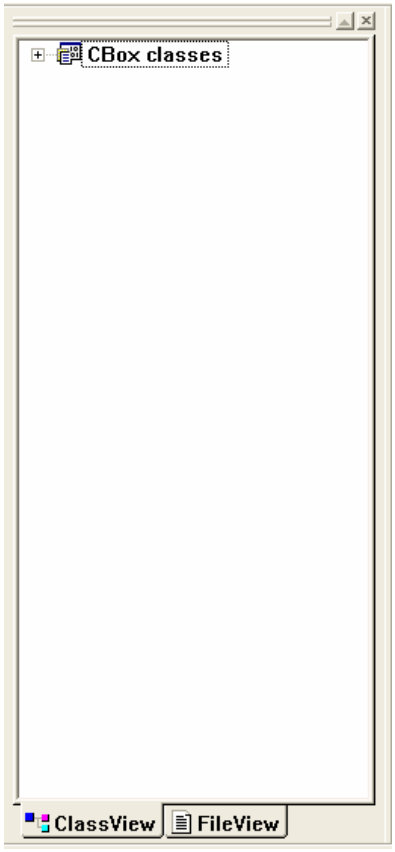
目的

此篇文章主要是教導如何使用 Visual C++ 所提供的工具來建立和維護類別，更詳細的的內容可以參考「Visual C++ 6 教學手冊，蔡明志譯」

範例程式

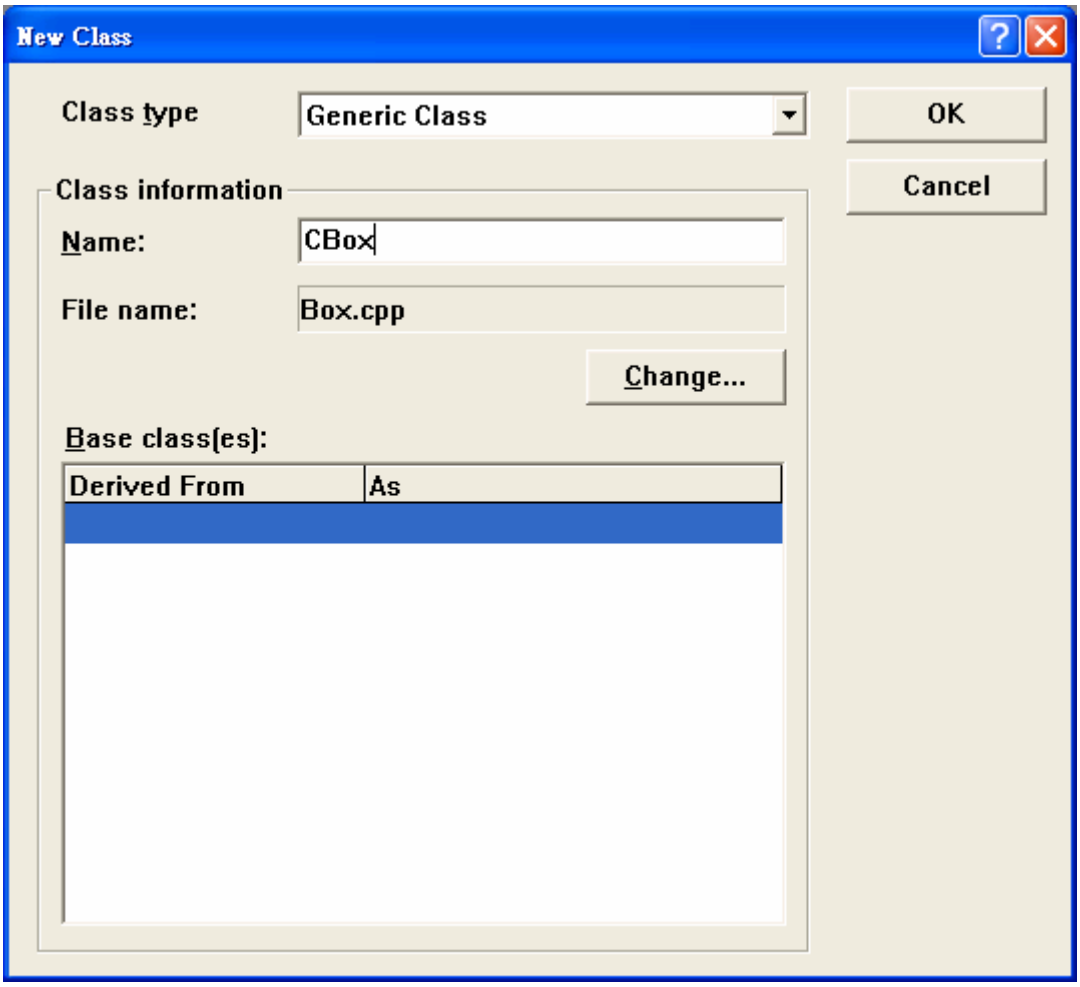
1. 建立一個「Console Application」的新專案，命名爲「CBox」
   建立完之後，會出現下方的視窗：



Workspace 視窗
- ClassView 頁籤：
  提供專案所有類別的瀏覽
- FileView 頁籤：
  顯示專案包含的所有檔案

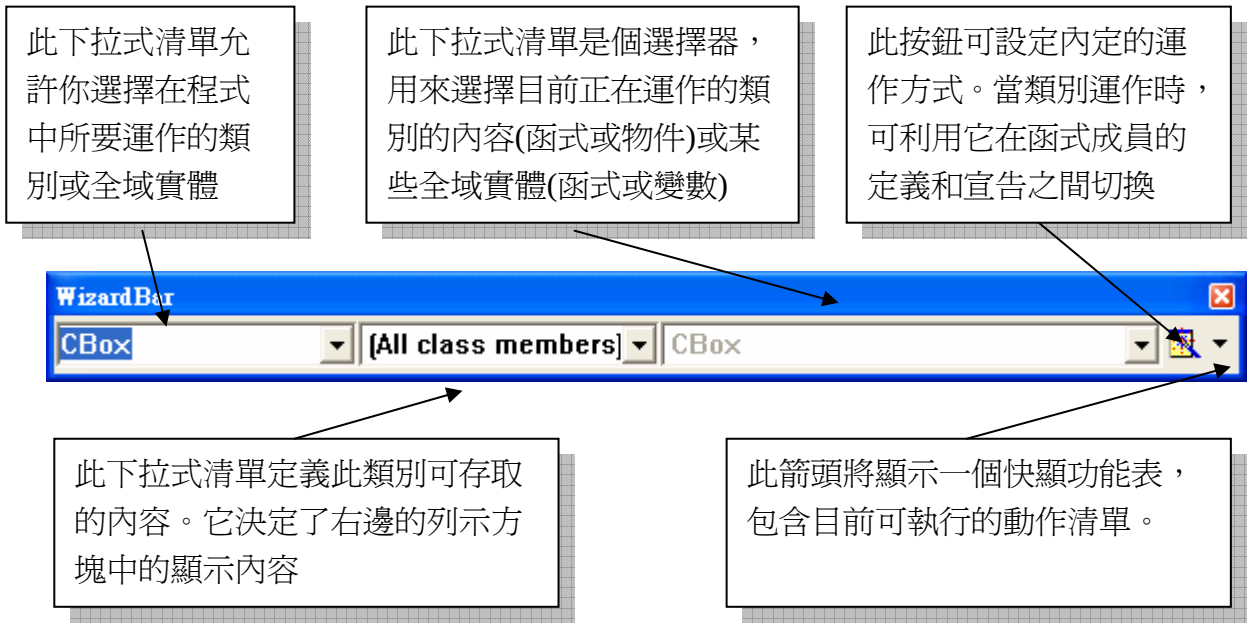
2. 在 ClassView 裡的 CBox classes 的位何地方按「右鍵」，選擇「New Class…」
   1) 在 Class Information 區域中的欄位 Name，輸入「CBox」，而 File name 會自動得建立「Box.cpp」。若考慮使用其它名稱時，可以按「Change…」按鈕來修改檔案名稱。
   2) 類別定義的程式碼會儲存在副檔名.h 的檔案中
   3) 函式定義的程式碼會儲存在副檔名.cpp 的檔案中

Note：
此篇文章類別及類別的資料成員之命名，依照 MFC 的慣例
- 所有類別名稱之前必需加上前置字 **C**
- 類別的資料成員之前要加上前置字 **m_**



按下「Ok」之後將發生以下幾件事

- Box.h 檔案被建立，內容包含 CBox 類別的主要定義。
- Box.cpp 檔案也被建立，內容包含 CBox 類別的主要實作程式。
- Wizard Bar 被啓動。



3. 在 ClassView 的 CBox classes 左邊按 + 號，將顯示出樹狀結構。由樹狀結構可以看到目前專案已定義了 CBox，而專案中的所有類別都將顯示在樹狀結構中。在樹狀結構的類別名稱上按兩下滑鼠左鍵或按 Wizard Bar，即可看到類別定義的原始碼。

```
// Box.h: interface for the CBox class.
//
//////////////////////////////////////////////////////////////////////

#if !defined(AFX_BOX_H__87CA4768_31DD_4FC1_A682_73544304B6F9__INCLUDED_)
#define AFX_BOX_H__87CA4768_31DD_4FC1_A682_73544304B6F9__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class CBox
{
public:
    CBox();
    virtual ~CBox();

};

#endif
// !defined(AFX_BOX_H__87CA4768_31DD_4FC1_A682_73544304B6F9__INCLUDED_)
```

程式的結構：

- 註解

```
// Box.h: interface for the CBox class.
//
//////////////////////////////////////////////////////////////////////
```

- 避免檔案被引入超過一次的編譯器假指令
  Visual C++會自動加入這種措施，這些代號的個數和文字每個機器並不相同，這是爲了 100%確定每個代碼都是獨一無二的。

```
#if !defined(AFX_BOX_H__87CA4768_31DD_4FC1_A682_73544304B6F9__INCLUDED_)
#define AFX_BOX_H__87CA4768_31DD_4FC1_A682_73544304B6F9__INCLUDED_
...
#endif
/ !defined(AFX_BOX_H__87CA4768_31DD_4FC1_A682_73544304B6F9__INCLUDED_)
```

- 第二組#if-#endif 假指令也是類似的功能。假如編譯器夠新，則#pragma once 假指令將會被執行，這表示不必再次打開檔案，開過一次就夠了，這比編譯器每次碰到#include 都要開檔來測試代碼的速度快多了。

```
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
```

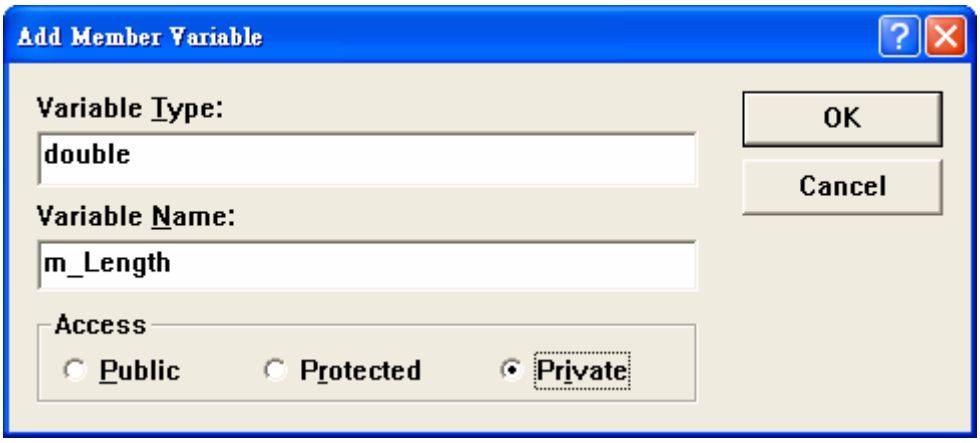
- 類別定義的大綱，包含內定類別建構子和解構子

```
class CBox
{
public:
    CBox();
    virtual ~CBox();

};
```

4. 加入資料成員 (加入 private 資料成員 m_Length, m_Breadth, m_Height)

1) 在 ClassView 的 CBox 按右鍵，選擇「Add Member Variable…」



同樣的方式，加入另位兩個資料成員。當然，你也可以直接在編輯視窗中，以手動的方式輸入資料成員。

```
// Box.h: interface for the CBox class.
//
//////////////////////////////////////////////////////////////////////

#if !defined(AFX_BOX_H__87CA4768_31DD_4FC1_A682_73544304B6F9__INCLUDED_)
#define AFX_BOX_H__87CA4768_31DD_4FC1_A682_73544304B6F9__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class CBox
{
public:
    CBox();
    virtual ~CBox();

private:
    double m_Breadth;
    double m_Height;
    double m_Length;
};

#endif // !defined(AFX_BOX_H__87CA4768_31DD_4FC1_A682_73544304B6F9__INCLUD
```

5. 定義建構子 (Constructor)

1) 打開 Box.h，將類別定義所宣告的建構子改爲：

```
CBox(double lv = 1.0, double bv = 1.0, double hv = 1.0);
```

2) 更改建構子的實作碼，打開 Box.cpp，將建構子的定義改爲：

```
CBox::CBox(double lv, double bv, double hv)
{
    lv = lv <= 0.0 ? 1.0 : lv;      // Ensure positive
    bv = bv <= 0.0 ? 1.0 : bv;      // dimensions for
    hv = hv <= 0.0 ? 1.0 : hv;      // the object

    m_Length  = lv > bv ? lv : bv;  // Ensure that
    m_Breadth = bv < lv ? bv : lv;  // length >= breadth
    m_Height  = hv;
}
```

3) 存檔

絕竅：
從.cpp 檔的函式「定義」切換到.h 檔的函式「宣告」

- 在編輯視窗的類別定義中，將游標指到函式宣告那一行，然後在 Wizard Bar 裡按下  圖示，視窗內容會切換到函式定義。反過來也一樣。
- 在 ClassView 視窗中，在要存取的函式名稱上按右鍵，然後按「Go to Definition」，游標會切到相關的.cpp 的函式標題；若再按「Go to Declaration」則又切換回相關的.h 的函式標題。

6. 加入函式成員 (加入 4 個成員函式)
   - double GetHeight() const
   - double GetBreadth() const
   - double GetLength() const
   - double Volume() cosnt

   在 ClassView 的 CBox 按右鍵，選擇「Add Member Function…」

按「OK」之後，函式宣告會被加入 Box.h 檔的類別定義中，同時函式定義的骨幹則被加入 Box.cpp 中。此時可以切換到類別所在的.cpp 檔，撰寫函式的主體及註解。

如果該函式成員爲 inline 的，則不需要透過以上的方式來加入，而直接在類別定義中加上宣告及主體。