

**應用軟體設計實習**

Hw-CH4

任課教師：黃士嘉 老師

班級：電子三甲

學生：謝佳丞 106360122

心得:

這次的練習繼承和介面，在C++裡面的多重繼承是指一個子類別可以繼承多個類別的意思，而在Java裡的一個子類別只能繼承一個父類別，在Java裡只能多重繼承”介面”而已，我很好奇為甚麼同為物件導向的語言為何在這個方面會有差異?

原來在多重繼承在使用上是要非常小心的，舉個例子來說:

假設有一個木造門則:  
  
1. 此木造門是門的一種(a kind of)。  
木造門(子類別) 可繼承 門(父類別)  
2. 但門不是木造門的一部份(a part of)。  
  
3. 木造門是木製品的一種。  
木造門(子類別) 可繼承 木製品(父類別)  
4. 但木製品不是木造門的一部份。木製品也不是門的一種。  
  
5. 門也不是木製品的一種。

木製品跟門，在實用設計上不應該互相繼承。

由此可以發現多重繼續在使用時必須非常小心而且在許多時候其實我們並不需要多重繼承的。而Java在這方面做得比較謹慎，但是多重繼承在程式設計上還是備受所需，所以Java提供了一個interface，可供一個子類別去繼承，那麼剩下的問題就是interface跟class的差別到底在哪裡了?

interface就是一種介面規定欲溝通的兩物件其通訊該有的規範有哪些。如以Java程式語言的角度來看Java的interface則表示:一些函式或資料成員為另一些屬於不同類別的物件所需共同擁有則將這些函式與資料成員定義在一個interface中然後讓所有不同類別的Java物件可以共同操作使用之。

介面(interface)的特性:

* 介面和類別一樣有成員和方法，但是沒有建構式。
* 介面的成員不會加修飾子，但還是等於在成員前面加public static final修飾子、方法前面加上abstract修飾子。也就是說介面的欄位是常數，方法是抽象方法。
* 介面和類別很像，但是介面卻無法建立物件。也就是說，無法使用new來建立新的物件。

**抽象方法的好處，基本上就是可以透過一個共通的介面（interface），來操作不同類型的物件。**

我依然覺得interface與class的使用時機還是令我感到疑惑，但是我現在分的清他們的區別，可能以後多寫一些程式就能區分了吧?

上傳: