

CH2

# Visual C# 程式的組成與語法規則

# 目錄

- 2-1 Visual C# 程式基本架構
- 2-2 Visual C# 程式語法規則說明
- 2-3 Visual C# 輸入與輸出
- 2-4 程式設計演算法的介紹

## 2-1 Visual C# 程式基本架構

- 方案(solution) 可以包含多個專案(project) , Visual C# 提供「方案總管」視窗，用於顯示目前專案所使用的程式碼、專案組態檔、系統函式庫與專案所需各類檔案等資訊
- 下圖為前一章「簡單加法計算機」的「方案總管」視窗，「參考」為系統函式庫，「App.config」為專案組態檔，「Form1.cs」為程式碼與目前專案所使用的元件。

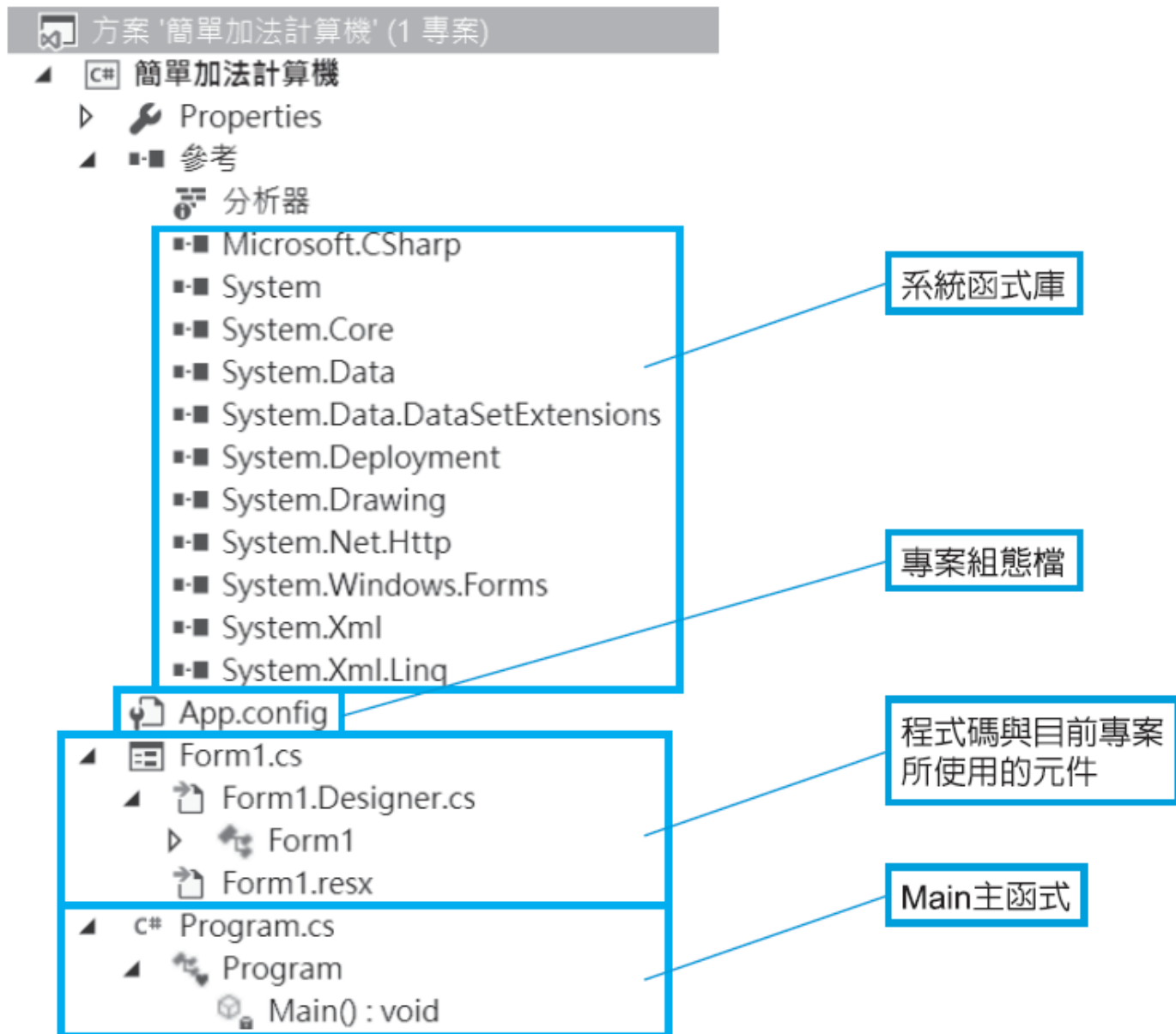
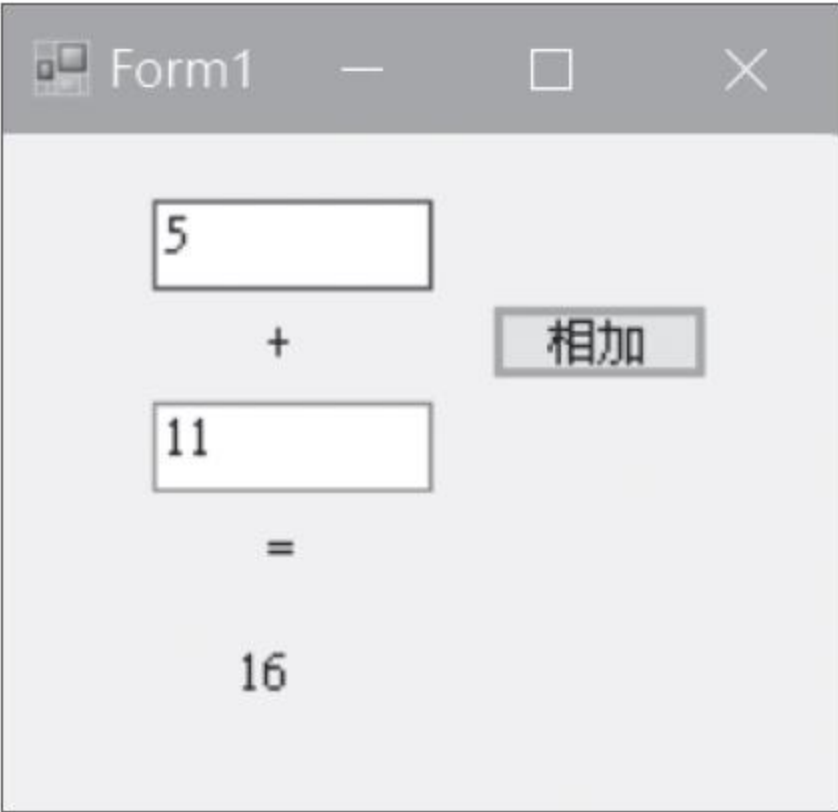



圖 2-1 「簡單加法計算機」的「方案總管」視窗

# 2-1-1 Visual C# 教學常用專案格式

- 在教學上，經常使用Visual C# 開發視窗化程式與命令提示字元程式，資料的輸入與輸出都使用視窗元件，稱作視窗化程式
- 在「命令提示字元」視窗使用鍵盤輸入資料，計算結果直接顯示在螢幕上，稱作命令提示字元程式。視窗化程式與命令提示字元程式的範例如下。

表 2-1 視窗化程式與命令提示字元程式

視窗化程式	命令提示字元程式
 <p>使用文字方塊輸入數字 5 與 11，相加後獲得 16，顯示在下方標籤上。</p>	 <p>在命令提示字元使用鍵盤輸入數字 5 與 11，在螢幕顯示相加結果為「16」。</p>

## 2-1-2 Visual C# 視窗化程式的程式結構

- 在Visual Studio 中使用「檔案→新增→專案」
- ❶選擇「Visual C#」
- ❷選擇「Windows 傳統桌面」
- ❸選擇「Windows Form App」
- ❹輸入專案的名稱
- ❺點選「瀏覽(B)…」可以選擇專案資料夾
- ❻點選「確定」，就會新增Visual C# 視窗化程式專案。

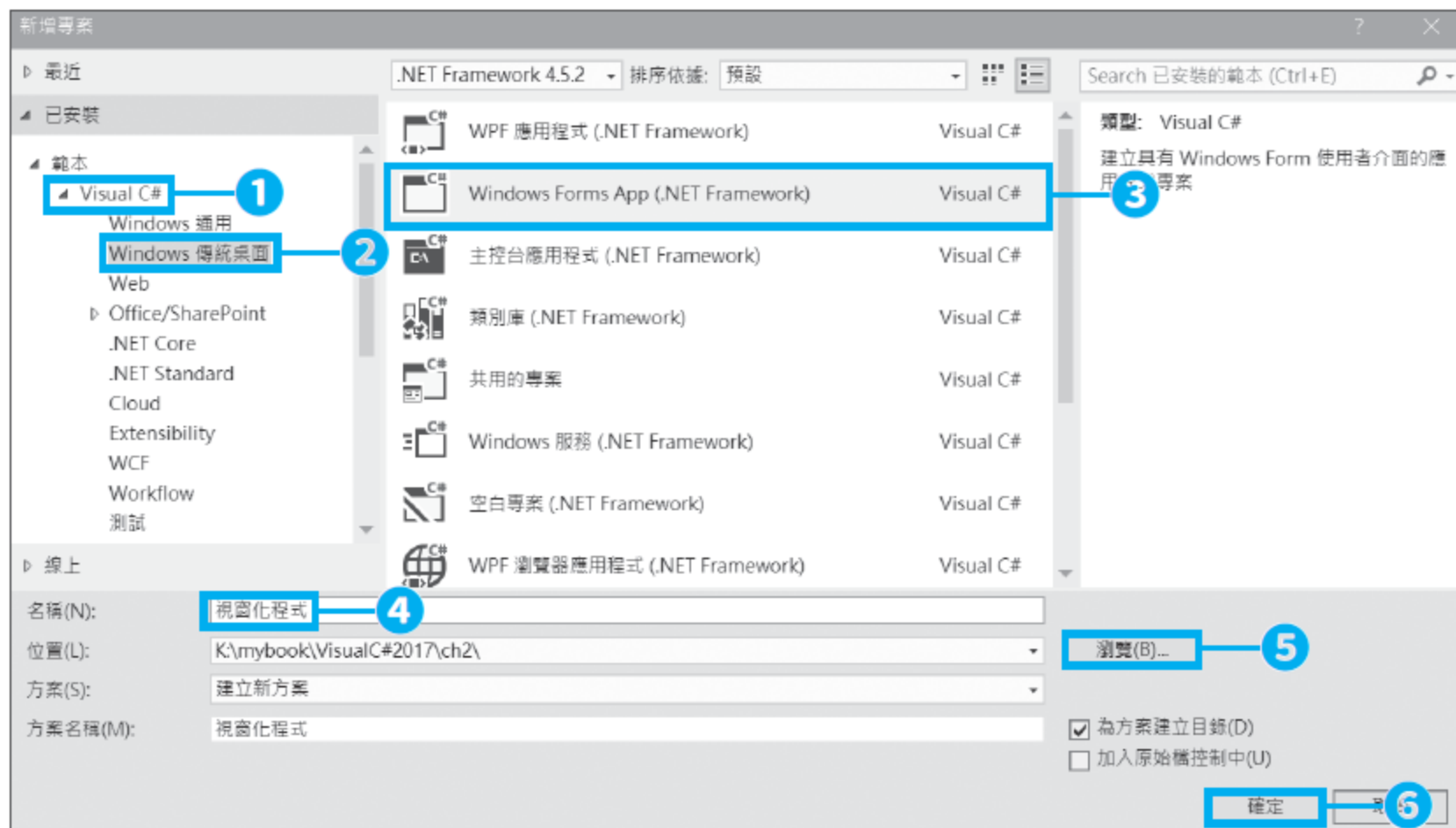


圖 2-2 新增 Visual C# 視窗化程式專案



- 以下為新增的視窗化程式專案，在Form1 點選滑鼠右鍵，接著選擇「檢視程式碼」。

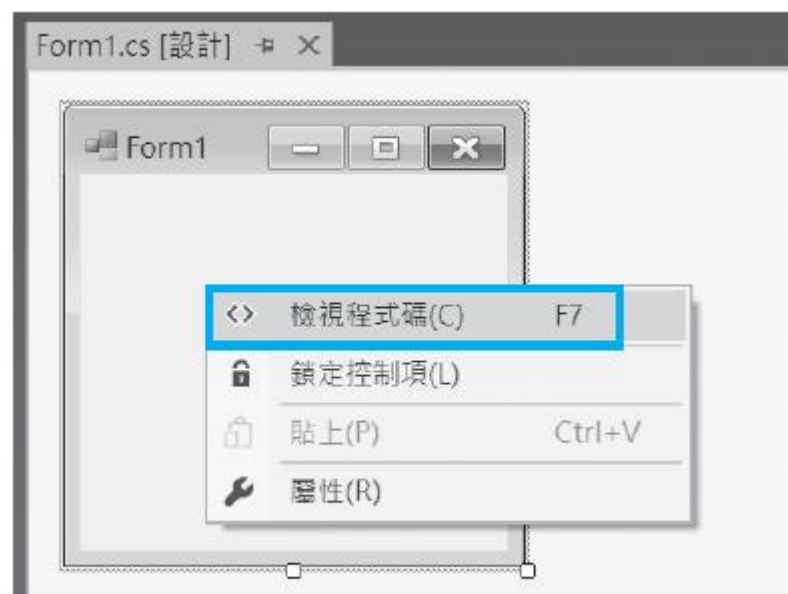


圖 2-3 選擇「檢視程式碼」

行號

Program.cs程式

```
01 using System;
02 using System.Collections.Generic;
03 using System.Linq;
04 using System.Threading.Tasks;
05 using System.Windows.Forms;
06
07 namespace 視窗化程式
08 {
09     static class Program
10     {
11         /// <summary>
12         /// 應用程式的主要進入點。
13         /// </summary>
14         [STAThread]
15         static void Main()
16         {
17             Application.EnableVisualStyles();
18             Application.SetCompatibleTextRenderingDefault(false);
19             Application.Run(new Form1());
20         }
21     }
22 }
```

行號	Form1.cs程式
01	using System;
02	using System.Collections.Generic;
03	using System.ComponentModel;
04	using System.Data;
05	using System.Drawing;
06	using System.Linq;
07	using System.Text;
08	using System.Threading.Tasks;
09	using System.Windows.Forms;
10	
11	namespace 視窗化程式
12	{
13	public partial class Form1 : Form
14	{
15	public Form1()
16	{
17	InitializeComponent();
18	}
19	}
20	}

## 2-1-3 Visual C# 命令提示字元程式( 主控台應用程式) 的程式結構

- 在Visual Studio 中使用「檔案→新增→專案」
- ❶選擇「Visual C#」
- ❷選擇「Windows 傳統桌面」
- ❸選擇「主控台應用程式」
- ❹輸入專案的名稱
- ❺點選「瀏覽(B)…」可以選擇專案資料夾
- ❻點選「確定」，就會新增Visual C# 命令提示字元程式( 主控台應用程式) 專案。

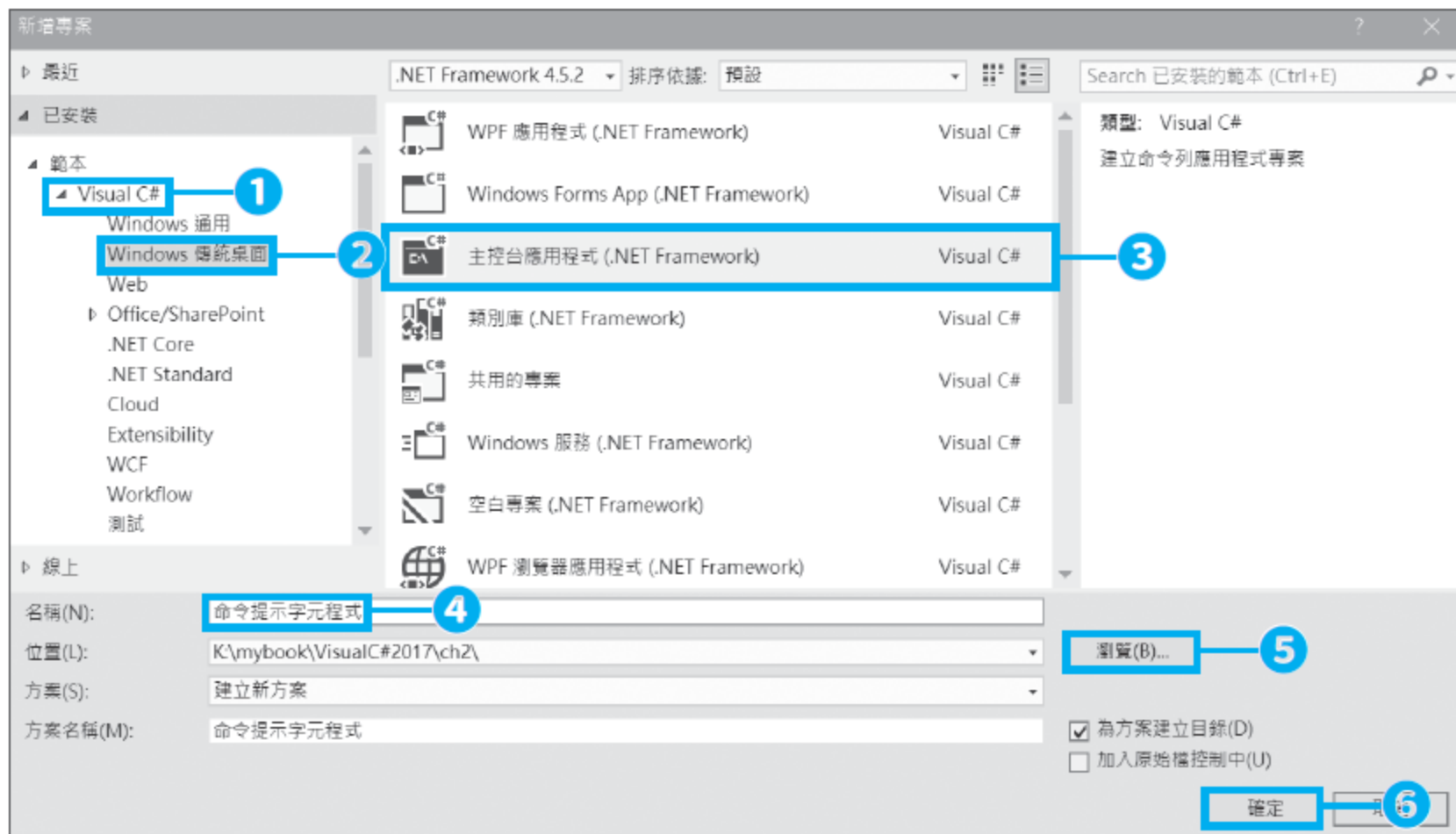


圖 2-4 新增 Visual C# 命令提示字元程式 (主控台應用程式) 專案

行號	Program.cs程式
01	using System;
02	using System.Collections.Generic;
03	using System.Linq;
04	using System.Text;
05	using System.Threading.Tasks;
06	
07	namespace 命令提示字元程式
08	{
09	class Program
10	{
11	static void Main(string[] args)
12	{
13	}
14	}
15	}

## 2-2 Visual C# 程式語法規則說明

- 程式區塊

- C# 以類別(class) 為執行單位，類別內可以定義資料與函式
- 函式內以程式區塊可以是一行到多行的程式碼、條件判斷結構(if) 與迴圈結構(for 與while) 等

- 程式區塊內可以包含程式區塊，在本書之後章節會介紹這些程式區塊，表示區塊的大括號({}) 需成對出現。

```
if (條件判斷){
```

```
}
```



- 程式區塊內程式碼需要進行縮行，這樣才容易辨別程式區塊的範圍，Visual C#會自動縮行4個半形字元的距離，如下。

```
if (條件判斷){  
    條件判斷結果為true，需要執行的程式;  
}
```

- 程式區塊內可以包含程式區塊

- 例如：while迴圈包含if條件判斷，while迴圈內的程式需縮行4個半形字元，if條件判斷內的程式也需要縮行4個字元，如下。

```
while (迴圈條件判斷){  
    if (條件判斷){  
        條件判斷結果為true，需要執行的程式;  
    }  
}
```

- 程式區塊內的程式

- 程式區塊內可以由一行或多行程式組合起來，一行程式相當於一句話，多行程式相當於一段話，一句話或一段話可以敘述一個想法，一行或多行程式也可以完成一個功能

```
int num, price, total;
```

```
num = 5;
```

```
price = 100;
```

```
total = num * price;
```

- 多行程式可以寫成一行

- 可以將多行程式寫在同一行，以半形分號「;」為間隔符號，如下。

num = 5;price=100;total = num \* price;

- 註解

- Visual C#的註解使用兩個半形斜線「//」，兩個半形斜線「//」後所串接的文字到該行結束都被視為註解，會被自動忽略；也可以使用「/\*」與「\*/」之間的文字都視為註解，可以跨好幾行。

/\*

以下程式碼計算總金額

\*/

Dim num, price, total As Integer //宣告變數

num = 5

price = 100

total = num \* price //計算總金額

## 2-3 Visual C# 輸入與輸出

- 2-3-1 使用Console 進行輸入與輸出
  - 表2-2函式庫System.Console的重要函式

函式	說明
Console.Read()	讀取一個字元，回傳該字元。
Console.ReadLine()	讀取一整行的字串，直到使用者按下Enter鍵回傳字串。
Console.Write(string)	將string顯示在螢幕上。
Console.WriteLine(string)	將string顯示在螢幕上，在結尾加上換行符號。

- 表2-3 以符號表示輸出時的控制字元

符號	說明
\t	定位字元Tab
\r	換行字元Carriage Return
\n	換行字元Line Feed
\'	單引號
\"	雙引號
\\	反斜線

# 範例2-1 簡單加法計算機Console版

- 寫一個程式允許使用者從鍵盤輸入兩個數字，會自動顯示相加結果到螢幕上。
- (本專案所在資料夾：ch2\簡單加法計算機Console版)
- 程式結果預覽
  - 輸入5，輸入11，顯示「相加結果為16」。



```
K:\mybook\VisualC#2017\ch2\簡單加法計算機Console版\  
5  
11  
相加結果為16
```



## 行號 程式碼

```
01 using System;
02 using System.Collections.Generic;
03 using System.Linq;
04 using System.Text;
05 using System.Threading.Tasks;
06
07 namespace 簡單加法計算機Console版
08 {
09     class Program
10     {
11         static void Main(string[] args)
12         {
13             int a, b, c;
14             a = Convert.ToInt32(Console.ReadLine());
15             b = Convert.ToInt32(Console.ReadLine());
16             c = a + b;
17             Console.WriteLine("相加結果為"+c);
18             Console.ReadLine();
19         }
20     }
21 }
```

# 程式解說

- 第11到19行：函式Main的範圍，函式Main為命令提示字元程式第一個執行的函式。
- 第13行：宣告變數a、變數b 與變數c 為整數，下一章會詳細介紹變數的概念。

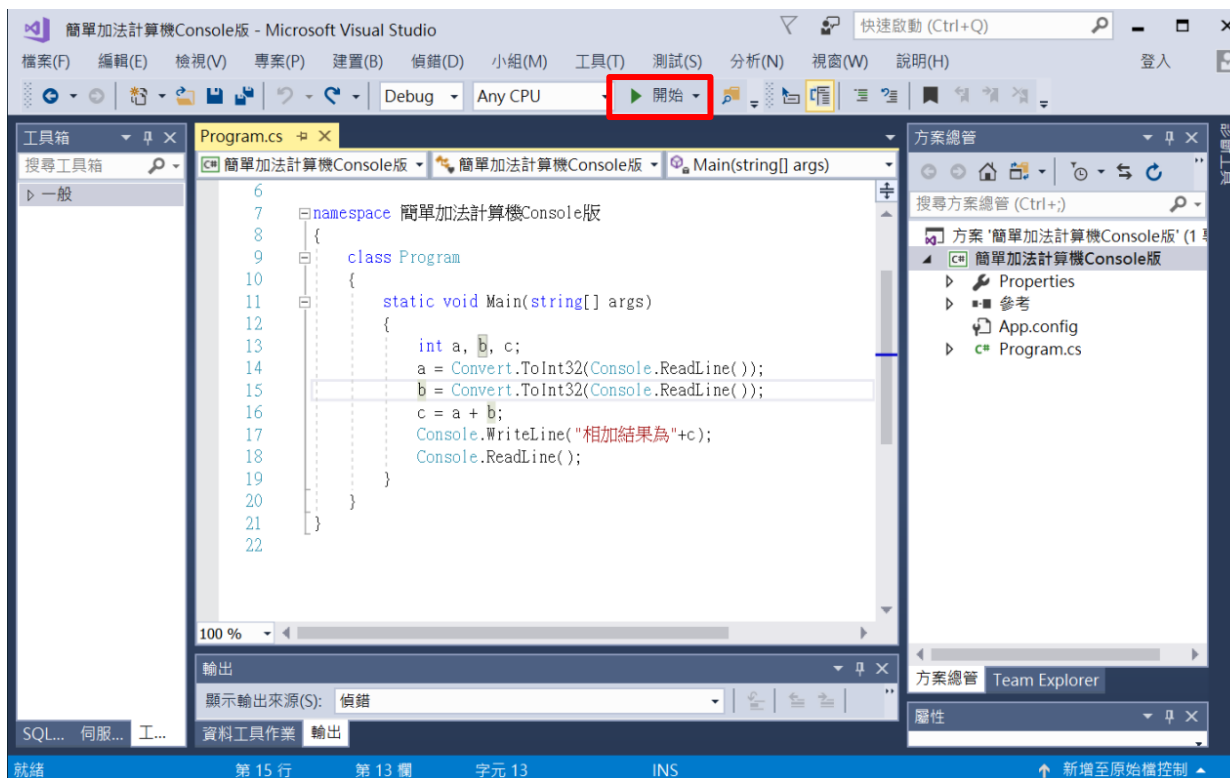
- 第14 行：使用`Console.ReadLine` 從鍵盤輸入字串直到輸入Enter 鍵為止，利用函式`Convert.ToInt32` 將輸入字串轉成數字，將數字儲存到變數a。
- 第15 行：使用`Console.ReadLine` 從鍵盤輸入字串直到輸入Enter 鍵為止，利用函式`Convert.ToInt32` 將輸入字串轉成數字，將數字儲存到變數b。

- 第16行：將變數a 與變數b 相加的結果儲存到變數c。
- 第17 行：使用Console.WriteLine 輸出字串「相加結果為」串接變數c，其中「+」為字串串接符號。

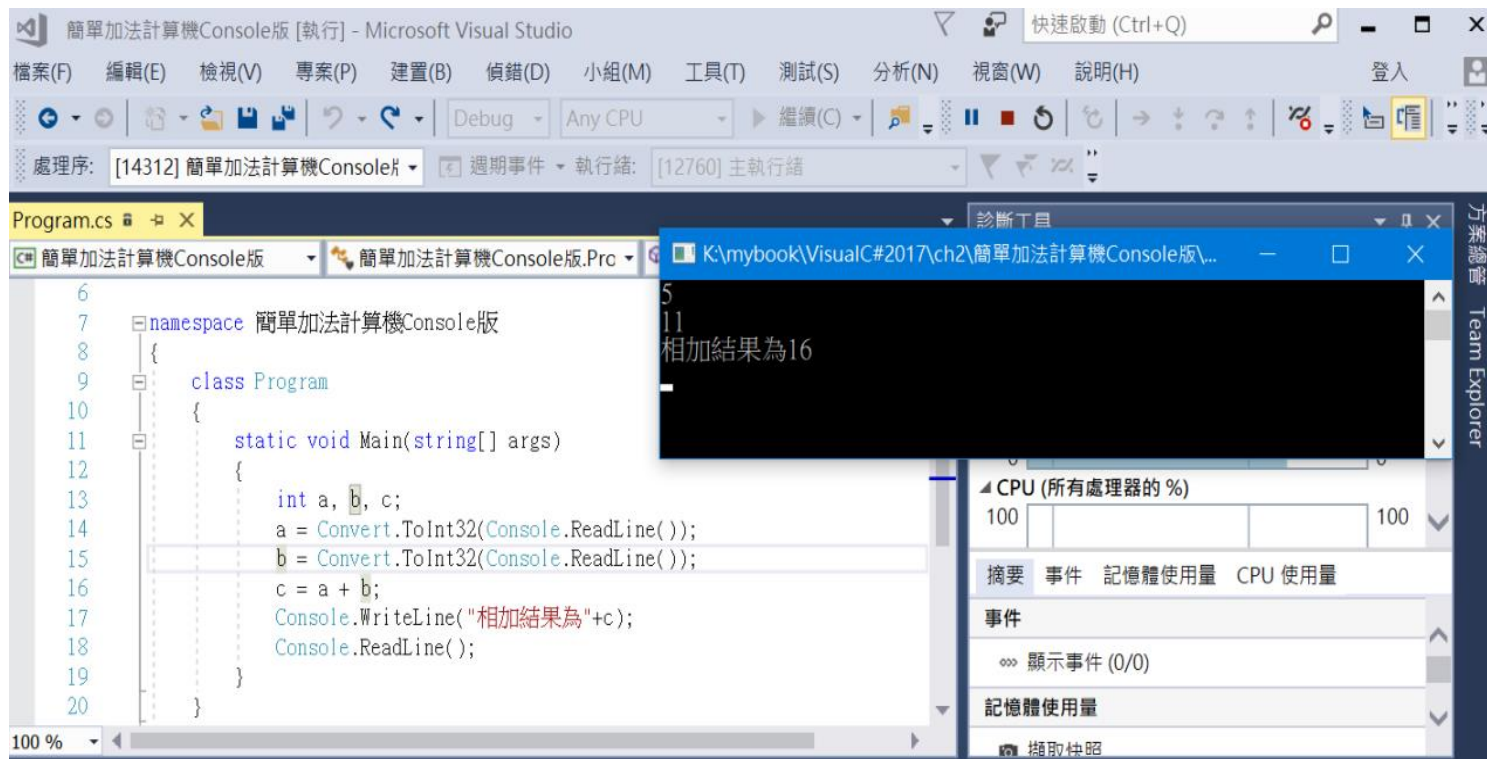
- 第18 行：為了讓程式可以停止看到結果，本程式加上 `Console.ReadLine` 等待鍵盤輸入字串，直接按下Enter 鍵程式就會結束，可以將此行刪除，但執行時會看不到結果。

# 執行程式

- 點選「開始」會將程式碼編譯成執行檔，並且會自動執行程式。



- 在程式內輸入兩個數字，例如5與11，會產生相加的結果。



- 在專案資料夾下的資料夾bin的資料夾Debug會產生執行檔「簡單加法計算機Console版.exe」，也可以點選該檔案執行程式。





# 2-4 程式設計演算法的介紹

- 輸入

- 演算法可能有輸入，也可能沒有輸入，如果有輸入，需要明確的說明輸入的個數與每個輸入所表示的意義。

- 輸出

- 演算法至少一個以上的輸出，表示演算法執行後的結果。

- 明確性


- 所有演算法步驟都需要明確，演算法步驟不能有兩種以上的解釋，才能依照演算法轉換成程式碼。

- 正確性

- 演算法要能正確完成所需功能或解決問題，錯誤的演算法就需要修正。

- 有限性

- 電腦需要在有限步驟內執行程式，若演算法無法在有限步驟內完成，演算法就無法終止，轉換成的程式也無法執行完畢，無法獲得結果。

 表 2-4 流程圖圖示

流程圖圖示	意義
	程式流程的線，表示程式的處理順序。
	條件選擇，在菱形內寫入條件判斷。
	程式的敘述區塊，寫出所需完成的功能。
	開始或結束，看到此圖表示程式的開始或結束。
	程式所需的輸入與輸出。
	連接點，當流程圖過長可以使用連接點將過長流程圖切割成多個流程圖組合起來，也可以避免流程線過長或交叉。

- 假設要判斷一個數字是奇數還是偶數，使用文字敘述、虛擬碼與流程圖表示演算法，如下。
- 1. 以文字敘述表示演算法
  - 若數字除以2 的餘數等於0，則數字為偶數，否則數字為奇數。
- 2. 以虛擬碼表示演算法

```
if num % 2 == 0
    print num 是偶數
else
    print num 是奇數
```

- 3. 以流程圖表示演算法

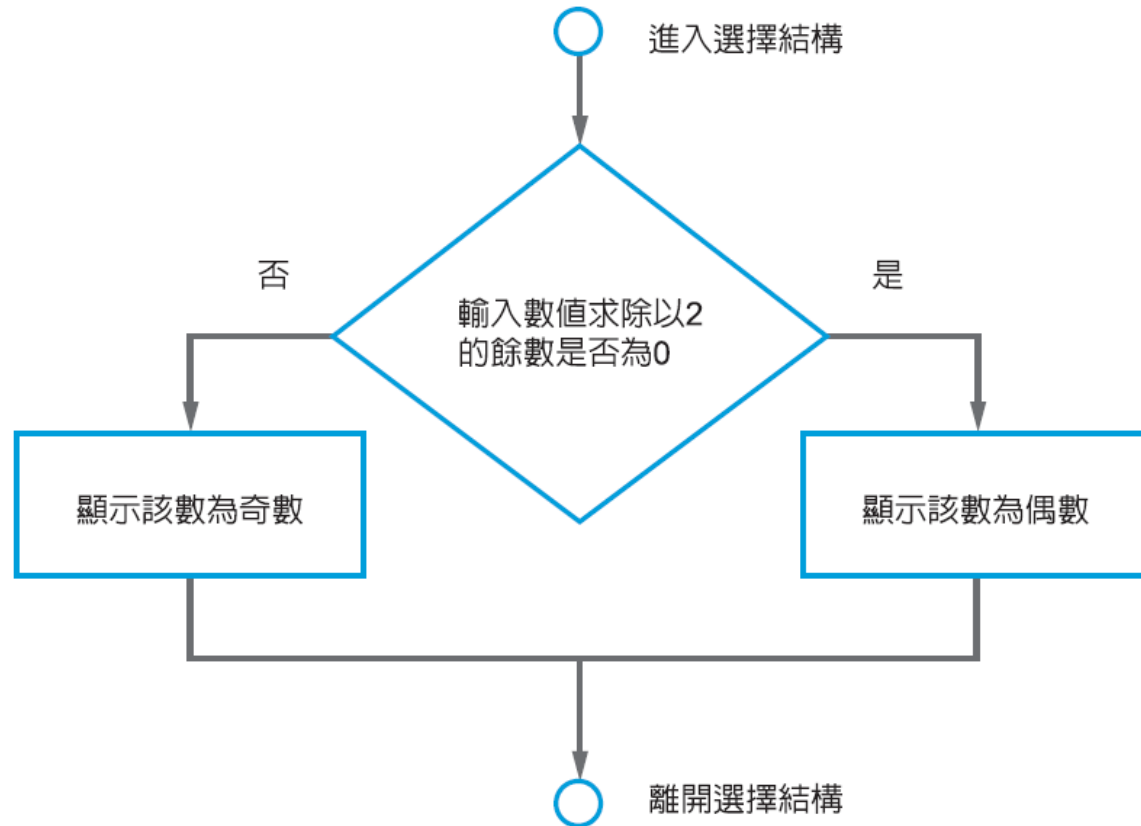


圖 2-9 以流程圖表示演算法