第八章　套件的使用

本章內容

- 8-1　Java 的 .class 檔案
- 8-2　套件 (Package) 的編譯與執行
- 8-3　引用套件下的類別
- 8-4　使用 jar 檔

8-1 Java 的 .class 檔案

- Java 的原始程式碼 (.java 檔) 在經過編譯後會產生「.class」檔案。而「.class」檔案數目會依據原始檔中定義的類別數目而定。例如：以下的程式碼會產生三個 .class 檔。

```
class Dog{ }
class Fish{ }
class Cat{ }
```

- 如果 Java 的原始檔中所有類別的型態都是預設型態，您可以用任何一個類別的名稱當作是檔案的名稱。
- 雖然 Java 的原始檔中可以有多個預設型態的類別，但每一個 Java 原始檔中只能有一個「public」的類別，而且，該 Java 程式檔案的檔名必需和「public」類別的名稱相同，否則，程式編譯時會產生錯誤訊息。

8-2 套件 (Package) 的編譯與執行

- 「**套件 (package)**」 概念的主要用途就是用來將類別做分類管理，並且讓類別的使用能夠更方便。
- 定義及編譯套件
  - 如果要定義套件，我們可以使用以下語法

```
package packagename;
```

- 套件的定義敘述句必需置放於程式中非註解的第一行。
- 程式中不可以有一行以上的「package」敘述句。

8-2 套件 (Package) 的編譯與執行…

- package 的名稱可以是一組有意義的文字，也可以是一連串以句號 （ . ） 組合而成的文字。 您可以將套件名稱中的句號 （ . ） 當做是 Windows 作業系統下的表示路徑的分隔符號 （ \ ）。例如:

```
package stock.sale;
public class Customer {}
```

- 預設的情況下，原始檔在編譯後，會在原始檔所在的目錄下產生「 stock 」資料夾，並在「 stock 」資料夾下再產生「 sale 」子資料夾，而編譯後的「 Customer.class 」檔案就會放在「 sale 」資料夾中。

8-2 套件 (Package) 的編譯與執行…

- Java 程式中的每一個類別檔都一定是屬於某一個套件，位於同一個套件下的類別檔可以互相的引用。

- 對於沒有定義 package 敘述句的 Java 檔案，它們所產生的類別檔是置放於「**預設套件 (default package)**」中，也就是 Java 原始檔案所在的同一個目錄 (現行目錄) 下。

8-2 套件 (Package) 的編譯與執行…

- 如果要在現行目錄下直接編譯具有 package 敘述句的 Java 程式時，我們必需在「 javac 」指令後方再加上「 -d 」參數，參數後方再加上目錄名稱，目前我們可以使用句點 (.) 來代替現行目錄。最後再加上需要編譯的程式名稱。例如:

```
javac –d . myPackage.java
```

- 如果需要，您也可以一次編譯兩個以上的 Java 原始檔，例如:

```
javac –d . myPackage.java myPackage.java
```

8-2 套件（Package）的編譯與執行…

- 如果不特別指定「-d」參數來編譯具有「package mypackage;」敘述的 java 原始檔。以範例中的「mypackage.java」為例，我們可以依照以下的方式來完成開發及編譯的流程:

  一、在現行目錄下建立 mypackage 資料夾。

  二、在現行目錄下使用編輯軟體撰寫程式。

  三、將程式檔儲存在現行目錄下的 mypackage 資料夾中，並命名為「mypackage.java」。

  四、在現行目錄下，使用「javac」指令，並指明程式檔的儲存相對路徑及檔名，來編譯程式。指令如下：「javac .\mypackage\mypackage.java」。

8-2-1 執行套件

- 在編譯完宣告有 package 敘述句的 Java 原始檔後，如果要執行編譯完成的類別檔，您不可以直接進入該資料夾下去執行檔案。否則會發生找不到執行類別的錯誤。
- 要執行套件中的類別檔時，您必需指明套件的名稱及需要執行的類別檔的名稱， package 和 package 之間，或是 package 和 class 名稱之間必需以「.」作區隔。例如:

```
C:\Work\Chap8>java mypackage.PackageTest
```

8-2-2 使用 classpath 參數執行套件

- 執行 package 時，您也可以利用「 classpath 」參數來指明需要執行的檔案。
- 例如：如果我們要執行的檔案名稱為「 PackageTest.class 」。該檔案是位於「 C:\Work\Chap8 」目錄的「 mypackage 」套件 (目錄)　下。我們可以在任何的資料夾下，利用以下的指令來執行該檔:

```
C:\Work>java  -classpath  .;c:\Work\Chap8\
mypackage.PackageTest
```

8-2-3 使用 classpath 參數執行套件…

- 如果您確定您的應用程式是放在「c:\Work\Chap8\」路徑下的「mypackage」套件下的「PackageTest.class」檔案。您也可以命令視窗的指令行上直接設定「CLASSPATH」的路徑為：

```
set CLASSPATH=%CLASSPATH%;C:\Work\Chap8\
```

- 只要是目前的命令視窗未關閉，您可以在任何位置上

```
C:\Work>java myackage.PackageTest
```

8-3 引用套件下的類別

- 我們可以利用「 import 」敘述來將相關的套件引入，並使用該套件中的類別檔。 import 敘述的使用方式如下：

```
import java.awt.color;
import java.awt.*;
```

- 您可以在程式中重覆的使用「 import 」敘述，以便引入多個套件。程式中也可以使用「 * 」來代表要引入套件下的所有類別。
- 每個 Java 程式都會有預設的 import 敘述句，該敘述句會將 java.lang 套件下的所有類別引入。

8-3-1 使用 import static 敘述－J2SE 5.0

- 新的 import 敘述句可以直接將類別中的 static 方法引入，讓使用該方法時，不需要再加上類別的名稱。新的 import 敘述句多了一個 static 關鍵字，語法如下：

```
import static  套件名稱 . 類別名稱 . 方法名稱 ;
```

- 例如：如果要把 out 方法引入，我們可以使用以下的程式

```
import static java.lang.System.out;
```

-

```
out.println();
```

8-3-2 使用存取修飾子

- 使用不同的存取修飾子可以限制類別或是成員的存取模式，參考下表:

位置	private	(default)	protected	public
同一類別	統	▪	▪	▪
同一套件中的子類別		▪	▪	▪
同一套件，但不是子類別		▪	▪	▪
不同套件的子類別			▪	▪
不同套件，也不是子類別				▪

8-3-3 Import static 的問題

- **屬性遮蔽的問題**
  - 由於使用 import static 引入靜態屬性或是方法時，可以在使用上不需要再加上類別的名稱，因此，在識別上衍生一些問題。當您使用 import static 敘述句時，如果在類別或方法中宣告了相同名稱的屬性或是方法，則使用 import static 敘述句引入的屬性的值會被類別或方法中相同名稱的屬性所遮蔽。
- **方法的衝突問題**
  - 如果使用 static import 敘述同時引入這兩個類別中具有相同名稱的方法時，程式會產生編譯時的錯誤。

8-4 使用 jar 檔

- 「jar」是 Java Archive 的簡稱，它可以當作是類別的集合。
- jar 是以 ZIP 的壓縮格式來處理的，您可以將某個 package 下的所有類別檔或是子目錄打包成一個 jar 檔案。
- 您只提供該 jar 檔，別人就可以使用您開發的程式了。您可以使用「jar」指令將檔案打包成單一的 jar 檔。「jar」的指令為:

```
jar [參數] 目的檔名 要加入的檔案或路徑
```

8-4 使用 jar 檔 …

- 「jar」的指令可以使用的參數如下:

參數	作用
c	建立一個新的 jar 檔
t	列出 jar 檔案的內容
x file	解開 jar 檔案內 file 檔名的檔案，如果省略 file 參數，則會解開 jar 檔案內所有的檔案
f	指定需要處理的 jar 檔
v	在標準輸出的裝置上產生訊息
m	加入清單檔
o	只將檔案加入 jar 檔中，但不壓縮
M	不建立清單檔
u	更新某個 jar 檔的內容
-C	切換至某個目錄再執行 jar 相關指令

8-4 使用 jar 檔 …

- 更新 JAR 檔案的內容：使用 u 參數

```
jar -uvf test.jar myPackage\Update.class
```

- 顯示 JAR 檔案的內容：使用 tf 參數

```
jar uvf test.jar myPackage
```

- 取出 JAR 檔案的內容：使用 xf 參數

```
jar xf test.jar
```

使用 JAR 檔案中的類別

- 您有三種方式可以使用到某個 JAR 檔中的類別。
  - 第一種方式是直接將 JAR 放在「 JAVA_HOME 」所設定的路徑中。
  - 第二種方式是在每次編譯引入套件的 Java 檔案時，使用「 classpath 」參數來指明 JAR 檔案的位置。
  - 第三種方式是直接設定 CLASSPATH 的路徑。

```
javac –classpath C:\Work\Chap8\display.jar
UseJar.java
```