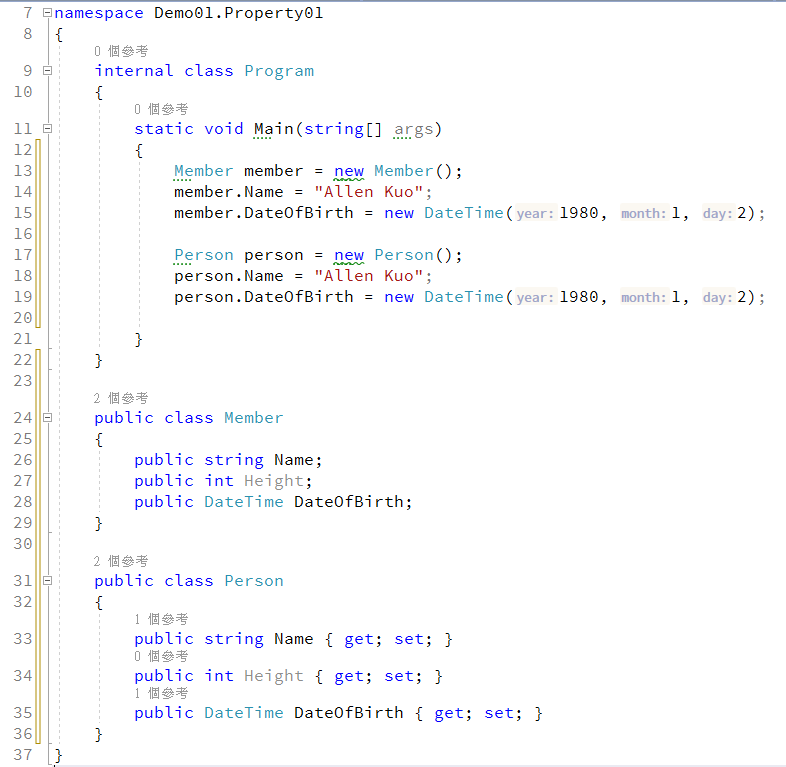
**屬性/欄位**

Member 宣告了三個 Field(欄位), Person 宣告了三個 Property(屬性)

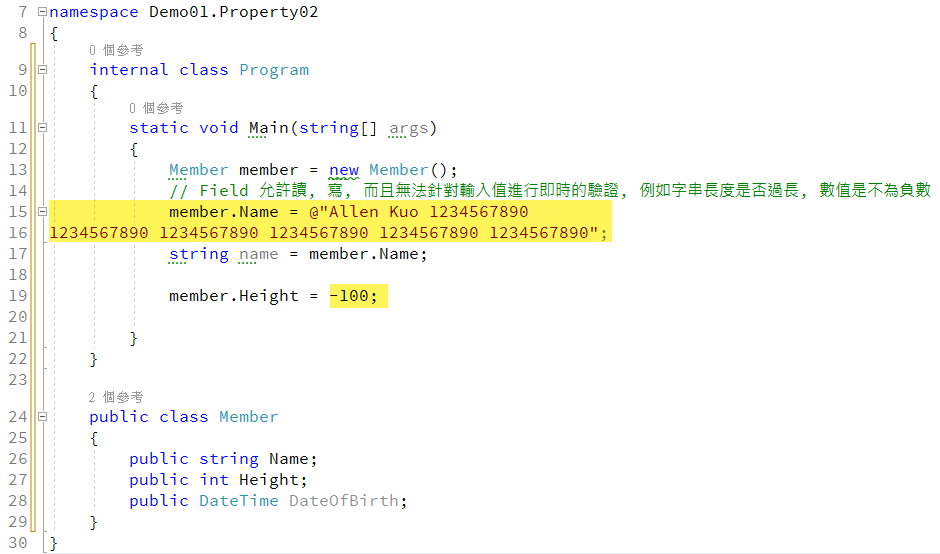
在 line 13 ~ 19 叫用時, 程式碼看起來是相同的, 看不出區別, 這也是初學者很容易發出的疑問

要回答這個問題之外, 我們先學習 Field, Property 各自如何宣告



# Field 的宣告

Field 允許讀, 寫, 而且無法針對輸入值進行即時的驗證, 例如字串長度是否過長, 數值是不為負數



# Property 的宣告

屬性宣告時, 區分 get, set

## 寫入屬性

叫用 set,

通常會將傳入的值存放在一個 private field 中,

在 set 裡, 有一個名稱為‘value’的隱含參數, 它的型別與 property 相同, 它表示外界傳入的值

在 set 中, 可以針對 value 進行欄位驗證, 若不合格就丟出異常 (Exception)

## 讀取屬性

時, 叫用get, 程式碼內容通常是讀取 private field並回傳



## 唯讀屬性

如果在 property 只撰寫 get, 可以達到禁止外界指派屬性值的功能, 將該屬性塑造成 readonly(唯讀), 如下例所示, 只要輸入生日, 便能得知 Age(年紀), 而 Age 的屬性值, 是經過計算而來的, 也不宜允許修改



## 唯寫屬性

相似地, 若想宣告 writeOnly(唯寫) 屬性, 可以在 property 裡只撰寫 set 即可

# Expression body definitions(運算式主體定義)

Ref : <https://learn.microsoft.com/zh-tw/dotnet/csharp/programming-guide/classes-and-structs/properties>

屬性存取子通常是由只會指派或傳回運算式結果的單行陳述式所組成。 您可以將這些屬性實作為運算式主體成員。 運算式主體定義包含 => 符號，後面接著要從屬性指派或擷取的運算式。

唯讀屬性可以將存取子實 get 作為運算式主體成員。 在此情況下，不會使用 get 存取子關鍵字和 return 關鍵字。 下列範例會將唯讀 Name 屬性實作為運算式主體成員。



C# 6 新增此屬性寫法, 用來簡單地撰寫出 readonly property

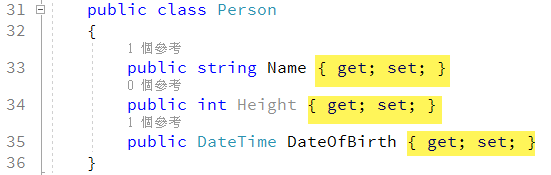
C# 7 允許在 set 也運用此寫法

# 自動實作屬性

Ref : <https://learn.microsoft.com/zh-tw/dotnet/csharp/programming-guide/classes-and-structs/properties>

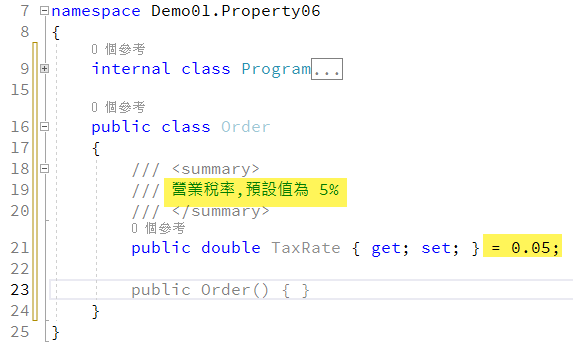
在某些情況下，屬性 get 和 set 存取子只會指派值，或從備份欄位擷取值，而不需要包含任何額外的邏輯。 藉由使用自動實作屬性，您可以簡化程式碼，同時讓 C# 編譯器無障礙地為您提供支援欄位。

如果屬性同時具有 get 和 set (或 get 和 和 init) 存取子，則必須自動實作兩者。 您可以使用 get 和 set 關鍵字，但不提供任何實作，來定義自動實作屬性。 下列範例會重複上一個範例，不同之處在於 Name 和 Price 為自動實作屬性。



## 屬性初始化器

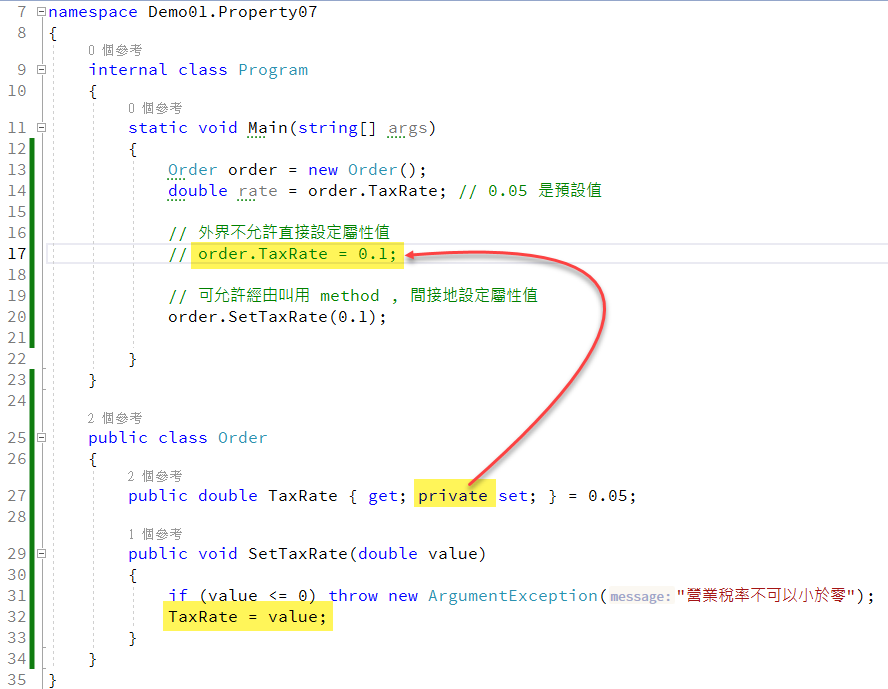
C# 6 支援自動屬性的初始化器, 允許簡單地給予屬性初始值, 稍後仍允許修改屬性值



## Get /set 的可訪問性

若標註private, 表示只能在類別自己內部 get/set 此屬性值, 外界無法叫用

另外可以標註 internal,沒有標註表示 public



## rivate set 與 唯讀屬性有什麼差異?

如下圖所示, Name設計成 readonly, Email設計成 private set,只允許類別裡叫用set; 有什麼區別呢?

* Line 24,25, 在class裡, 可以設定Name, Email,即使 Name 是readonly
* Line 31,32, 但在static member中, 就只能叫用Email, 無法設定Name屬性了



## Init Setter

C# 9 (.NET 5) 新增 init setter

Ref : <https://learn.microsoft.com/zh-tw/dotnet/csharp/language-reference/proposals/csharp-9.0/init>

這個宣告, 可以讓屬性只允許在類別的建構函數才能設定屬性值, 之後就無法再修改



# Field / Property的區別

類別的這兩種 member(成員)由於外界叫用方法極為相似, 所以初學者經常分不清楚要用哪一種

## 兩者的差異

* Field 一定是可以讀/寫; 屬性則可以視需要設計成 readonly/writeonly ,也可以做更仔細的設定,限制是否在類別內才能設定或者只能在建構函數才能設定
* 由於property可以在 set 裡撰寫程式, 因此可以達到驗證的功能, 避免外界指定不合宜的屬性值

以上是兩個比較大的差異

## 何時要應用Field/Property

* public 成員, 建議使用 property, 可以比較能保護傳入值是否合理
* Field 則通常宣告成private , 方便在類別裡被叫用, 但外界無法存取它

不少其他套件在針對類別做自動存取時, 都只針對 property, 因此就算您宣告 public field, 那些套件也無法正常運行; 因此建議用上述兩個原則: public 成員 用 property; private 成員則用 field