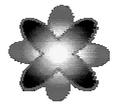


第十一章 衍生新的類別

重點整理

對Java來說，根據既有的類別為基礎，很輕易的就能夠衍生出新的類別。換言之，既有的類別再加上衍生的類別，對於程式開發有一定的幫助。本章的重點，就是要把這些產生新類別的方法介紹給您認識。

- ▶ 繼承
- ▶ 父類別
- ▶ 子類別
- ▶ `super ()`
- ▶ `protected` 成員
- ▶ 改寫
- ▶ `final`
- ▶ `Object` 類別

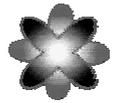


關於類別的繼承

Java可以根據既有的「Car類別」為基礎，輕易的就產生另一個可以用來表示賽車的「RacingCar類別」，像這樣以既有的類別為基礎，衍生出另一個類別的現象，稱為：

類別的延伸(extends)

換句話說，新的類別可以從舊類別繼承而來，包括舊類別的性質和功能（2者統稱為成員）也會一併繼承到新類別當中。



類別的延伸

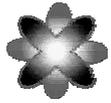
在宣告子類別的時候，必須在程式中使用extends這個關鍵字，後面再加上父類別的名稱。只要宣告的方式正確，可以從父類別Car衍生出子類別RacingCar。

延伸父類別之後，就可以宣告子類別。

而子類別會繼承父類別的成員。

語法

```
class 子類別名稱 extends 父類別名稱  
{  
    子類別新增的成員...  
    子類別的建構元(參數...)  
    {  
        ...  
    }  
}
```



產生子類別的物件

父類別延伸之後可以設計新的子類別。

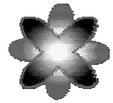


```
class Sample1
{
    public static void main(String args[])
    {
        RacingCar rccar1;
        rccar1 = new RacingCar();
        rccar1.setCar(1234, 20.5);
        rccar1.setCourse(5);
    }
}
```

產生子類別的物件

呼叫繼承而來的method

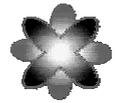
呼叫子類別新增的method



呼叫父類別的建構元

如果您沒有任何的指定，產生子類別的物件時，會在子類別建構元的開頭，先呼叫父類別中沒有引數的建構元並加以執行。

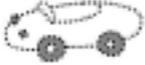
父類別的建構元其實並沒有繼承給子類別，不過子類別仍然會自動呼叫父類別的建構元，為的就是要幫助子類別對繼承自父類別的成員進行初始值的設定工作。



使用super()

如果您想要從子類別的建構元，呼叫父類別當中特定的建構元，必須使用super()關鍵字。

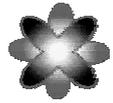
```
public Car()  
{  
}  
  
public Car(int n, double g)  
{  
}
```



```
public RacingCar(int n, double g, int c)  
{  
    super(n, g);  
}
```



透過super()可以
呼叫特定的建構元



從子類別存取其他類別的成員資料

所謂的成員(member)，指的是一個類別(class)內部的field和method。

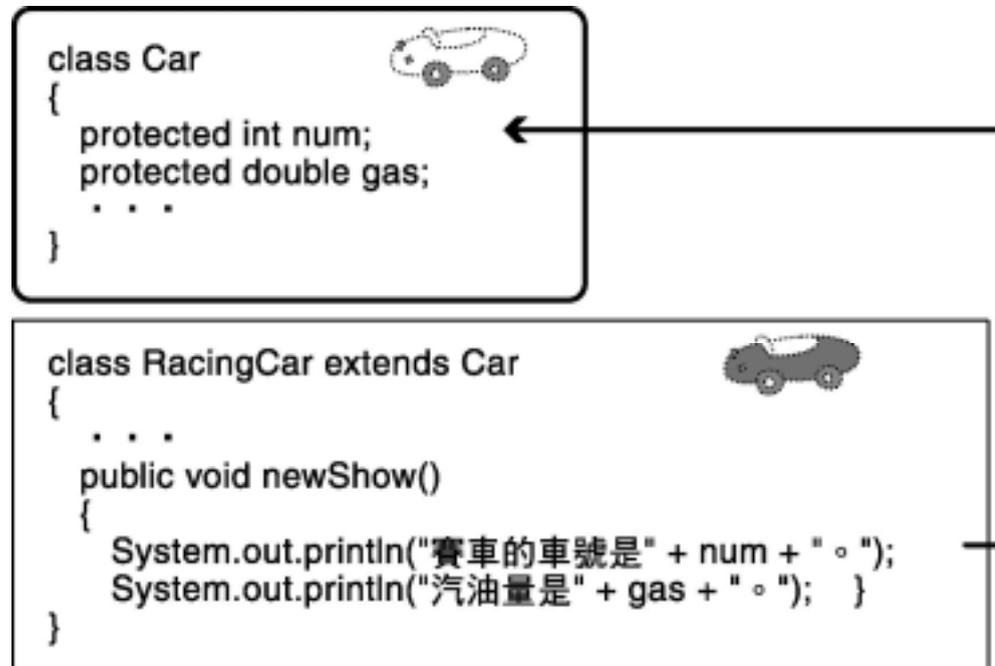
如果父類別裏面的成員原本就是私有成員，則子類別將無法存取到這些私有類別的資料。

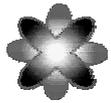
■ 私有成員

私有成員不能被同一類別以外的其他程式存取，因此如果父類別裏面的成員原本就是私有成員，則子類別將無法存取到這些私有類別的資料。

■ 保護成員

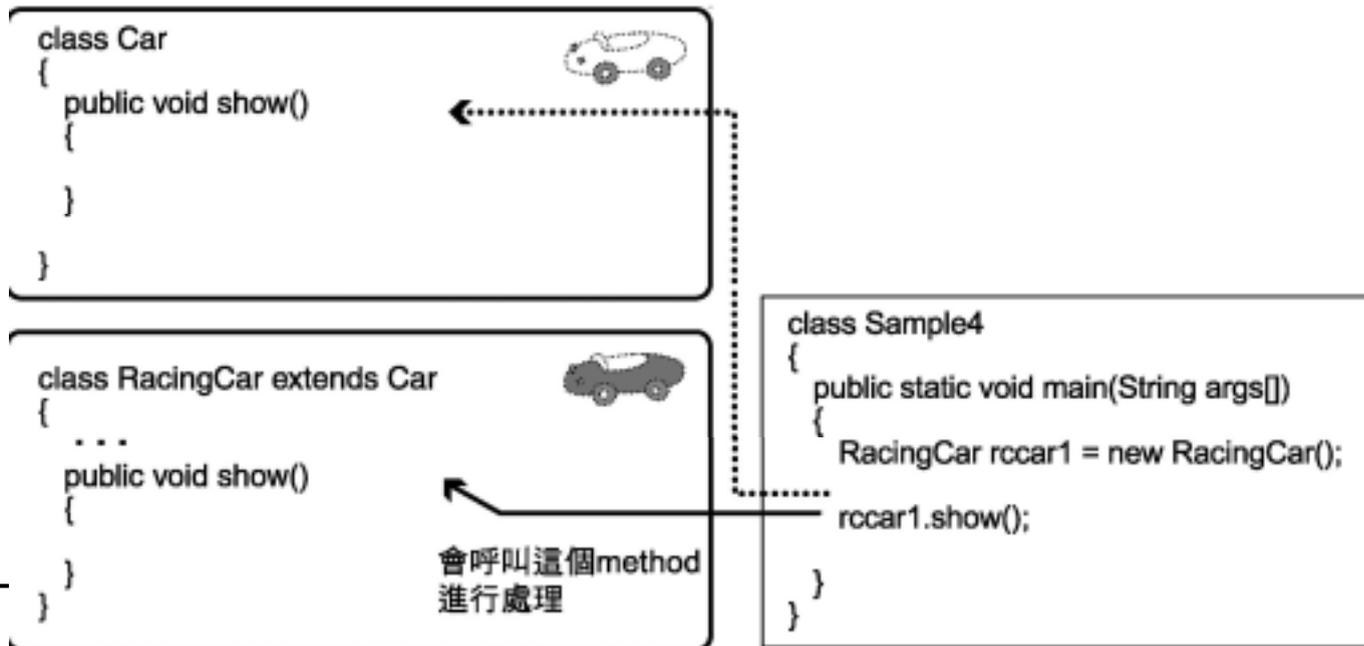
父類別內部的成員一旦設定為「protected」，其性質和private member有所不同，也就是說：子類別可以存取父類別內部protected member的資料

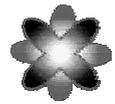




overriding method

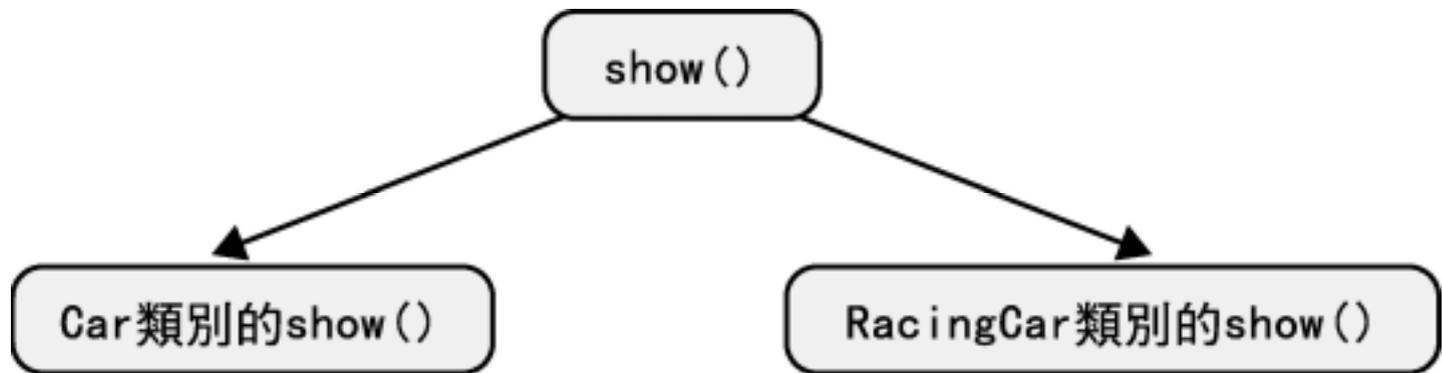
在子類別中定義一個和父類別的method名稱、參數數目、資料型態完全相同的method，用來取代父類別當中原本的method功能，稱為改寫(overriding)。

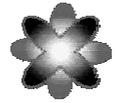




瞭解改寫(overriding)的重要性

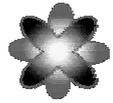
在一個稍為大型的程式中一定會產生各式各樣的物件，因此對各種類別所產生的物件加以管理是必然的事，於是有人就想到利用下列這種「父類別的陣列」來儲存各種類別所產生的物件。





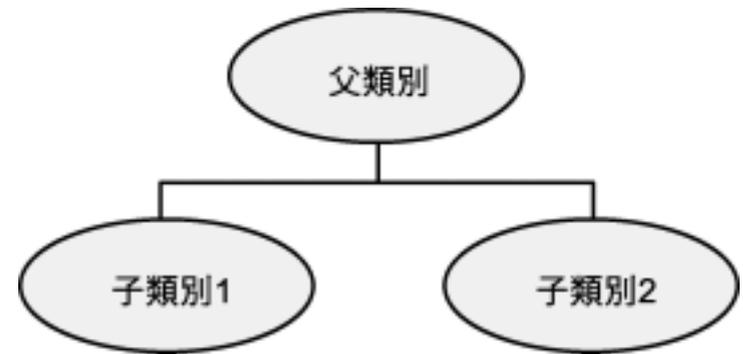
指定 final

當您在設計類別的時候，其中可能有一些父類別的 method 您根本不想讓它在子類別中產生被改寫 (overriding) 的現象。如果有這種需求時，您必須在父類別的 method 的第 1 行加上 final，就會禁止被改寫現象繼續產生。

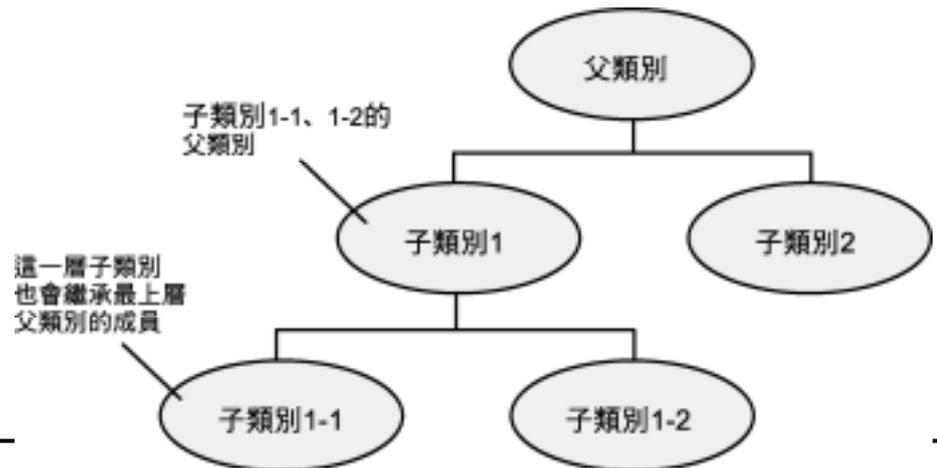


建立類別的階層

對Java來說，一個父類別可以延伸出數個子類別，這時候整個類別家族的結構如右圖所示：



子類別再繼續往下延伸，可以再產生新的子類別。



✿ 關於Object類別

如果該類別當時沒有指定任何父類別，則「Java會自動設定讓該類別繼承java.lang.Object這個類別做為父類別」。

語法

```
class Car •  
{  
    ...  
}
```

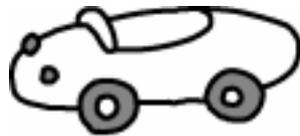
什麼都沒有指定時，就會以Object類別做為父類別，自己則變成子類別

❁ 定義toString() method

toString() method，這個method的主要用意是：

把物件轉成字串並將結果傳回原呼叫程式。

使用toString() method時，最好是自行定義toString() method、然後再加以呼叫，比較能獲得自己想要的結果。



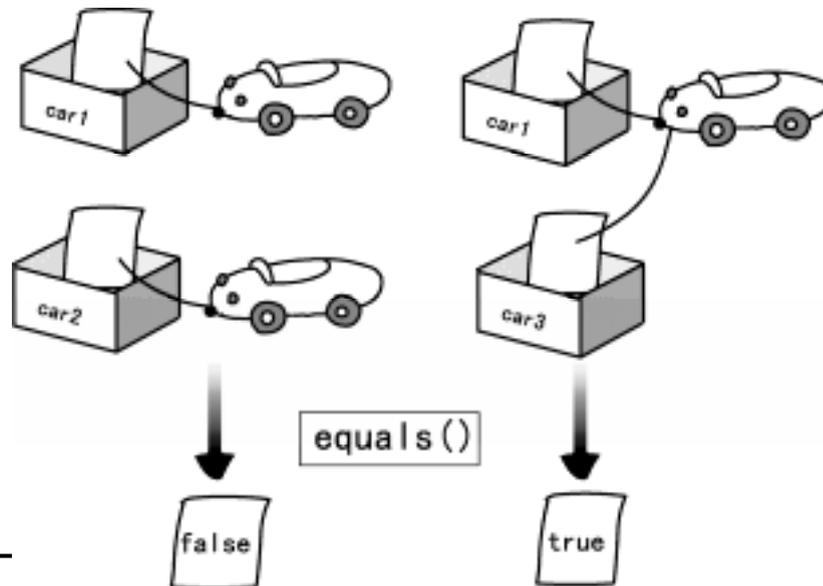
toString()

"車號：○ 汽油量：×"

✿ 使用equals() method

equals() method，其主要用意是：

驗證某兩個物件是否相同，是的話傳回true，否的話傳回false。



使用getClass() method

getClass() method，這個method的用意是：

傳回「該物件究竟屬於哪一個類別」這樣的相關資訊。

getClass() method會把某物件的相關類別資料匯集起來並傳回一個Class類別的物件。

