

深入Java物件導向程式設計

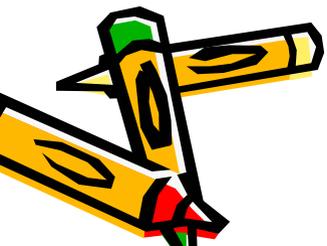
第八章

Java2全方位學習



大綱

- 套件的使用
- 修飾子的使用
- 抽象類別
- 介面



套件

- package
- 類別的分類
- 對應至檔案目錄



套件的宣告

- 所有程式碼的第一行
- package陳述式
- 通常將Hostname倒過來做為套件名稱
- 所有英文字母都是小寫
- 用句點(.)分隔
- 檔案系統上需建構對應的目錄，將檔案放入其中

程式編譯與執行

- 需切換目錄到套件所在的根目錄
- 編譯
 - javac <套件對應目錄>\<Java檔案名稱>
- 執行
 - java <套件名稱>.<類別名稱>

import陳述式

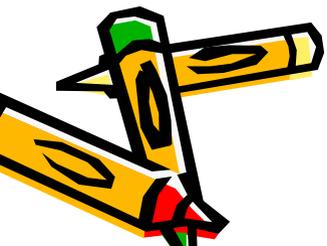
- 載入指定的套件
- 可使用萬用字元(*)
 - 不包含其下的子套件
- 需寫在package陳述式之後，類別宣告之前
- 預設import
 - java.lang.*

CLASSPATH設定

- 讓javac或java找到的所需要的類別
- 設定
 - 環境變數
 - 執行時設定
 - -classpath參數
- 預設CLASSPATH(.)

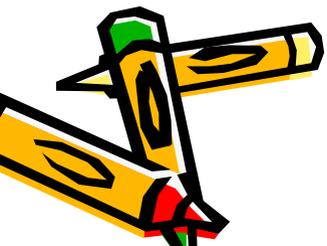
CLASSPATH衝突

- 預設CLASSPATH
- 路徑順序
- 執行參數覆蓋

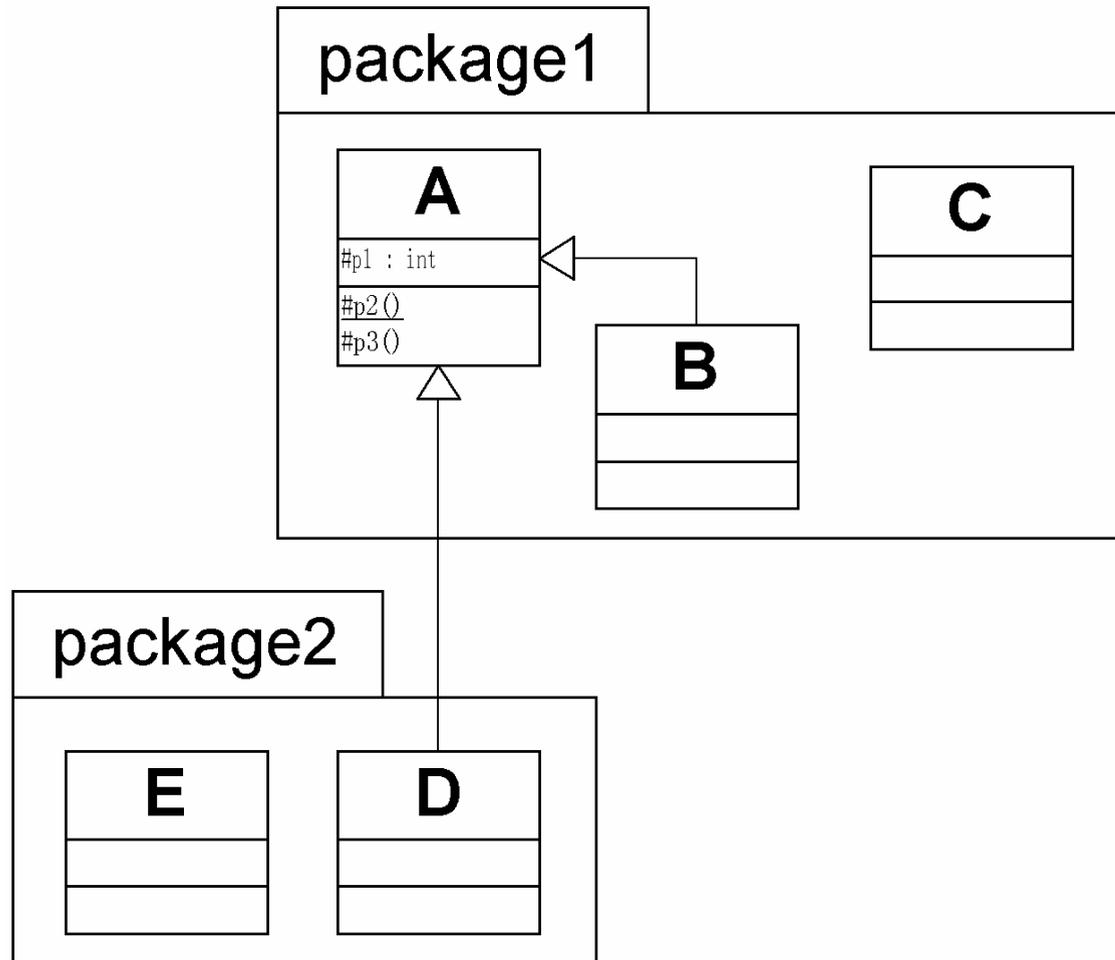


存取權修飾子

- public
- protected
- (default)
 - 沒有指定其它三個存取權修飾子時
 - package
- private

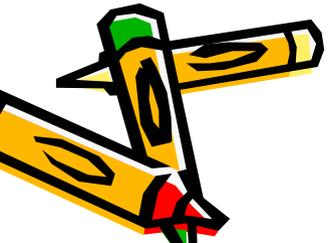


圖例



protected

- 保護成員(存取)
 - 套件
 - 繼承



使用時機(一)

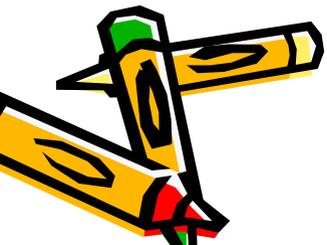
- 只有公用的程式中的某些方法，例如前面使用到的setLegs等...或是常數，才會使用使用public修飾子。基本上，屬性通常是使用private不會使用prulic修飾子
- 某個成員對於大部份程式來說不太會去使用到，但是對於別的package中繼承的子類別中會去使用到，則使用protected修飾子

使用時機(二)

- 如果只想讓同package中的類別所使用的方法或是屬性，則使用(default)修飾子
- 對於只是自己類別內部所使用的屬性或方法，則使用private修飾子來隱藏它們
- 如果你不確定該用何種修飾子，最好設定為(default)，這樣需要開放一點時，可以用覆寫的方式來讓它更開放

(default)

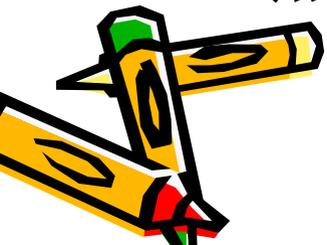
- 沒有指定其它三個存取權修飾子時
- 只有同套件的類別能存取該成員



final修飾子

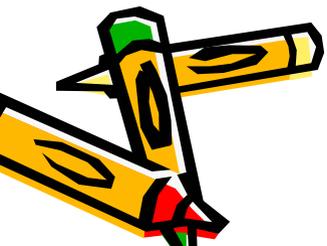
類別

- 不可被繼承
- 屬性
 - 常數
 - 大寫英文字母
- 方法
 - 無法被覆寫



final和static

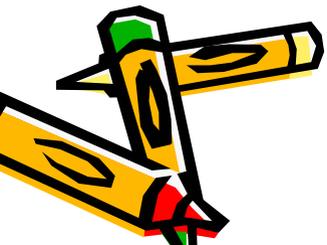
- 一起使用可節省記憶體的使用
- 但也不一定兩者一定要一起使用



抽象類別

類別中包含了抽象的方法

- 抽象方法
 - 只有方法宣告，沒有方法實作
- abstract修飾子

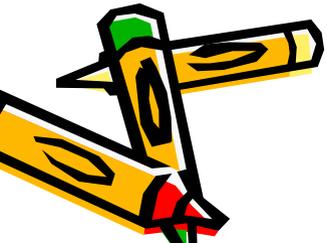


抽象類別使用限制

- 不能產生實體物件
 - 還是需要建構子
- 繼承抽象類別的子類別中，必需實作那些抽象方法，否則子類別也必需宣告為抽象類別

介面

- 完全抽象類別
- 成員
 - 常數
 - 抽象方法
- 統一操作的方式
- interface 關鍵字



介面的宣告

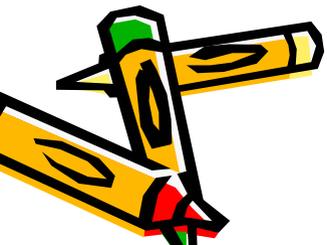
- <修飾子> interface <介面名稱>
[extends <其它介面>]
- 屬性自動宣告為static final
- 方法自動宣告為abstract

介面的實作

- implements <介面名稱1>, <介面名稱2>...
- 一個類別可同時實作許多個不同的介面，但如果沒有把介面中有的方法實作出來的話，那麼該類別得宣告為抽象類別

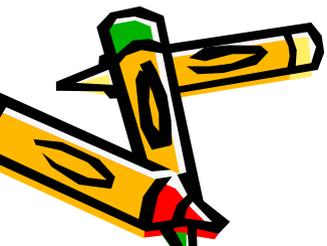
標記介面

- 功能上的分類
- 空的介面
 - 不包含任何的成員宣告



抽象類別與介面

- 功能的區分
- 單一繼承
- 可配合多形觀念使用



Q&A

