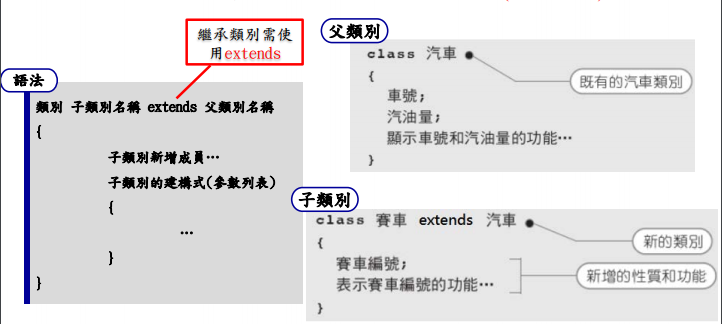
電子三甲 107360133 蔡東霖 Java Chapter4 心得報告

Chapter4主要在介紹繼承和介面。

根據既有的類別基礎產生另外一個新的類別，這個動作稱為類別的延伸(extends)，如果新類別是完全接受既有類別的成員，那麼則稱為繼承(inheritance)。



Ex:

Class Car{……………};

Class RacingCar extends Car{……………}

RacingCar為子類別，Car為父類別

* 私有成員(private member)：

私有成員無法被同一類別以外的其他程式存取，子類別也無法讀取父類別中原有的私有成員。

* 保護成員(protected member)：

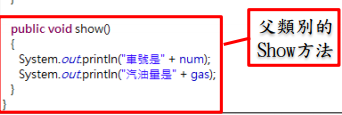
性質和private member不同，子類別可以讀取父類別中

原有的protected member。

**覆寫(Overriding)：**

覆寫代表著子類別繼承父類別的同時，改寫了父類別中原有的方法，但是該方法的參數、方法名稱、以及傳回值的型態都必須和原來的父類別方法相同。

父類別：



子類別：

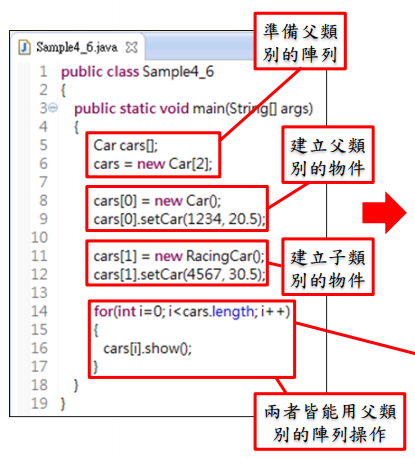


覆寫後會出現一個小三角形表示已覆寫。

**多型(polymorphism)：**

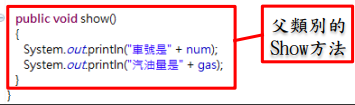
利用父類別的陣列來存取物件，再呼叫各個子類別中的方法，就能夠依照不同的情況來執行程式。

Ex：

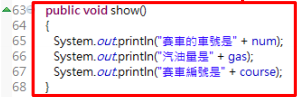


car[0]為car(父類別)物件，car[1]為RacingCar類別(子類別)物件。而兩者都是使用show方法顯示，但由於父類別以及子類別的show中程式不一樣，所以顯示出來的結果也不一樣。

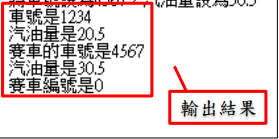
父類別show()：



子類別show()：



結果：

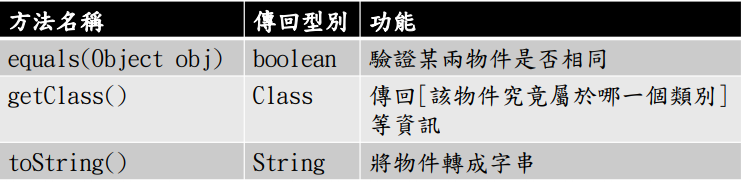


**Final：**

我們也可以在變數前加上final，來讓變數不在後面的子類別中被覆寫，也就是說變數不可修改。

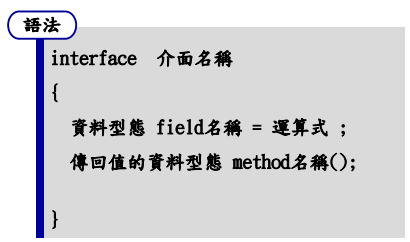
一個父類別可以延伸出多個子類別，子類別也能夠當別人的父類別，繼續往下延伸；但是一個子類別無法同時繼承多個父類別。

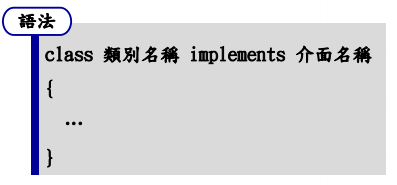
幾種常見的object方法：



**介面(interface)：**

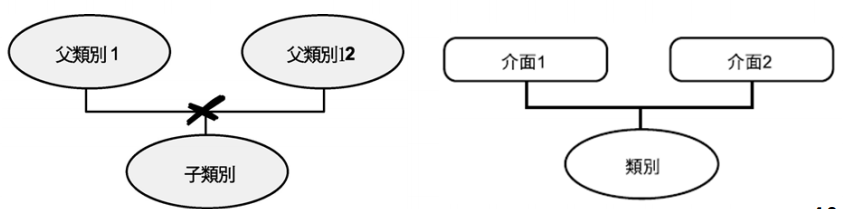
介面的宣告和一班類別不同的是在一開始要將class改成interface，其他部分則相同。



使用介面的方式是把介面和類別融合，當我們把介面納入到類別當中，這種情況稱為介面的實作(implementation)。

介面和類別不同的地方是，介面沒有建構式，成員也不會加上修飾子，且無法建立物件（無法使用new來建立新物件）。

上述有提到子類別無法繼承多個父類別，不過透過實作多個介面的方式也可以達到多重繼承的目的。



和一般類別相同的地方是，介面也可以使用延伸的方式來產生新介面，也是透過關鍵詞extends。

心得：

教的東西越來越多了，這次的繼承功能讓程式的功能整個複雜了起來，尤其是繼承中的多型功能，雖然目前還想不到我可以如何去利用他完成一項具體的功能，但看著PPT上的範例就感覺這項功能非常的強大。也謝謝老師的講解，讓我比較能夠理解其中的意義。