物件與類別

物件導向，一個接觸程式起可能早就聽到爛的名詞，而我也終於要開始接觸它了!

在物件導向程式觀念尚未成熟前，軟體開發是以程序導向為概念。程序導向是把大問題分割成小問題，若小問題還是複雜，就再繼續拆分直到電腦可以處理以及能夠撰寫為程序或是函式為止，並透過主程式連結呼叫，形成一系列對電腦下達的指令。這個概念下資料和方法是分開思考的，初期開發速度快，但當程式因需求變得龐大，資料和方法彼此相依的複雜度增加，後續的維護、擴充會變得較為困難。

物件導向(Object Oriented Programming，OOP)在1960年代被提出之後，許多語言都擁有物件導向的精神，如:C#、Python、C++、Java、PHP…等。它將資料跟方法一併思考包在一起成為物件，讓程式設計變得模組化，結構化後可以被不同流程重複使用，物件間彼此獨立又能互相呼叫。

什麼是物件(Object)?

萬物皆物件，以真實世界來說，你睜眼所見的人、山、海、車、房子等，都是物件。每個物件都有其對應的特徵和行為，物件的特徵以屬性(Properties)來描述，行為則以方法(Methods)來描述。

物件的屬性(Properties)

對於物件的一種靜態描述。

舉例來說，我是Cynthia(物件)，我有深棕色頭髮，小麥色皮膚，身高168公分以及其他很多可以描述我的特徵。頭髮、皮膚、身高這些都是關於我這個人擁有的屬性。有些屬性屬於某個物件獨有，有些則是大家都有，例如我的好鄰居區塔也有頭髮、皮膚、身高這些可以描述她的屬性。物件的屬性可以由物件自行改變，例如我發憤圖強決心減肥體重從200公斤減成50公斤~(好可怕XD)~。也可以由外界改變，例如我請髮型師幫我把深棕色頭髮變成綠色的。

物件的方法(Methods)

物件的行為，對物件的動態描述。

以手機為例，一支手機擁有的功能為照相、上網、通話…等，這些都是它可以執行的行為。物件之間可以透過這些行為(方法)來互動(傳遞參數)。以我們寫過的Console.WriteLine(“Hello World!”)為例，WriteLine是Console的其中一個方法，Hello World!是參數，這個方法要產生的行為是輸出並將游標換行。

什麼是類別(Class)?

類別就是建構某些相似物件的設計藍圖，用來定義這些類似物件的屬性與方法，而物件則是根據這個藍圖實體化(Instantiation)後的個體。

以車子這個類別來說，衍生的物件可能有腳踏車、三輪車、轎車…等。他們都具有輪子、方向盤、煞車裝置等屬性。也都具有前進、後退、轉彎、停止等方法。類別本身是一種分類(資料型別)，實際上不存在，不能執行方法。衍生出的實體物件例如腳踏車、三輪車、轎車等才能夠進行前進、轉彎等行為。

類別對物件做分門別類，形成一群類似物件的集合。由於同一類別的物件可能都具有相同的屬性及方法，在撰寫程式時要指定物件下達命令。例如三輪車要前進3格，轎車要後退1格，三輪車跟轎車是根據車子這個類別實體化的物件，前進跟後退是他們可以執行的方法，3格跟1格是給他們的資料(參數)，要寫成:

三輪車.前進(3);

轎車.後退(1);

再舉一個程式的例子

int A =10;

int B =5;

int是整數類別，我們可以透過宣告來衍生出A跟B，A跟B都是屬於整數，但他們彼此是不同物件，int不能執行相加這個方法，但是A跟B可以執行相加這個方法。

所以類別跟物件的關係為: