國立屏東大學 資訊工程學系 物件導向程式設計

13. Information Hiding and Encapsulation

- Information Hiding
- Encapsulation
- Access modifiers
- Setters and Getters

在物件導向程式設計裡面有兩個很重要的概念,就是資訊隱藏(information hiding)和封裝(encapsulation)☐ 資料隱藏是指當一個物件被外界存取的時候,外界無法得知物件內部的運作方式,或者是物件內部使用了哪些私有的資料成員和函式成員。而封裝指的是因為外界無法得知物件內部情形,只能使用物件提供給外界存取的函式,就好像一台有外殼包裝的機器,您只能看到按鈕,看不到裡面的電路板。

要做到資訊隱藏(information hiding)可以使用存取修飾字,讓物件的使用者無法存取內部的資料項目與方法。至於封裝(encapsulation)[則可以透過設計供外界使用的界面來完成。

13.1 存取修飾字

要達到資訊隱藏和封裝,就一定要了解Java如何控制類別[field與method的存取權限]public[protected和private這三個存取修飾字(access modifier)是Java用來控制存取權限的識別字。

就如同字面上的意思,這三個access modifier的意思分別是公開的、被保護的和私有的,分別有著不同的程度的保護層級。

- public是指對存取權限完全的公開,任何物件都可以存取這個成員,當然也包括了子類別□□
- protected是指在套件內、在子類別都可以存取,只有套件外的物件無法存取。
- private是限制最嚴格的access modifier□只有在本身類別內部可以存取。

我們將Java語言提供的三個存取修飾字[]public[]protected與private[]再加上不使用修飾字的話共有四種情形,彙整於table 1

位置	private	protected	public	不使用修飾字
同一個類別中	v	V	V	V
同一個package中的子類別		V	V	V
同一個package中的其它類別		V	V	V
不同package中的子類別		V	٧	
不同package中的其它類別			V	

Tab. 1: Access Modifiers and Accessibility

13.2 類別定義與存取控制

一般來說,我們都會將類別的field和某些只供內部使用的method設為private□然後開放一些method做為

Total: 231172

Last update: 2019/07/02 15:01

和外界溝通的介面,這些method通常是設為public或者是protected 以好處是類別內部的程式修改不會影響到其它的類別,而其它外界類別因為無法存取類別內部的filed 所以會減少不可預知的程式錯誤,而能夠控制程式除錯的可能範圍。

我們把定義類別的語法,增加<mark>存取修飾字(access modifier)</mark>,來限制類別□field與method的使用,其語法如下:

13.3 供外部使用的界面: Setters and Getters

適當地定義存取權限,讓其它程式只能透過我們所開放的method來存取field[可以確保程式碼的正確性與安全性。如果想開放field的存取,我們通常會設計成public的setXXX()與getXXX()methods來讓他人使用。

- setters又稱為Mutators□其method命名通常為setXXX()□
- getters又稱為Accessors□其method命名通常為getXXX()□

現在,我們再將Person類別修改如下:

```
class Person
{
    // field declarations
    private String firstname;
    private String lastname;

    // constructors
    Person()
    {
        firstname = "unknown";
        lastname = "unknown";
    }
}
```

```
Person(String f, String l)
      firstname = f;
      lastname =l;
   }
   // method declarations
   //setters
   public void setFirstname(String f)
   {
      firstname = f;
   }
   public void setLastname(String l)
      lastname = l;
   }
   //gettters
   String getFirstname()
   {
      return firstname;
   }
   String getLastname()
      return lastname;
   }
   public void showInfo()
      System.out.println("Name: " + firstname + " " + lastname );
   }
}
```

關於子類別的部份,本書後續章節會加以介紹。

```
From:
https://junwu.nptu.edu.tw/dokuwiki/ - Jun Wu的教學網頁
國立屏東大學資訊工程學系
CSIE, NPTU
Total: 231172

Permanent link:
https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=java:encapsulation

Last update: 2019/07/02 15:01
```

Jun Wu的教學網頁 國立屏東大學資訊工程學系 CSIE, NPTU Total: 231172