗(W) ? X
Demo.java x
1
2 public class Demo {
3     public static void main(String[] args) {
4         System.out.println("Hello Word");
5         System.out.println("Good Luck To You");
6     }
7 }
8 }
9
Jav length: 155 lines: 9 Ln: 2 Col: 18 Sel: 0|0 Dos\Windows UTF-8 INS
```

## (D)步驟 4：儲存檔案

儲存於 D:\java 目錄下，檔名為 Demo.java，如下圖所示，其中『存檔型態』選擇「所有檔案」。儲存後，點選『命令提示字元』視窗，並執行 dir 命令，觀察儲存檔案後的結果。



進入命令提示字元，觀察儲存結果，如下：

```
命令提示字元
C:\Users\0508>d:
D:\>md 40411101
D:\>cd 40411101
D:\40411101>dir/b
Demo.java
D:\40411101>
```

### (E) 步驟 5：編譯 Java 程式 - javac

利用 javac 將原始檔案 ( Demo.java ) 編譯成 Bytecode 檔案 ( Demo.class )。如下圖所示。

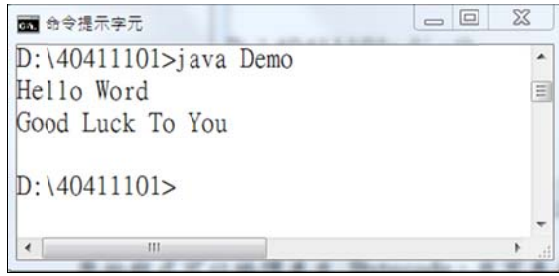
```
D:\40411101>dir/b
Demo.java
D:\40411101>javac Demo.java
D:\40411101>dir/b
Demo.class
Demo.java
```

如果原始檔案內有語意錯誤的地方，將會出現錯誤訊息，就必須重新利用『記事本』修改原始檔案；如果沒有錯誤的地方，則不會顯示任何訊息，再利用 dir/b 檢視是否產生 Bytecode 檔 ( Demo.class )。

### (F) 步驟 6：執行 Bytecode 程式 - java

產生 Bytecode 後，便可利用 JVM ( java 命令 ) 執行，下圖所示。





```
命令提示字元
D:\40411101>java Demo
Hello Word
Good Luck To You
D:\40411101>
```

原始程式可以編譯產生 Bytecode，並不表示一定可以利用 JVM ( java 命令 ) 執行。雖然可以執行，但不表示所編寫的程式，能滿足所需。因此，如不能執行或執行結果不正常，都必須再利用『記事本』重新修改程式。

## 1-6 Java 程式範例

### 1-6-1 範例研討：翻轉教室公告

#### (A) 程式功能：Ex1\_1.java

吾人期望顯示翻轉教室的公告如下：

```
***** 翻轉教室公佈事項 *****
(1) 學生上課前可收視 youtube 預習
(2) 每次播放視訊教學，以不超過 10 分鐘為限
(3) 課堂上學生輸入程式範例練習，有疑問隨時提問老師
(4) Youtube 播放挑戰題的製作技巧後，請學生自我挑戰
(5) 下課前，學生必須將範例與自我挑戰的製作結果上傳給老師
(6) 每次練習 4~5 題，一個學期練習 70 題以上，一定會寫程式
```

#### (B) 程式範例：

```
01 public class Ex1_1 {
02     public static void main(String[] args) {
03         System.out.println("***** 翻轉教室公佈事項 ***** ");
04         System.out.println("(1) 學生上課前可收視 youtube 預習");
05         System.out.println("(2) 每次播放視訊教學，以不超過 10 分鐘為限");
06         System.out.println("(3) 課堂上學生輸入程式範例練習，有疑問隨時提問老師");
07         System.out.println("(4) Youtube 播放挑戰題的製作技巧後，請學生自我挑戰");
08         System.out.println("(5) 下課前，學生必須將範例與自我挑戰的製作結果上傳給老師");
09         System.out.println("(6) 每次練習 4~5 題，一個學期練習 70 題以上，一定會寫程式");
10     }
};
```

```
}  
}
```

## 1-6-2 自我挑戰：上課注意事項

### (A) 程式功能：PM1\_1.java

吾人期望列印本課程上課注意事項：

```
***** Java 程式設計(一)上課公約 *****
```

- (1) 本課程採用『做中學』教學方式，學生多演練，老師從旁指導
- (2) 老師減少講課時間，儘量讓學習自我練習，並隨時提問老師
- (3) 課堂中隨時做範例與自我挑戰題目，學生一定要有書，老師會檢查
- (4) 慢來同學如沒聽到老師講解，請自行收視 youTube，有問題再提問
- (5) 下課前，學生必須將範例與自我挑戰的製作結果上傳給老師
- (6) 缺課同學，請自行收視 youTube 練習，需繳交作業

### (B) 程式提示：

```
01 ...  
02 System.out.println("***** Java 程式設計(一)上課公約 *****");  
03 System.out.println("(1) 本課程採用『做中學』教學方式，學生多演練，老師從旁指導");  
04 System.out.println("(2) 老師減少講課時間，儘量讓學習自我練習，並隨時提問老師");  
05 System.out.println("(3) 課堂中隨時做範例與自我挑戰題目，學生一定要有書，老師會檢查");  
06 System.out.println("(4) 慢來同學如沒聽到老師講解，請自行收視youTube，有問題再提問");  
07 System.out.println("(5) 下課前，學生必須將範例與自我挑戰的製作結果上傳給老師");  
08 System.out.println("(6) 缺課同學，請自行收視 youTube 練習，需繳交作業 ");  
09  
10  
11 ....  
12
```