

程式設計-使用Visual Basic 2010




第5章 迴圈結構

迴圈結構



- 假設要撰寫程式能夠產生**1000**個訊息視窗，上面寫著「**Hello**」，若不使用迴圈結構，需寫**1000**個「**Msgbox("Hello")**」，如下表。

產生1000個訊息視窗，上面寫著「Hello」程式碼(1)

| | | |
|-----------------|--|------------------------|
| MsgBox("Hello") |  | 1000個Msgbox("Hello") |
| MsgBox("Hello") | | |
| MsgBox("Hello") | | |
| ... | | |
| MsgBox("Hello") | | |

迴圈結構



- 使用迴圈結構可以簡短程式碼達成相同功能，如下表。

產生1000個訊息視窗，上面寫著「Hello」 程式碼(2)

```
For k = 1 To 1000 Step 1
```

```
    MsgBox("Hello")
```

```
Next
```

迴圈結構

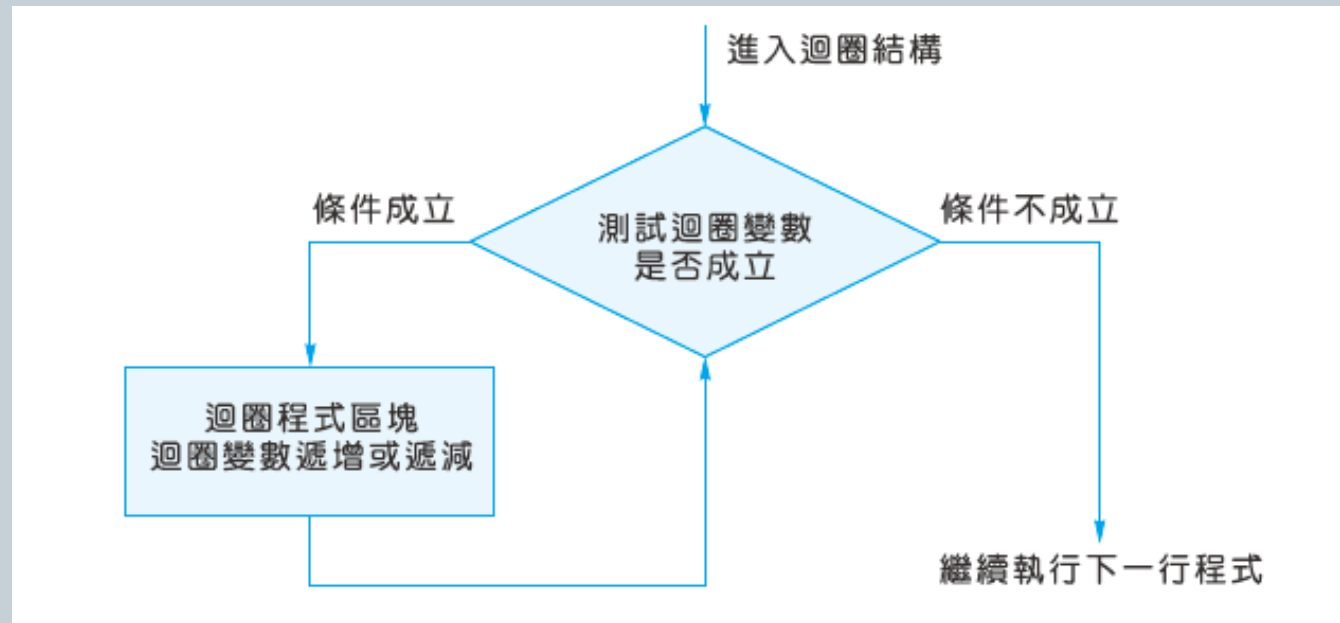


- 善用迴圈結構才能有效利用電腦的運算能力與簡化程式碼。
- 電腦每秒鐘可執行幾億次的指令，迴圈結構是重複執行程式的結構。
- 許多問題的解決都涉及迴圈結構的使用，例如：加總、排序、找最大值等。

迴圈結構

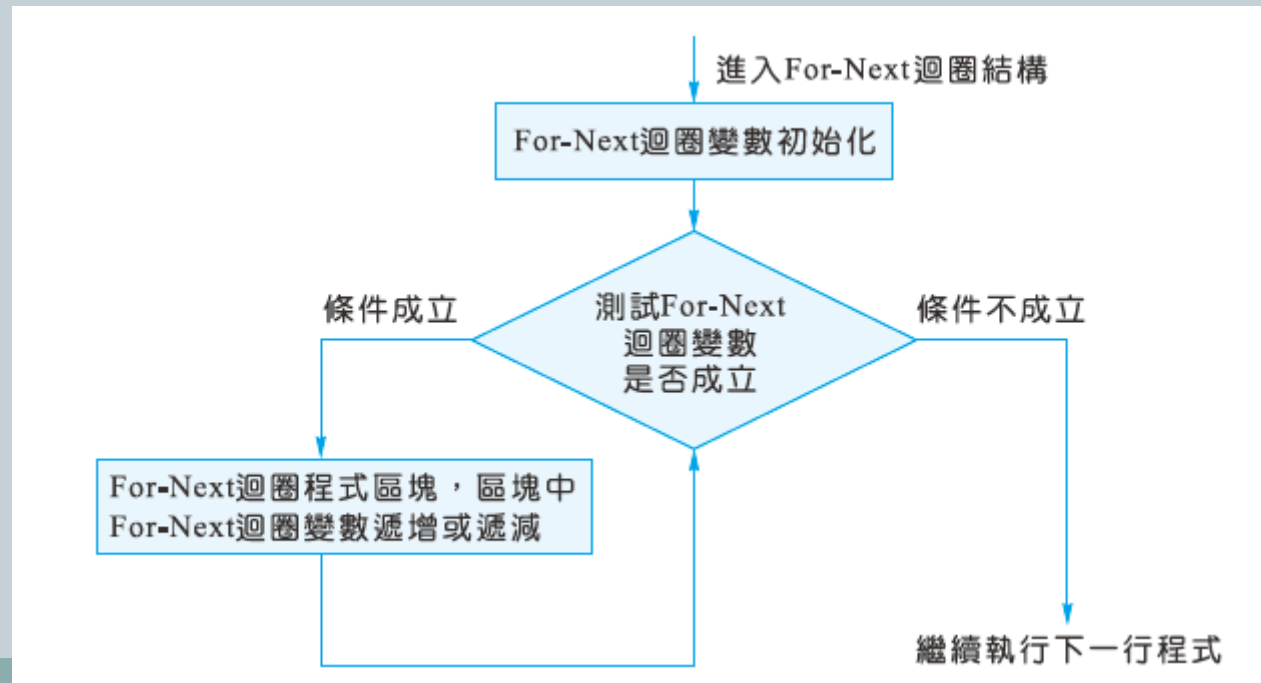


- 迴圈結構會進行迴圈變數的測試，若測試迴圈變數成立，則進行迴圈程式區塊，迴圈程式區塊執行結束後，迴圈變數進行遞增或遞減，會再次測試迴圈變數，若成立就繼續執行；不成立就跳出迴圈，執行下一行程式。
- 流程圖表示



迴圈結構—使用For-Next

- **For-Next**迴圈結構通常用於已知重複次數的程式，迴圈結構中指定迴圈變數的初始值、遞增或遞減值、終止值，迴圈變數將由初始值變化到終止值，每次依照遞增或遞減的值變化。
- 流程圖表示



迴圈結構—使用For-Next



- 程式語法

| For-Next程式語法 | 程式範例(印出1000個Hello) |
|--|---|
| For 迴圈變數 = 起始值 To 終止值 Step 遞增(減)值 重複的程式 Next | For k = 1 To 1000 Step 1 MsgBox("Hello") Next |

- 說明

- **For-Next**內迴圈變數由起始值變化到終止值，每重複執行一次程式，迴圈變數就會遞增(減)值，重複執行迴圈內程式，直到超過終止值後停止執行。

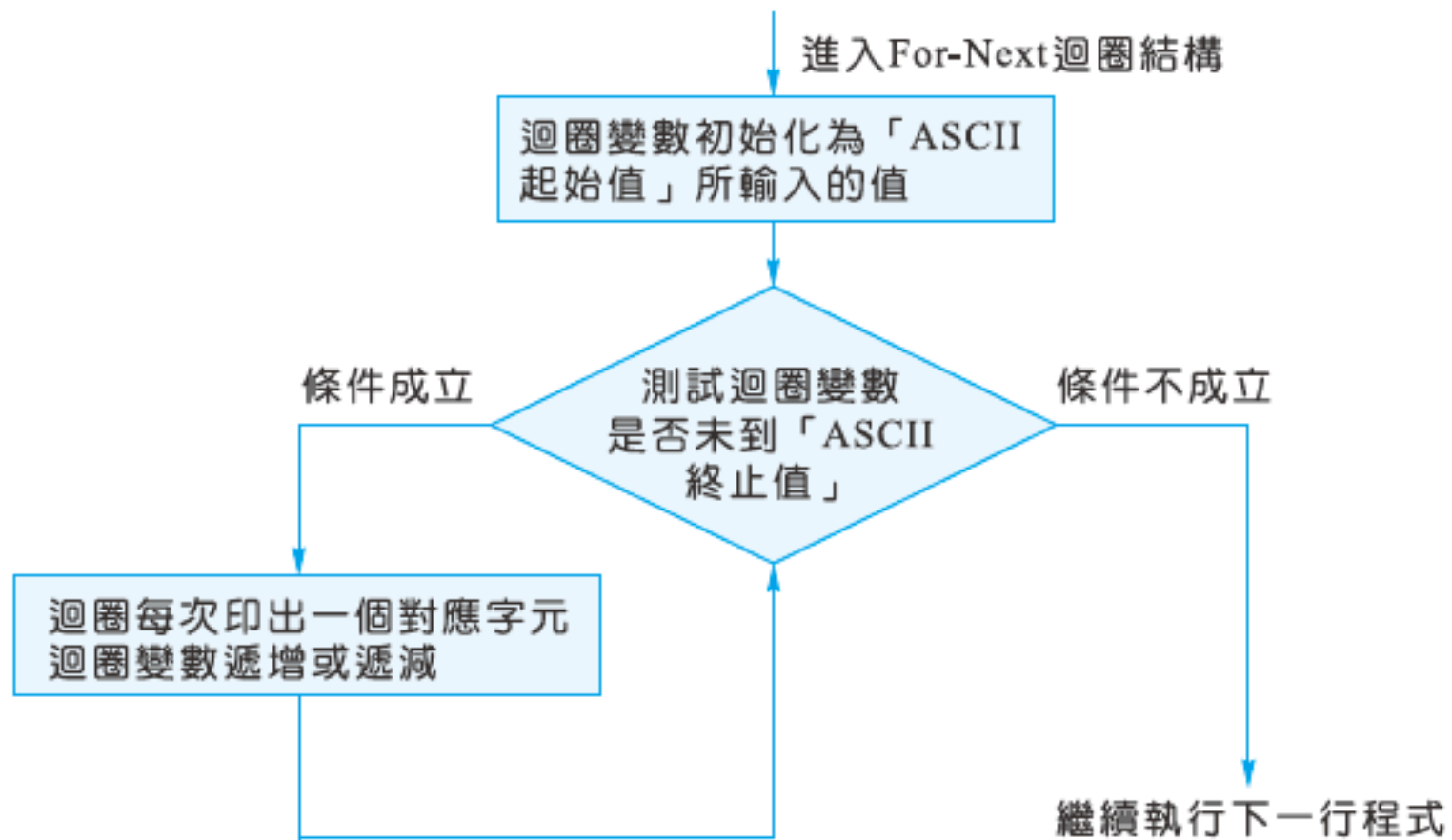
產生ASCII碼(ch5\產生ASCII碼\)



- 電腦中所有資料皆以二進位方式儲存，大小寫英文字母、數字都有國際標準的二進位編碼，這樣的編碼稱為ASCII碼，如A的ASCII碼為65、B的ASCII碼為66、C的ASCII碼為67，依此類推。
- 請寫一程式，利用迴圈與Chr函式列出指定的ASCII碼所對應的字元，流程圖表示如下。Chr函式會將數值轉換成字母，例如：Chr(65)輸出就會變成字母「A」。

產生ASCII碼(ch5\產生ASCII碼\)

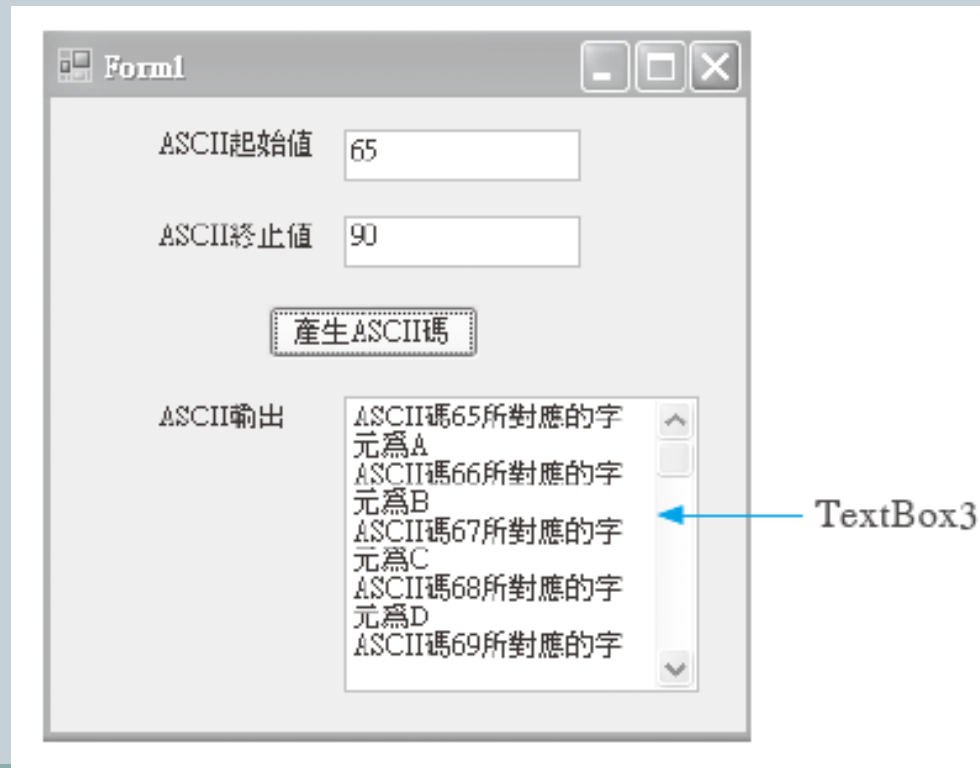
- 流程圖表示



產生ASCII碼(ch5\產生ASCII碼\)

- 預覽結果

- 於ASCII起始值輸入「65」，ASCII終止值輸入「90」，按下「產生ASCII碼」按鈕，結果顯示在下方TextBox3中。



Form1

ASCII起始值 65

ASCII終止值 90

產生ASCII碼

ASCII輸出

ASCII碼65所對應的字元為A
ASCII碼66所對應的字元為B
ASCII碼67所對應的字元為C
ASCII碼68所對應的字元為D
ASCII碼69所對應的字元為E

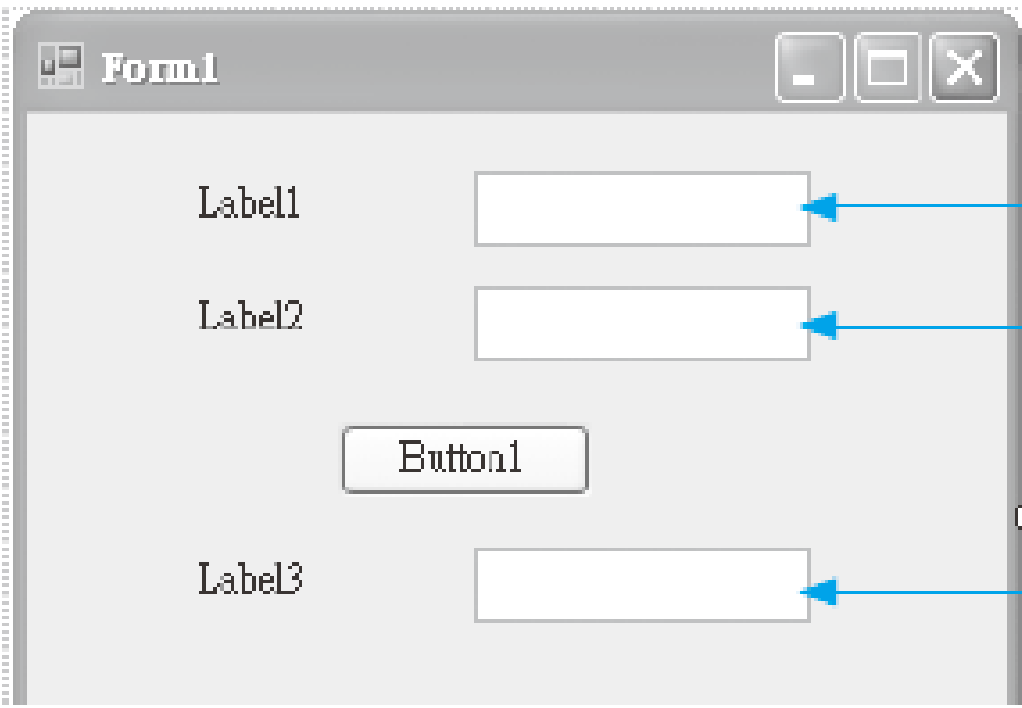
TextBox3

產生ASCII碼(ch5\產生ASCII碼\)



- 表單配置

- 在表單中新增三個**TextBox**，三個**Label**與一個**Button**。



The screenshot shows a Windows form titled "Form1" with a standard Windows XP-style title bar (minimize, maximize, close buttons). The form contains the following controls:

- Label1**: A text label positioned to the left of the first text box.
- TextBox1**: A text input box, indicated by a blue arrow from the label "TextBox1" on the right.
- Label2**: A text label positioned to the left of the second text box.
- TextBox2**: A text input box, indicated by a blue arrow from the label "TextBox2" on the right.
- Button1**: A button with the text "Button1" centered on it.
- Label3**: A text label positioned to the left of the third text box.
- TextBox3**: A text input box, indicated by a blue arrow from the label "TextBox3" on the right.

Blue arrows point from the labels "TextBox1", "TextBox2", and "TextBox3" on the right side of the form to their respective text input boxes.

產生ASCII碼(ch5\產生ASCII碼\)

- 屬性設定

| 物件 | 屬性 | 設定值 |
|----------|------------|-----------|
| Label1 | Text | ASCII起始值 |
| Label2 | Text | ASCII終止值 |
| Label3 | Text | ASCII輸出 |
| Button1 | Text | 產生 ASCII碼 |
| TextBox3 | Multiline | True |
| TextBox3 | Size | 150,125 |
| TextBox3 | ScrollBars | Both |



The screenshot shows a Windows form titled "Form1" with a standard Windows XP-style title bar. The form contains the following elements:

- A label "ASCII起始值" (ASCII Start Value) followed by a text input box.
- A label "ASCII終止值" (ASCII End Value) followed by a text input box.
- A button labeled "產生ASCII碼" (Generate ASCII Code) centered below the input boxes.
- A label "ASCII輸出" (ASCII Output) to the left of a large, empty, multi-line text box.
- The text box has a vertical scrollbar on its right side.

產生ASCII碼(ch5\產生ASCII碼\)



- 程式碼新增與解說
 - 點選「產生ASCII碼」按鈕，新增程式碼在Button1_Click函式中，如下方程式區塊。

| 行號 | 程式碼 |
|----|---|
| 1 | Public Class Form1 |
| 2 | Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click |
| 3 | For i = Val(TextBox1.Text) To Val(TextBox2.Text) |
| 4 | TextBox3.Text = TextBox3.Text & "ASCII碼" & i & "所對應的字元為" & Chr(i) & vbNewLine |
| 5 | Next |
| 6 | End Sub |
| 7 | End Class |

產生ASCII碼(ch5\產生ASCII碼\)

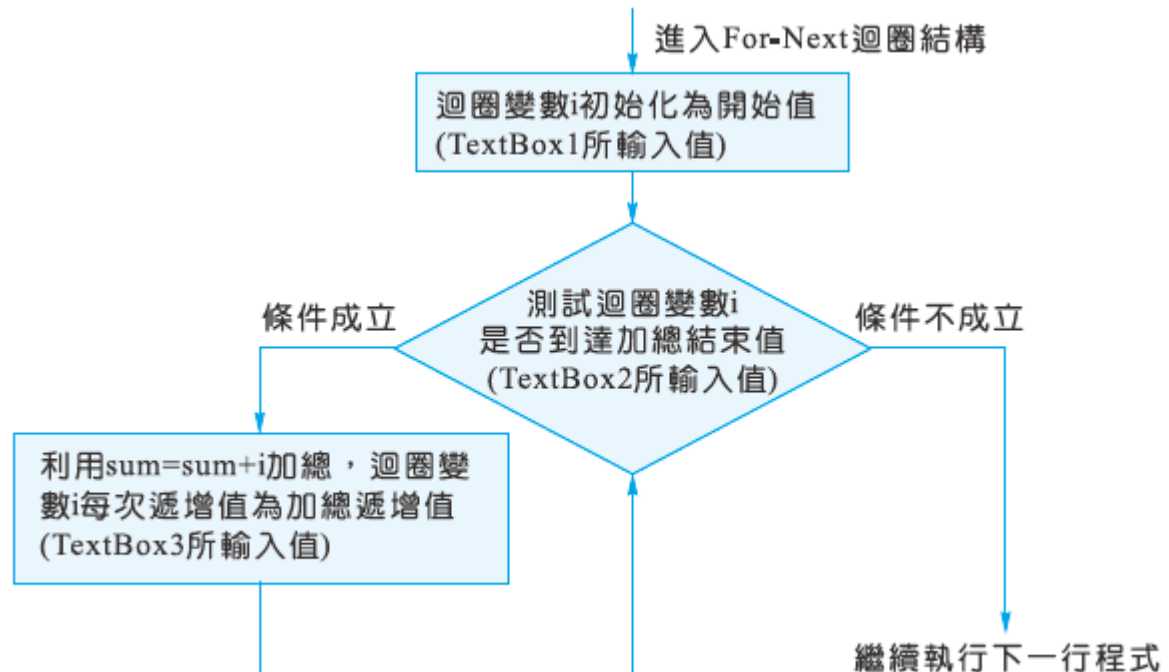


- 解說

- 第3行到第5行：使用**For**迴圈，其中**i**值變化由使用者輸入的**ASCII**起始值到**ASCII**終止值，若不寫**Step**(每次遞增(減)值)，表示每次遞增**1**。
- 第4行：使用**TextBox3**元件輸出**ASCII**值所對應的字元為**Chr(i)**，**i**為迴圈變數當作**Chr**函式的輸入值，**Chr**函式會將輸入值**i**，轉換成該值在**ASCII**表所對應的字母。**vbNewLine**為**Visual Basic**的換行字元，表示換行。

加總(🌀 ch5\加總\)

- 寫一程式允許使用者輸入加總的開始值、結束值與遞增值，計算數值加總的結果。例如：要計算 $1+3+5+7+9+11$ 的結果，就輸入1為開始值、11為結束值、2為遞增值。
- 流程圖表示



加總(🌀 ch5\加總\)

- 預覽結果

- 於加總開始值輸入「1」、加總結束值輸入「11」、加總遞增值輸入「2」，按下「加總」按鈕，結果顯示在**TextBox4**中。

Form1

| | |
|-----------------------------------|----|
| 加總開始值 | 1 |
| 加總結束值 | 11 |
| 加總遞增值 | 2 |
| <input type="button" value="加總"/> | |
| 加總結果 | 36 |

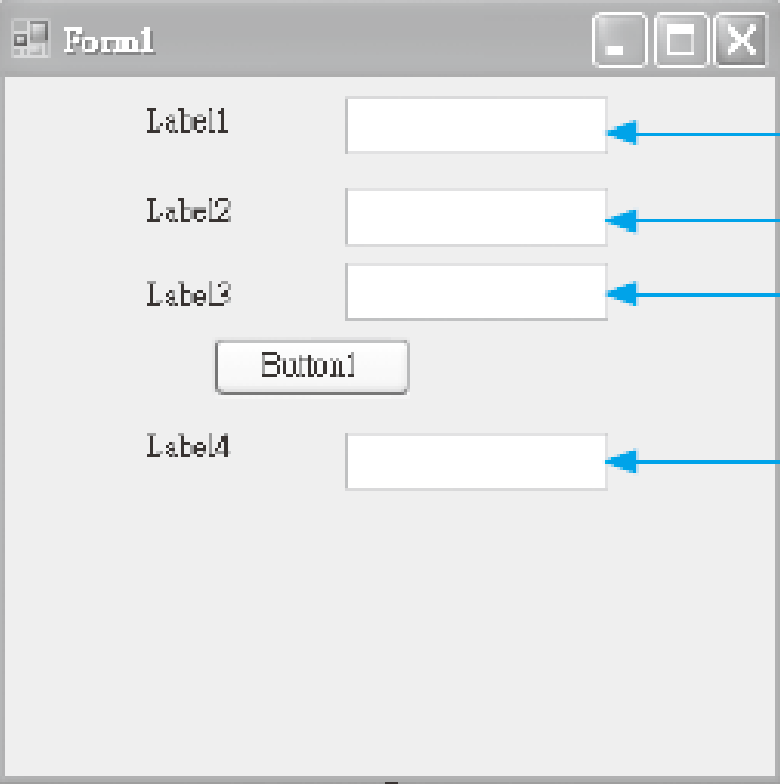
TextBox4

加總(ch5\加總\)



- 表單配置

- 在表單中新增四個**TextBox**，四個**Label**與一個**Button**。



The screenshot displays a Windows Form titled "Form1". It contains four labels (Label1, Label2, Label3, Label4) and four text boxes (TextBox1, TextBox2, TextBox3, TextBox4) arranged vertically. A button labeled "Button1" is positioned between Label3 and Label4. Blue arrows point from the text box labels on the right to their respective text boxes in the form.

加總(ch5\加總\)

• 屬性設定

| 物件 | 屬性 | 設定值 |
|---------|------|-------|
| Label1 | Text | 加總開始值 |
| Label2 | Text | 加總結束值 |
| Label3 | Text | 加總遞增值 |
| Label4 | Text | 加總結果 |
| Button1 | Text | 加總 |



The screenshot shows a Windows form titled "Form1" with a standard title bar (minimize, maximize, close buttons). The form contains the following elements:

- A label "加總開始值" followed by a text input box.
- A label "加總結束值" followed by a text input box.
- A label "加總遞增值" followed by a text input box.
- A button labeled "加總" (Calculate).
- A label "加總結果" followed by a text input box.

加總(ch5\加總\)



- 程式碼新增與解說

- 點選「加總」按鈕，新增程式碼在**Button1_Click**函式中，如下方程式區塊。

| 行號 | 程式碼 |
|----|---|
| 1 | Public Class Form1 |
| 2 | Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click |
| 3 | Dim sum As Integer = 0 |
| 4 | For i As Integer = Val(TextBox1.Text) To Val(TextBox2.Text) Step Val(TextBox3.Text) |
| 5 | sum = sum + i |
| 6 | Next |
| 7 | TextBox4.Text = sum |
| 8 | End Sub |
| 9 | End Class |

加總(ch5\加總\)



- 解說

- 第3行：宣告**sum**變數為整數變數，並初始化為**0**。
- 第4行到第6行：使用**For**迴圈，其中**i**值變化由使用者輸入的加總開始值到加總結束值，每次依加總遞增值遞增，利用變數**sum**暫存加總結果，迴圈中**sum**值的變化將會在下一頁詳細介紹。
- 第7行：將加總結果顯示於**TextBox4**元件。

加總(🌀 ch5\加總\)



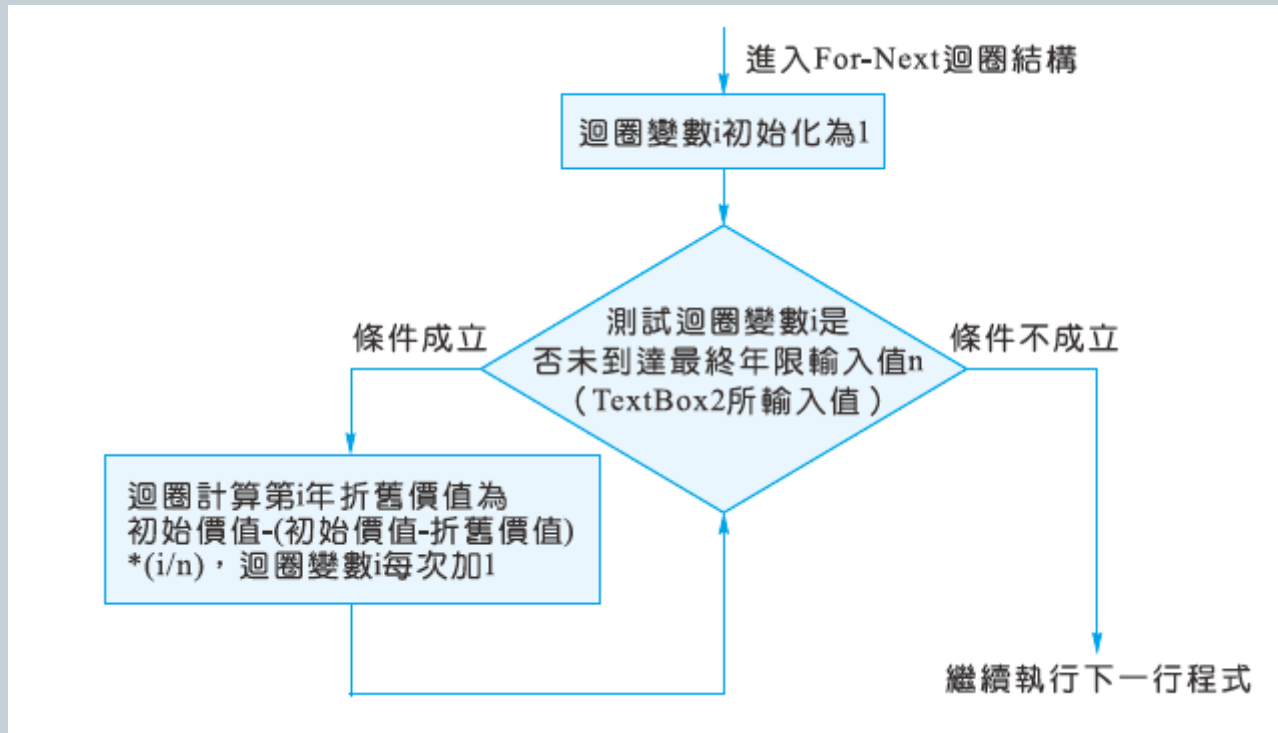
- 舉例說明加總使用 $\text{sum}=\text{sum}+i$ 原理，如下表，在**Visual Basic**中等號右邊($\text{sum}+i$)的算式會先計算，結果回存到等號左邊(sum)。

```
Dim sum As Integer=0
For i As Integer = 1 To 11 Step 2
    sum = sum + i
Next
```

| i值 | sum加總過程 | sum加總後 |
|------|-----------|--------|
| i=1 | sum=0+1 | sum=1 |
| i=3 | sum=1+3 | sum=4 |
| i=5 | sum=4+5 | sum=9 |
| i=7 | sum=9+7 | sum=16 |
| i=9 | sum=16+9 | sum=25 |
| i=11 | sum=25+11 | sum=36 |

折舊計算(ch5\折舊計算\)

- 寫一程式允許使用者輸入初始價值，幾年後的折舊價值，請輸出每一年的折舊價值。
- 流程圖表示



折舊計算(ch5\折舊計算\)

- 預覽結果

- 於初始價值輸入「2000」、n輸入「5」、n年後折舊價值為「1000」，按下「計算折舊」按鈕，結果顯示在折舊結果中。



The screenshot shows a Windows application window titled "Form1". It contains three input fields for depreciation calculation: "初始價值" (Initial Value) set to 2000, "n年後，n為" (n years later, n is) set to 5, and "n年後折舊價值" (n years later depreciation value) set to 1000. Below these is a button labeled "計算折舊" (Calculate Depreciation). Under the button is a label "折舊結果" (Depreciation Result) followed by a text box containing the following text:

| Year | Value |
|------|-------|
| 3年後 | 1400 |
| 4年後 | 1200 |
| 5年後 | 1000 |

折舊計算(ch5\折舊計算\)



- 表單配置

- 在表單中新增四個**TextBox**，四個**Label**與一個**Button**。

The screenshot shows a Windows form titled "Form1" with a standard Windows XP-style title bar. The form contains the following controls:

- Label1**: A label positioned to the left of **TextBox1**.
- Label2**: A label positioned to the left of **TextBox2**.
- Label3**: A label positioned to the left of **TextBox3**.
- Label4**: A label positioned to the left of **TextBox4**.
- TextBox1**, **TextBox2**, **TextBox3**, and **TextBox4**: Four empty text input boxes arranged vertically.
- Button1**: A button labeled "Button1" located between **Label3** and **Label4**.

Blue arrows indicate the mapping from the labels to the text boxes: Label1 to TextBox1, Label2 to TextBox2, Label3 to TextBox3, and Label4 to TextBox4.

折舊計算(ch5\折舊計算\)

- 屬性設定

| 物件 | 屬性 | 設定值 |
|----------|------------|-------------|
| Label1 | Text | 初始價值 |
| Label2 | Text | n年後， n為 |
| Label3 | Text | n年後折舊 價值 |
| Label4 | Text | 折舊結果 |
| Button1 | Text | 計算折舊 |
| TextBox4 | Multiline | True |
| TextBox4 | Size | 100,100 |
| TextBox4 | ScrollBars | Both |



The screenshot shows a Windows Form titled "Form1" with a standard Windows XP-style title bar. The form contains the following elements:

- A text label "初始價值" (Initial Value) followed by a single-line text box.
- A text label "n年後，n為" (n years later, n is) followed by a single-line text box.
- A text label "n年後折舊價值" (Depreciation value after n years) followed by a single-line text box.
- A button labeled "計算折舊" (Calculate Depreciation).
- A text label "折舊結果" (Depreciation Result) followed by a large multiline text box with vertical scrollbars.

折舊計算(ch5\折舊計算\)



- 程式碼新增與解說

- 點選「計算折舊」按鈕，新增程式碼在Button1_Click函式中，如下方程式區塊。

| 行號 | 程式碼 |
|----|-----|
|----|-----|

| | |
|----|---|
| 1 | Public Class Form1 |
| 2 | Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click |
| 3 | Dim value As Double |
| 4 | Dim beginValue As Double = Val(TextBox1.Text) |
| 5 | Dim endValue As Double = Val(TextBox3.Text) |
| 6 | Dim n As Integer = Val(TextBox2.Text) |
| 7 | For i As Integer = 1 To Val(TextBox2.Text) |
| 8 | value = beginValue - (beginValue - endValue) * i / n |
| 9 | TextBox4.Text = TextBox4.Text & i & "年後的價值為" & value & |
| | vbNewLine |
| 10 | Next |
| 11 | End Sub |
| 12 | End Class |

折舊計算(ch5\折舊計算\)

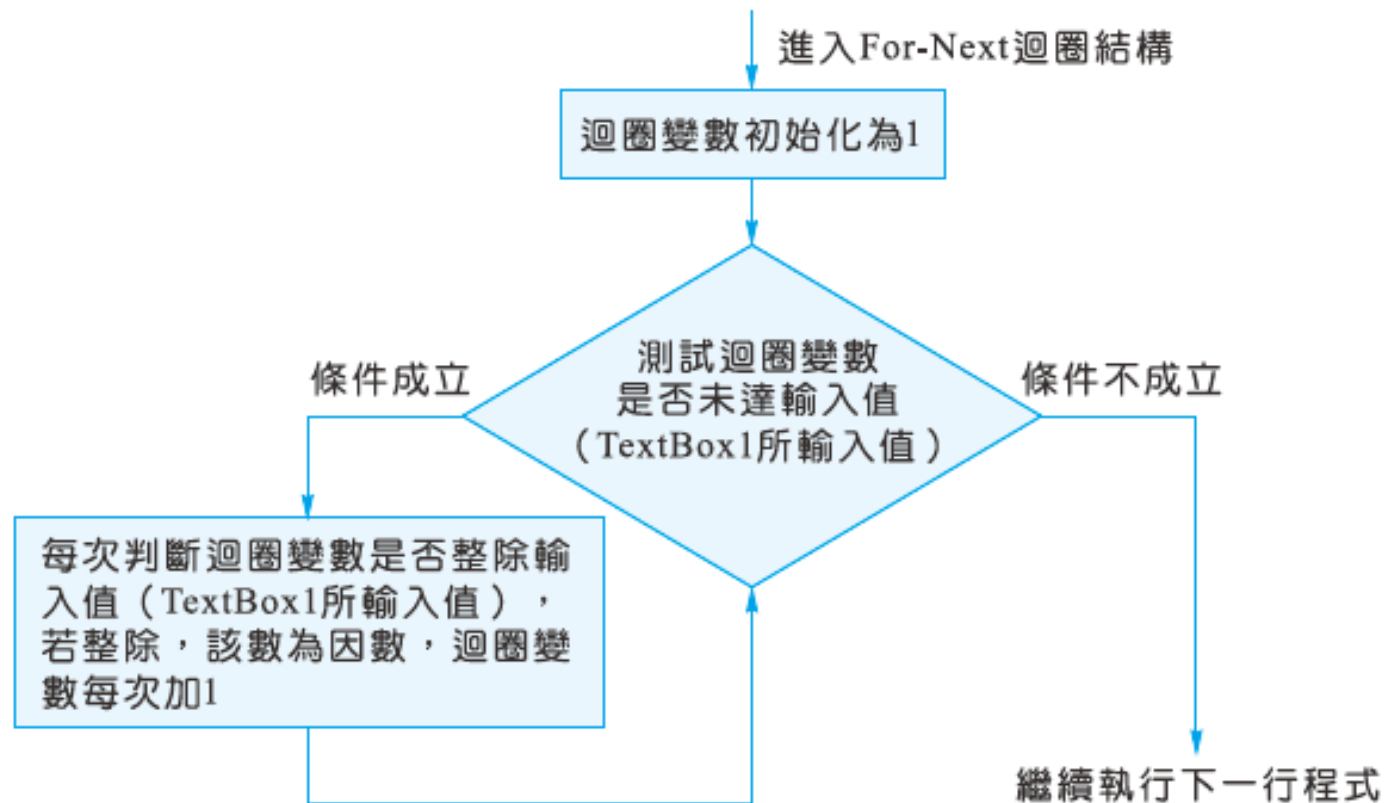


- 解說

- 第3行到第6行：宣告**value**為倍精度浮點數，儲存每年折舊計算後結果，**beginValue**與**endValue**為倍精度浮點數，分別儲存初始價值與**n**年後折舊價值，宣告**n**為整數變數，儲存由**TextBox2**所輸入的**n**值。
- 第7行到第10行：使用**For**迴圈，其中迴圈變數**i**值由1到**n(TextBox2所輸入值)**，每次遞增1，利用「**beginValue - (beginValue - endValue) * i / n**」，計算每年折舊價值，每年折舊價值輸出到元件**TextBox4**，**vbNewLine**為**Visual Basic**的換行字元，表示換行。

求出所有因數(ch5\求出所有因數\)

- 寫一程式允許使用者輸入數值，求出該數的所有因數。
- 流程圖表示



求出所有因數(ch5\求出所有因數\)

- 預覽結果

- 於輸入值中輸入「100」，按下「求出所有因數」按鈕，結果顯示在TextBox2中。

The screenshot shows a Windows Form titled 'Form1'. It contains a text box labeled '輸入值' (Input Value) with the number '100' entered. Below it is a button labeled '求出所有因數' (Find all factors). Below the button is a list box labeled '所有因數' (All factors) which contains the following numbers: 1, 2, 4, 5, 10, 20, 25, 50, and 100. A blue arrow points from the text 'TextBox2' to the list box.

| 所有因數 |
|------|
| 1 |
| 2 |
| 4 |
| 5 |
| 10 |
| 20 |
| 25 |
| 50 |
| 100 |

求出所有因數(ch5\求出所有因數\)



- 表單配置

- 在表單中新增兩個TextBox，兩個Label與一個Button。

The screenshot shows a Windows form titled "Form1" with a standard Windows XP-style title bar (minimize, maximize, close buttons). The form contains the following controls:

- Label1**: Located at the top left.
- TextBox1**: A text input field located to the right of Label1. A blue arrow points from the text "TextBox1" on the right to this box.
- Button1**: A button labeled "Button1" located below Label1 and TextBox1.
- Label2**: Located at the bottom left.
- TextBox2**: A text input field located to the right of Label2. A blue arrow points from the text "TextBox2" on the right to this box.

求出所有因數(ch5\求出所有因數\)

• 屬性設定

| 物件 | 屬性 | 設定值 |
|----------|------------|---------|
| Label1 | Text | 輸入值 |
| Label2 | Text | 所有因數 |
| Button1 | Text | 求出所有因數 |
| TextBox2 | Multiline | True |
| TextBox2 | Size | 150,150 |
| TextBox2 | ScrollBars | Both |



求出所有因數(ch5\求出所有因數\)



- 程式碼新增與解說

- 點選「求出所有因數」按鈕，新增程式碼在**Button1_Click**函式中，如下方程式區塊。

| 行號 | 程式碼 |
|----|--|
| 1 | Public Class Form1 |
| 2 | Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click |
| 3 | For i As Integer = 1 To Val(TextBox1.Text) Step 1 |
| 4 | If ((Val(TextBox1.Text) Mod i) = 0) Then |
| 5 | TextBox2.Text = TextBox2.Text & i & vbCrLf |
| 6 | End If |
| 7 | Next |
| 8 | End Sub |
| 9 | End Class |

求出所有因數(ch5\求出所有因數\)



- 解說

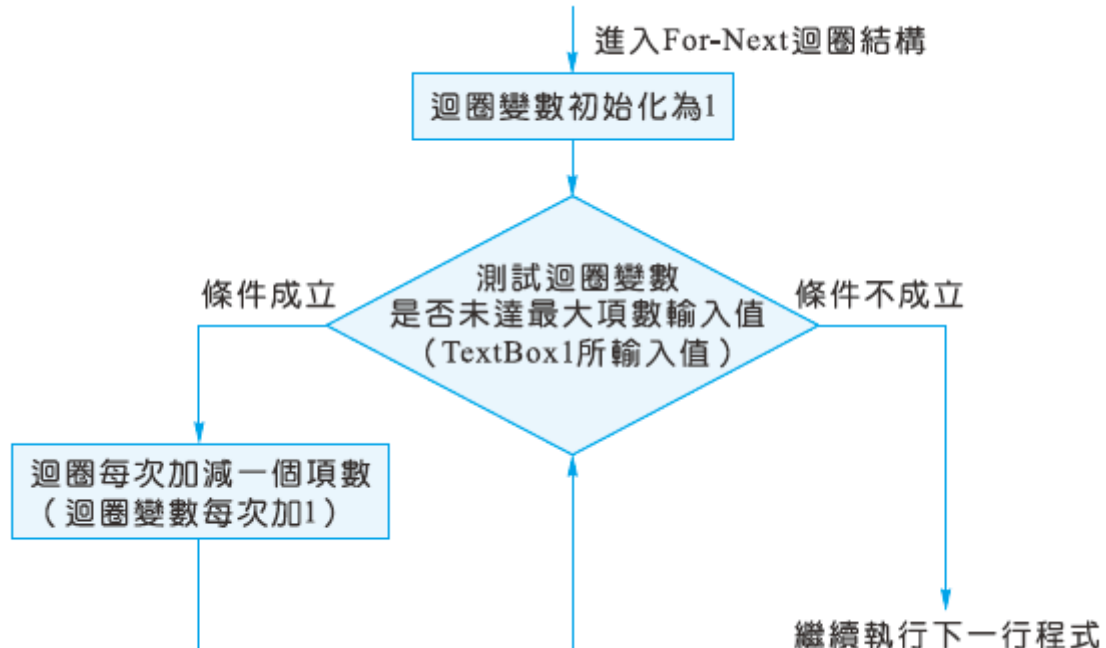
- 第3行到第7行：使用**For**迴圈，其中迴圈變數**i**值由1到輸入值(**TextBox1**所輸入值)，每次遞增1。
- 第4行到第6行：利用**If**判斷檢查輸入值與迴圈變數**i**相除後餘數是否為0，若餘數為0，表示該數為因數，輸出到元件**TextBox2**，**vbNewLine**為**Visual Basic**的換行字元，表示輸出到此換行。

- 程式如何求因數？

- **X**為**Y**的因數，表示**X**可以整除**Y**，意即**Y**除以**X**的餘數為0，在**Visual Basic**中提供運算子「**Mod**」為求餘數，求**Y**除以**X**的餘數轉換程式為「**Y Mod X**」，再利用選擇結構(**If**)判斷餘數是否為0，若是，則**Y**是**X**的因數。例如：**12 Mod 2**的運算結果為0，所以2為12的因數。

計算PI(ch5\計算PI\)

- 計算PI的公式，如下。
 - $PI=4(1-1/3+1/5-1/7+1/9-1/11+1/13-...)$
- 當項數越多，則所獲得的PI值越準確，寫一程式允許使用者輸入最大項數，求出PI值
- 流程圖表示




計算PI(ch5\計算PI\)



- 預覽結果

- 於最大項數中輸入「100000」，按下「計算PI」按鈕，結果顯示在TextBox2中。



Form1

最大項數 100000

計算PI

PI值 3.14160265348972

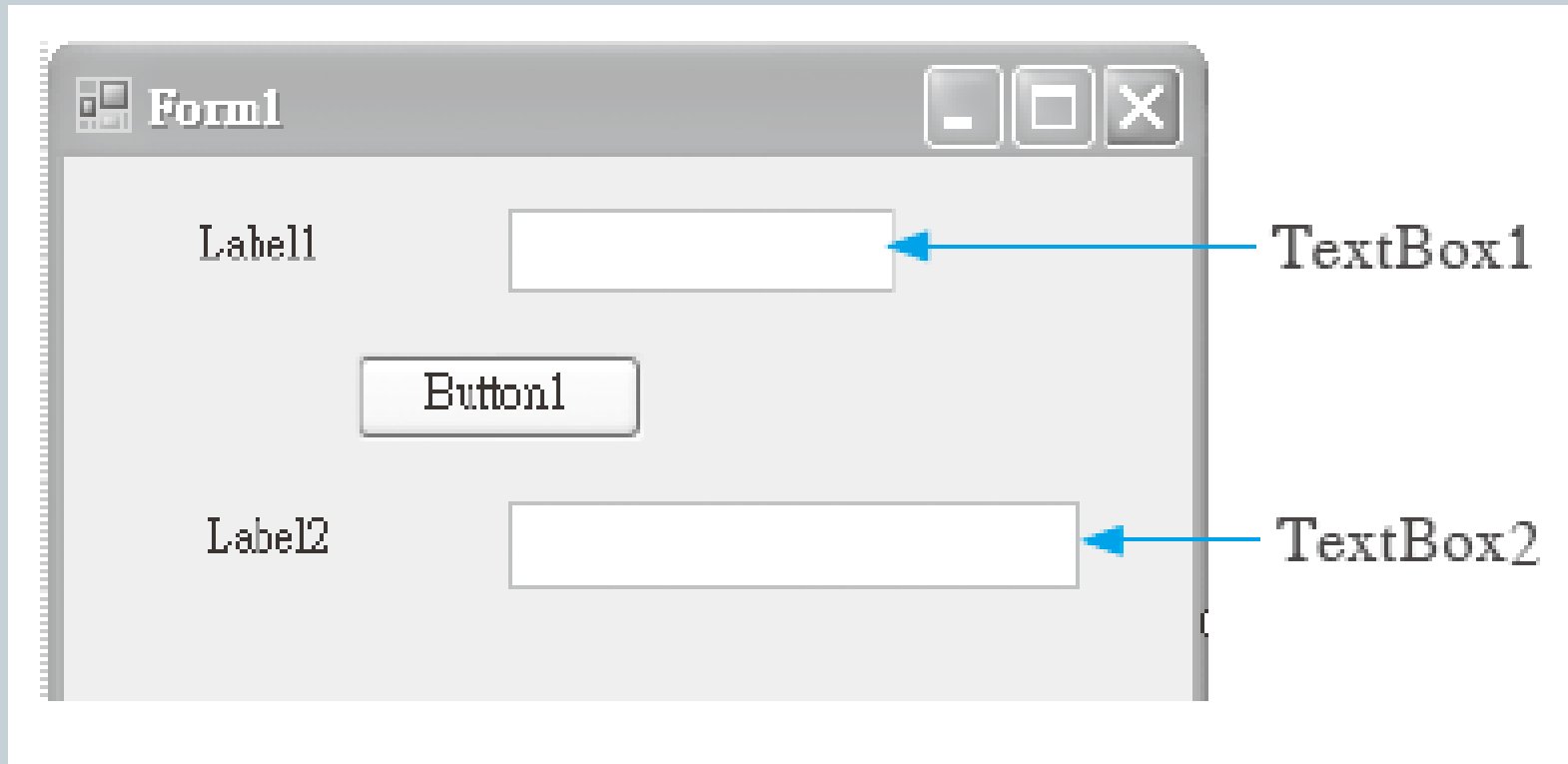
TextBox2

計算PI(ch5\計算PI\)



- 表單配置

- 在表單中新增兩個TextBox，兩個Label與一個Button。



The screenshot shows a Windows form titled "Form1" with a standard Windows XP-style title bar. Inside the form, there are four controls arranged in a grid-like fashion:

- Label1** is positioned to the left of **TextBox1**.
- Button1** is positioned below **Label1** and **TextBox1**.
- Label2** is positioned to the left of **TextBox2**.
- TextBox2** is positioned below **Label2** and **Button1**.

Blue arrows point from the external labels **TextBox1** and **TextBox2** to their respective text boxes on the form.

計算PI(ch5\計算PI\)



- 屬性設定

| 物件 | 屬性 | 設定值 |
|---------|------|-------|
| Label1 | Text | 最大項數 |
| Label2 | Text | PI值 |
| Button1 | Text | 計算 PI |



Form1

最大項數

PI值

計算PI(ch5\計算PI\)



- 程式碼新增與解說

- 點選「計算PI」按鈕，新增程式碼在**Button1_Click**函式中，如下方程式區塊。

| 行號 | 程式碼 |
|----|---|
| 1 | Public Class Form1 |
| 2 | Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click |
| 3 | Dim PI4 As Double = 1 |
| 4 | For x As Integer = 1 To Val(TextBox1.Text) Step 1 |
| 5 | PI4 = PI4 + ((-1) ^ x) * (1 / (2 * x + 1)) |
| 6 | Debug.Print(PI4) |
| 7 | Next |
| 8 | TextBox2.Text = PI4 * 4 |
| 9 | End Sub |
| 10 | End Class |

計算PI(ch5\計算PI\)



- 解說

- 第3行：宣告PI4為倍精度浮點數變數。
- 第4行到第6行：使用For迴圈，其中迴圈變數x值由1到最大項數輸入值(TextBox1所輸入值)，每次遞增1，PI4隨著迴圈不斷加(減)值(第5行)，PI4詳細計算細節如下張投影片。
- 第7行：將PI4值乘以4，得到PI顯示在TextBox2元件。

計算PI(ch5\計算PI\)



- 舉例說明加總使用 $PI4 = PI4 + ((-1)^x * (1 / (2 * x + 1)))$ 原理，如下表，在Visual Basic中，等號右邊 $PI4 + ((-1)^x * (1 / (2 * x + 1)))$ 的算式會先計算，結果回存到等號左邊(PI4)。

```
Dim PI4 As Double = 1
```

```
For x As Integer = 1 To Val(TextBox1.Text)
```

```
    PI4 = PI4 + ((-1) ^ x) * (1 / (2 * x + 1))
```

```
Next
```

| x值 | PI4加總過程 | PI4加總後 |
|------|------------------|-------------|
| x=1 | PI4=1-1/3 | PI4=0.66666 |
| x=3 | PI4=0.66666+1/5 | PI4=0.86666 |
| x=5 | PI4=0.86666-1/7 | PI4=0.72380 |
| x=7 | PI4=0.72380+1/9 | PI4=0.83492 |
| x=9 | PI4=0.83492-1/11 | PI4=0.74401 |
| x=11 | PI4=0.74401+1/13 | PI4=0.82093 |

迴圈結構—使用Do-While



- **Do-While**迴圈結構與**For-Next**迴圈結構十分類似，**Do-While**迴圈結構常用於**不固定次數的迴圈**，由迴圈中測試條件決定是否跳出迴圈，測試條件為真時繼續迴圈；當測試條件為假時結束迴圈。
- 如猜數字遊戲，兩人(**A**與**B**)玩猜數字遊戲，一人(**A**)心中想一個數，另一人(**B**)去猜，**A**就**B**所猜數字回答「猜大一點」或「猜小一點」，直到**B**猜到**A**所想數字，這樣的猜測就屬於不固定次數的迴圈，適合使用**Do-While**而不適合使用**For-Next**。

迴圈結構—使用Do-While



- 程式語法

| Do-While迴圈語法 | 程式範例 |
|---|--|
| 迴圈變數=初始值 Do While (迴圈變數 <= 終止值) 重複的程式 迴圈變數=迴圈變數+遞增(減)值 Loop | Dim j As Integer = 0 Do While (j <= 10) sum = sum + j j = j + 1 Loop |

迴圈結構—使用Do-While

- For-Next迴圈結構轉換成Do-While迴圈結構

| For-Next迴圈結構 | Do-While迴圈結構 |
|---|--|
| <pre>Dim sum As Integer = 0 For i As Integer = 1 To 10 Step 1 sum = sum + i Next</pre> <p>Diagram labels for For-Next:</p> <ul style="list-style-type: none">迴圈變數終止值 (points to 10 in "To 10")迴圈變數起始值 (points to 1 in "1 To 10")迴圈變數遞增(減)值 (points to Step 1) | <pre>Dim sum As Integer = 0 Dim i As Integer = 1 Do While (i <= 10) sum = sum + i i = i + 1 Loop</pre> <p>Diagram labels for Do-While:</p> <ul style="list-style-type: none">迴圈變數起始值 (points to 1 in "Dim i As Integer = 1")迴圈變數終止值 (points to <= 10 in "Do While (i <= 10)")迴圈變數遞增(減)值 (points to i = i + 1) |

迴圈結構—使用Do-While



- 前測式迴圈結構與後測式迴圈結構
- **Do-While**迴圈結構有兩種，分成前測式迴圈與後測式迴圈。
- 前測式迴圈是指先測試迴圈變數是否符合迴圈終止條件；後測式迴圈是指先執行迴圈一次，再測試迴圈變數是否符合迴圈終止條件，兩者的差異在於後測式迴圈至少執行一次。
- 要使用哪一種結構是看程式功能需求，如帳號密碼登入功能至少要让使用者輸入一次帳號密碼，再確認帳號密碼是否正確，就可以使用後測式迴圈結構

迴圈結構—使用Do-While



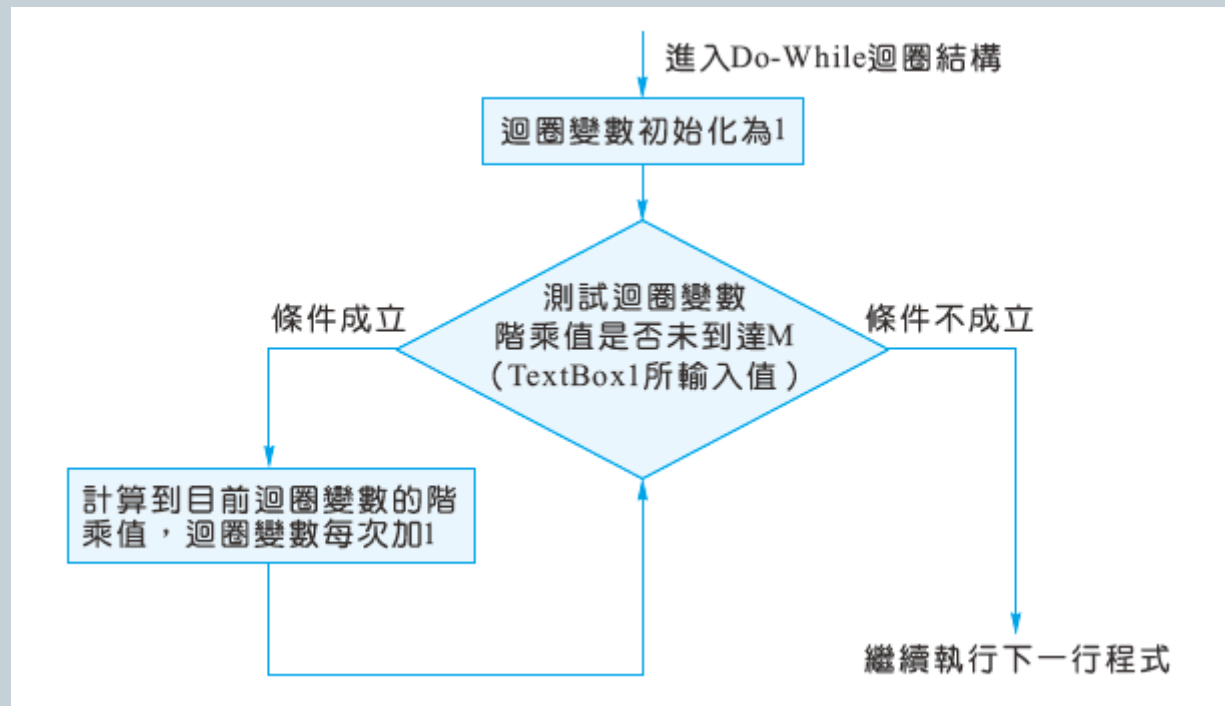
- 程式語法

| 前測式Do-While迴圈 | 後測式Do-While迴圈 |
|---|---|
| <pre>Dim sum As Integer = 0 Dim j As Integer = 1 Do While (j <= 10) sum = sum + j j = j + 1 Loop</pre> | <pre>Dim sum As Integer = 0 Dim j As Integer = 1 Do sum = sum + j j = j + 1 Loop While (j <= 10)</pre> |
| 先測試j是否小於等於10，再執行迴圈中動作。 | 先執行迴圈中動作，再測試j是否小於等於10。 |

註：兩者迴圈執行結束，sum結果一致都是55，兩者差異只在後測式先執行一次再測試迴圈變數。

階乘計算(ch5\階乘計算\)

- 請計算N為多少時，其階乘值超過M。N階乘表示為N!，其值為「 $1*2*3*...*(n-1)*n$ 」，使用Do-While迴圈計算，N!超過M的最小N值為何？
- 流程圖表示

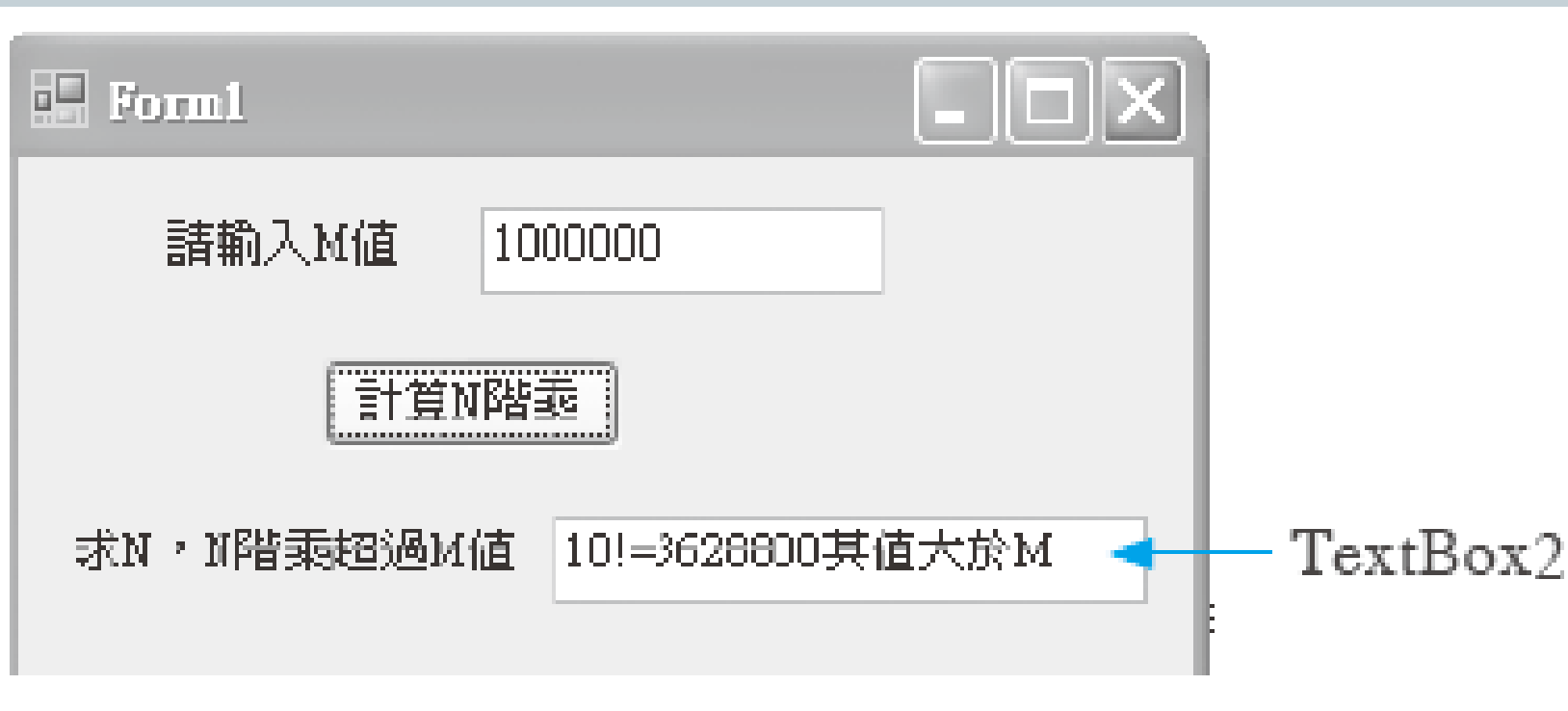


階乘計算(ch5\階乘計算\)



- 預覽結果

- 輸入M值為「1000000」，按下「計算N階乘」按鈕，結果顯示在TextBox2中。



Form1

請輸入M值 1000000

計算N階乘

求N，N階乘超過M值 10!=3628800其值大於M

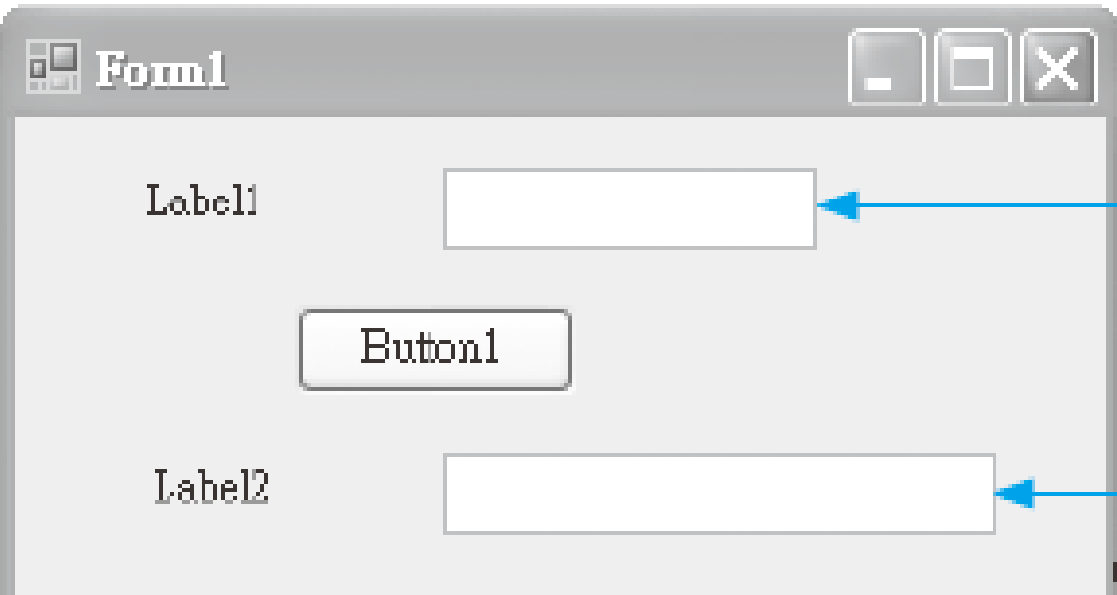
TextBox2

階乘計算(ch5\階乘計算\)



- 表單配置

- 在表單中新增兩個TextBox，兩個Label與一個Button。



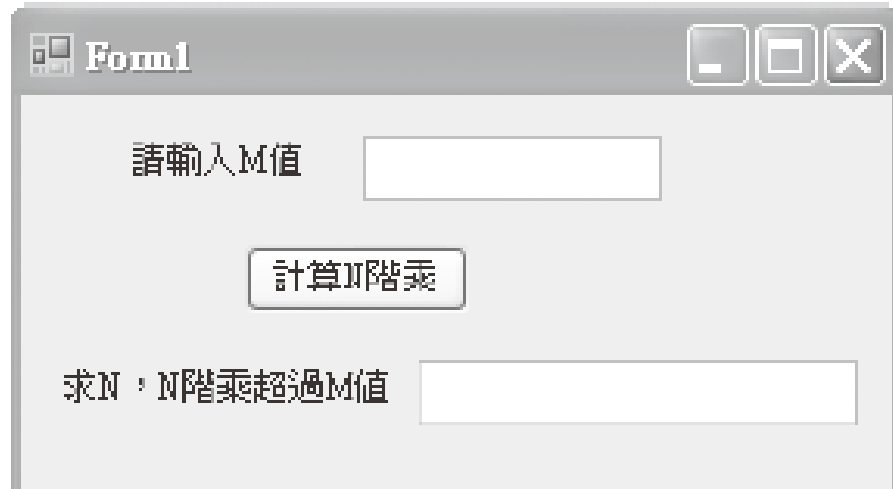
The screenshot shows a Windows form titled "Form1" with a standard Windows XP-style title bar. Inside the form, there are two labels, "Label1" and "Label2", positioned to the left of two text boxes, "TextBox1" and "TextBox2". A button labeled "Button1" is located between the two text boxes. Blue arrows point from the external labels "Label1" and "Label2" to the corresponding text boxes "TextBox1" and "TextBox2" respectively.

階乘計算(ch5\階乘計算\)



- 屬性設定

| 物件 | 屬性 | 設定值 |
|---------|------|------------|
| Label1 | Text | 請輸入 M值 |
| Label2 | Text | 求N，N階乘超過M值 |
| Button1 | Text | 計算 N階乘 |



The screenshot shows a Windows form titled "Form1" with a standard title bar (minimize, maximize, close buttons). The form contains the following elements:

- A label "請輸入M值" (Please enter M value) followed by a text input box.
- A button labeled "計算N階乘" (Calculate N factorial).
- A label "求N，N階乘超過M值" (Find N, N factorial exceeds M value) followed by a text input box.

階乘計算(ch5\階乘計算\)



- 程式碼新增與解說

- 點選「計算N階乘」按鈕，新增程式碼在**Button1_Click**函式中，如下方程式區塊。

| 行號 | 程式碼 |
|----|---|
| 1 | Public Class Form1 |
| 2 | Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click |
| 3 | Dim i As Integer = 1 |
| 4 | Dim fac As Integer = 1 |
| 5 | Dim M As Integer = Val(TextBox1.Text) |
| 6 | Do While (fac <= M) |
| 7 | i = i + 1 |
| 8 | fac = fac * i |
| 9 | Loop |
| 10 | TextBox2.Text = TextBox2.Text & i & "!=" & fac & "其值大於M" |
| 11 | End Sub |
| 12 | End Class |

階乘計算(ch5\階乘計算\)



- 解說

- 第3行：宣告*i*為迴圈變數，並初始化為1。
- 第4行：宣告*fac*為整數變數，用於儲存階乘結果。
- 第5行：宣告*M*為整數變數，當作迴圈條件測試，用於判斷階乘值是否超過*M*。
- 第6行到第9行：使用Do-While迴圈，其中迴圈變數*i*值由1到階乘值超過*M*的數值
- (第6行)，每次遞增1(第7行)，*fac*為計算階乘的結果(第8行)，*fac*詳細計算細節如次頁表格。
- 第10行：將階乘超過*M*的結果，顯示在TextBox2元件。

階乘計算(ch5\階乘計算\)

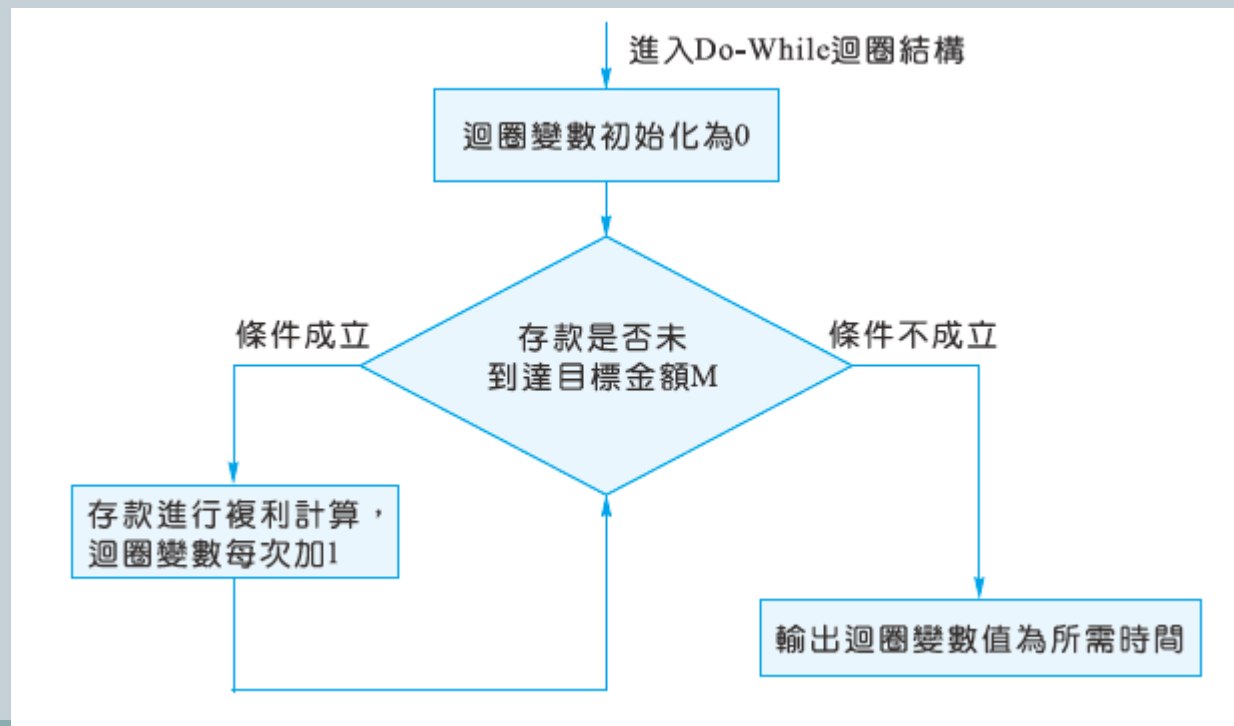


- 舉例說明加總使用 $\text{fac} = \text{fac} * i$ 原理，如次頁表格，在 **Visual Basic** 中，等號右邊 ($\text{fac} * i$) 的算式會先計算，結果回存到等號左邊 (fac)。

| 程式碼 | i值 | fac=fac*i相乘過程 | fac相乘後 |
|---|-----|---------------|----------|
| <pre>Dim i As Integer = 1 Dim fac As Integer = 1 Dim M As Integer = 1000000 Do While (fac <= M) i = i + 1 fac = fac * i Loop</pre> | i=2 | fac=1*2 | fac=2 |
| | i=3 | fac=2*3 | fac=6 |
| | i=4 | fac=6*4 | fac=24 |
| | i=5 | fac=24*5 | fac=120 |
| | i=6 | fac=120*6 | fac=720 |
| | i=7 | fac=720*7 | fac=5040 |
| | ... | ... | ... |

複利計算(ch5\複利計算\)

- 請寫一程式，計算若存一筆錢(N)在銀行，以複利計息，年利率為(X)，求出最少需要幾年才能達到目標金額(M)，流程圖表示如下。
- 流程圖表示

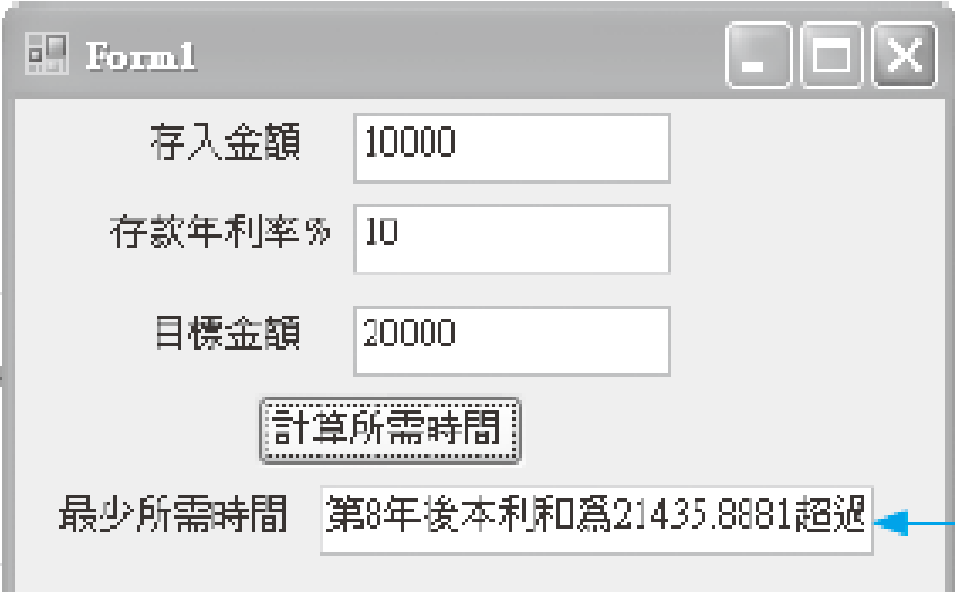


複利計算(ch5\複利計算\)



- 預覽結果

- 於存入金額中輸入「10000」(元)，存款年利率輸入「10」(%)，目標金額輸入「20000」(元)，按下「計算所需時間」按鈕，結果顯示在**TextBox4**中。



Form1

存入金額 10000

存款年利率% 10

目標金額 20000

計算所需時間

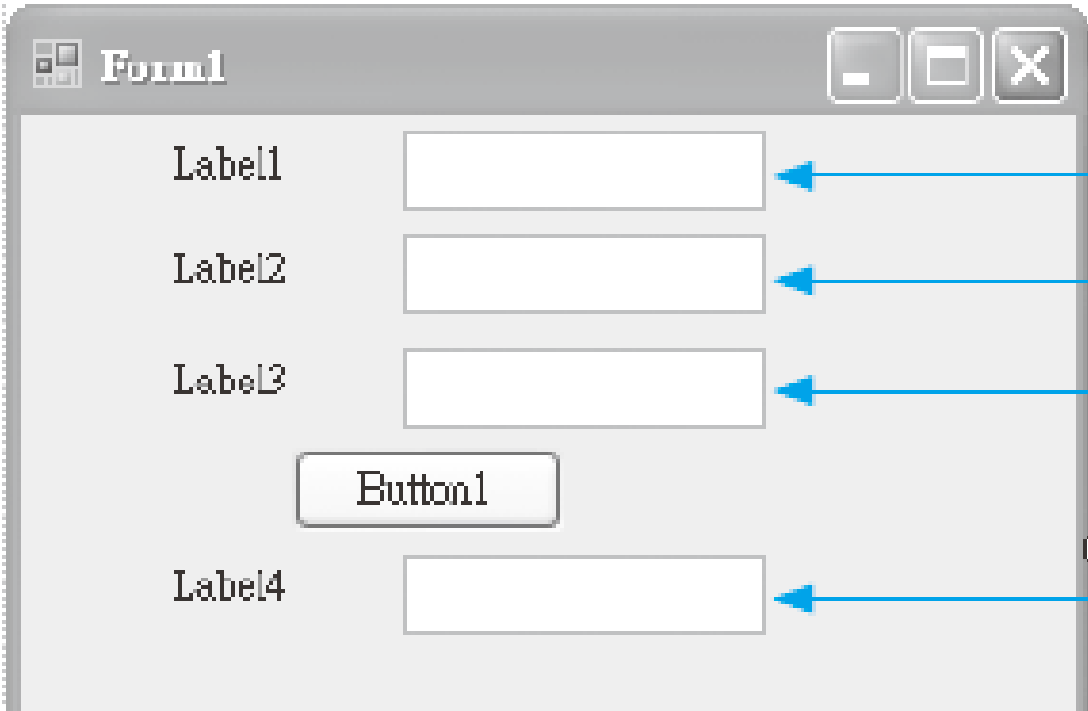
最少所需時間 第8年後本利和為21435.8881超過 ← TextBox4

複利計算(ch5\複利計算\)



- 表單配置

- 在表單中新增四個**TextBox**，四個**Label**與一個**Button**。



Form1

Label1

Label2

Label3

Button1

Label4

TextBox1

TextBox2

TextBox3

TextBox4

複利計算(ch5\複利計算\)



- 屬性設定

| 物件 | 屬性 | 設定值 |
|---------|------|---------|
| Label1 | Text | 存入金額 |
| Label2 | Text | 存款年利率 % |
| Label3 | Text | 目標金額 |
| Label4 | Text | 最少所需時間 |
| Button1 | Text | 計算所需時間 |



The screenshot shows a Windows form titled "Form1" with the following elements:

- 存入金額: Input field
- 存款年利率 %: Input field
- 目標金額: Input field
- 計算所需時間: Button
- 最少所需時間: Input field

複利計算(ch5\複利計算\)



○ 程式碼新增與解說

- ✧ 點選「計算所需時間」按鈕，新增程式碼在**Button1_Click**函式中，如下方程式區塊。

| 行號 | 程式碼 |
|----|---|
| 1 | Public Class Form1 |
| 2 | Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click |
| 3 | Dim N As Double = Val(TextBox1.Text) |
| 4 | Dim X As Double = Val(TextBox2.Text) |
| 5 | Dim M As Double = Val(TextBox3.Text) |
| 6 | Dim i As Integer = 0 |
| 7 | Do While (N <= M) |
| 8 | i = i + 1 |
| 9 | N = N * (1 + X / 100) |
| 10 | Loop |
| 11 | TextBox4.Text = TextBox4.Text & "第" & i & "年後本利和為" & N & "超過" & M |
| 12 | End Sub |

複利計算(ch5\複利計算\)



● 解說

- 第3行：宣告**N**為倍精度浮點數，並初始化為**TextBox1**元件的輸入，表示存入金額。
- 第4行：宣告**X**為倍精度浮點數，並初始化為**TextBox2**元件的輸入，表示存款年利率。
- 第5行：宣告**M**為倍精度浮點數，並初始化為**TextBox3**元件的輸入，表示目標金額。
- 第6行：宣告**i**為迴圈變數，並初始化為**0**，因為進**Do-While**迴圈後，本利和**N**計算後才遞增**i**，最後輸出才會正確。
- 第7行到第10行：使用**Do-While**迴圈，其中迴圈變數**i**值由**1**到本金和超過目標金額**M**(第10行)，每次遞增**1**(第8行)，**N**為計算本利和的結果(第9行)，**N**詳細計算細節如下張投影片。
- 第11行：將本利和超過**M**的結果，顯示在**TextBox4**元件。

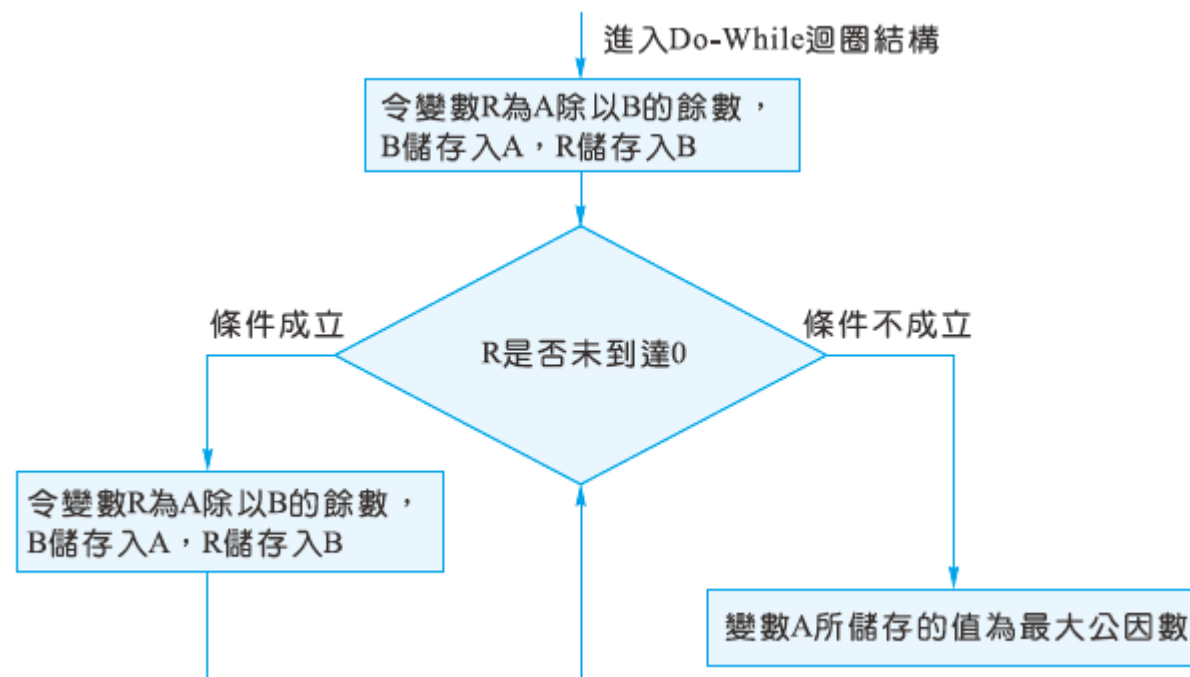
複利計算(ch5\複利計算\)

- 舉例說明加總使用 $N = N * (1 + X / 100)$ 原理，如下表，在**Visual Basic**中，等號右邊($N * (1 + X / 100)$)的算式會先計算，結果回存到等號左邊(N)。

| <pre>Dim N As Double = 10000 Dim X As Double = 10 Dim M As Double = 20000 Dim i As Integer = 0 Do i = i + 1 N = N * (1 + X / 100) Loop While (N <= M)</pre> | i值 | N = N * (1 + X / 100) 相乘過程 | N = N * (1 + X / 100) 相乘後 |
|--|-----|-------------------------------|------------------------------|
| | i=1 | N=10000*(1+10/100) | N=11000 |
| | i=2 | N=11000*(1+10/100) | N=12100 |
| | i=3 | N=12100*(1+10/100) | N=13310 |
| | i=4 | N=13310*(1+10/100) | N=14641 |
| | i=5 | N=14641*(1+10/100) | N=16105.1 |
| | i=6 | N=16105.1*(1+10/100) | N=17715.61 |
| | i=7 | N=17715.61*(1+10/100) | N=19487.171 |
| | i=8 | N=19487.171*(1+10/100) | N=21435.8881 |

求兩數的最大公因數(ch5\最大公因數\)

- 兩數的最大公因數定義為整除這兩數的最大數。假設兩數為**A**與**B**，求最大公因數可以使用**A**與**B**的最大公因數等於**B**與「**A**除以**B**的餘數」的最大公因數
- 流程圖表示

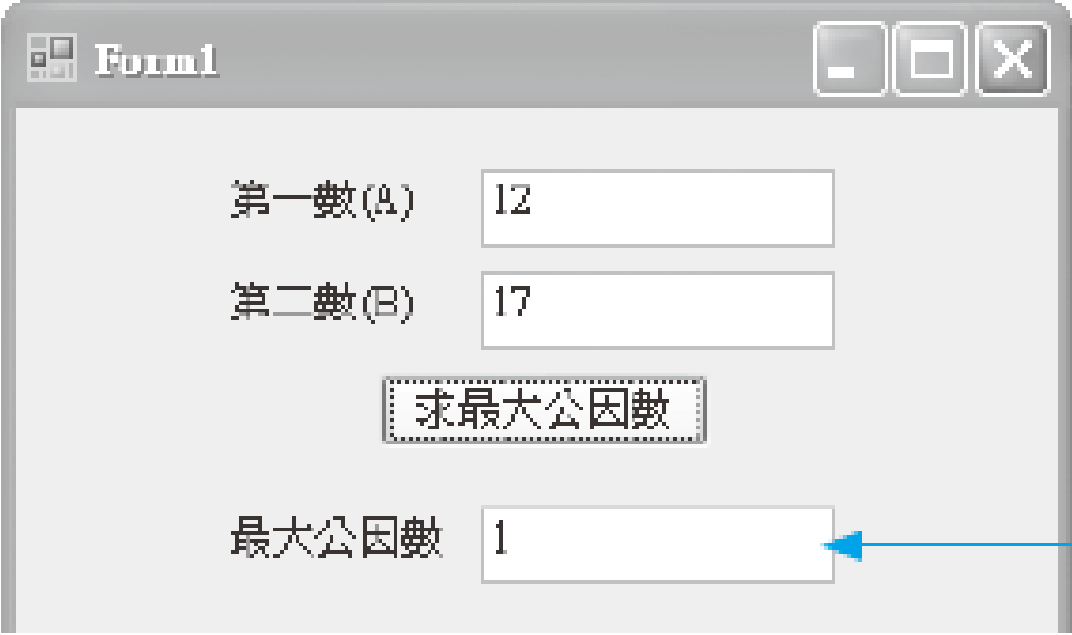


求兩數的最大公因數(ch5\最大公因數\)



- 預覽結果

- 於第一數(A)輸入「12」與第二數(B)輸入「17」，按下「求最大公因數」按鈕，結果顯示在**TextBox3**中。



Form1

第一數(A) 12

第二數(B) 17

求最大公因數

最大公因數 1

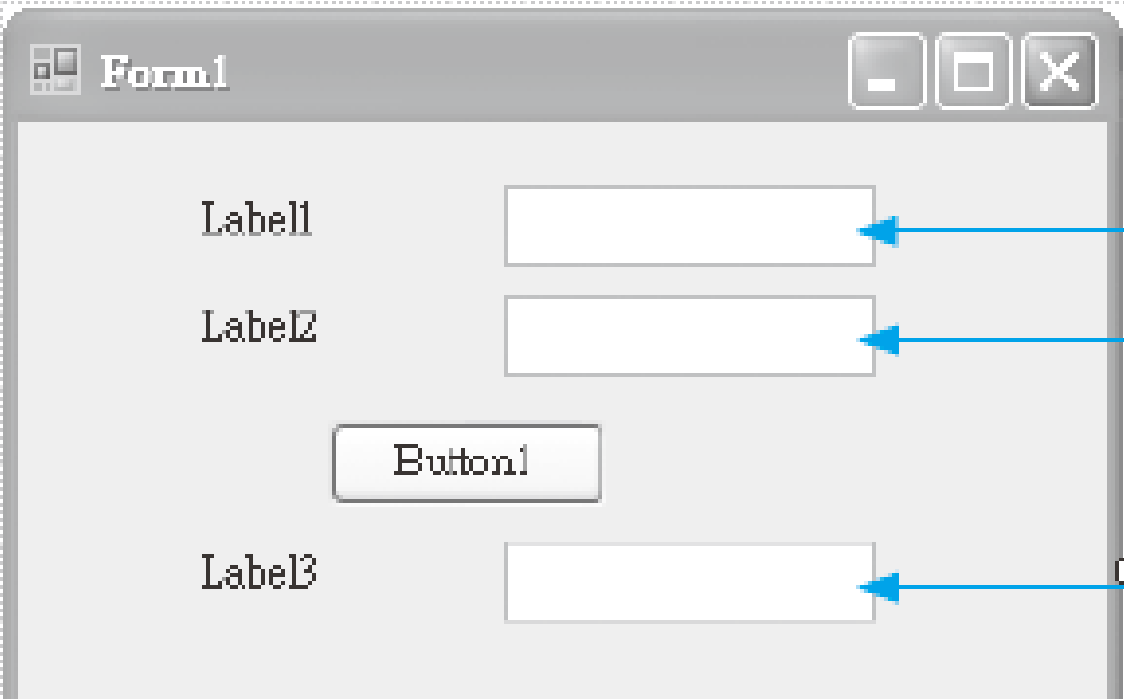
TextBox3

求兩數的最大公因數(ch5\最大公因數\)



- 表單配置

- 在表單中新增三個**TextBox**，三個**Label**與一個**Button**。



The screenshot shows a Windows Form titled "Form1" with a standard Windows XP-style title bar. Inside the form, there are three labels and three text boxes. The labels are "Label1", "Label2", and "Label3", arranged vertically on the left. The text boxes are "TextBox1", "TextBox2", and "TextBox3", arranged vertically on the right. A button labeled "Button1" is located between "Label2" and "Label3". Blue arrows point from each label to its corresponding text box: "Label1" to "TextBox1", "Label2" to "TextBox2", and "Label3" to "TextBox3".

求兩數的最大公因數(ch5\最大公因數\)

● 屬性設定

| 物件 | 屬性 | 設定值 |
|---------|------|---------|
| Label1 | Text | 第一數 (A) |
| Label2 | Text | 第二數 (B) |
| Label3 | Text | 最大公因數 |
| Button1 | Text | 求最大公因數 |



The screenshot shows a Windows form titled "Form1" with a standard title bar (minimize, maximize, close buttons). The form contains the following elements:

- A text box labeled "第一數 (A)" (First number (A)).
- A text box labeled "第二數 (B)" (Second number (B)).
- A button labeled "求最大公因數" (Calculate GCD).
- A text box labeled "最大公因數" (Greatest Common Divisor) for the result.

求兩數的最大公因數(ch5\最大公因數\)



- 程式碼新增與解說

- 點選「求最大公因數」按鈕，新增程式碼在**Button1_Click**函式中，如下方程式區塊。

| 行號 | 程式碼 |
|----|--|
| 1 | Public Class Form1 |
| 2 | Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click |
| 3 | Dim A As Integer=Val(TextBox1.Text) |
| 4 | Dim B As Integer = Val(TextBox2.Text) |
| 5 | Dim R As Integer |
| 6 | Do |
| 7 | R = A Mod B |
| 8 | A = B |
| 9 | B = R |
| 10 | Loop While (B <> 0) |
| 11 | TextBox3.Text = A |
| 12 | End Sub |
| 13 | End Class |

求兩數的最大公因數(ch5\最大公因數\)



- 解說

- 第3行：宣告**A**為整數，並初始化為**TextBox1**元件的輸入值。
- 第4行：宣告**B**為整數，並初始化為**TextBox2**元件的輸入值。
- 第5行：宣告**R**為整數。
- 第6行到第10行：使用**Do-While**迴圈，變數**R**為**A**除以**B**的餘數(第7行)，將**B**儲存入**A**(第8行)，將**R**儲存入**B**(第9行)，當**B**不等於0時繼續執行**Do-While**迴圈(第10行)。變數**A**、**B**與**R**舉例，如下張投影片。
- 第11行：將最大公因數**A**，顯示在**TextBox3**元件。

求兩數的最大公因數(ch5\最大公因數\)

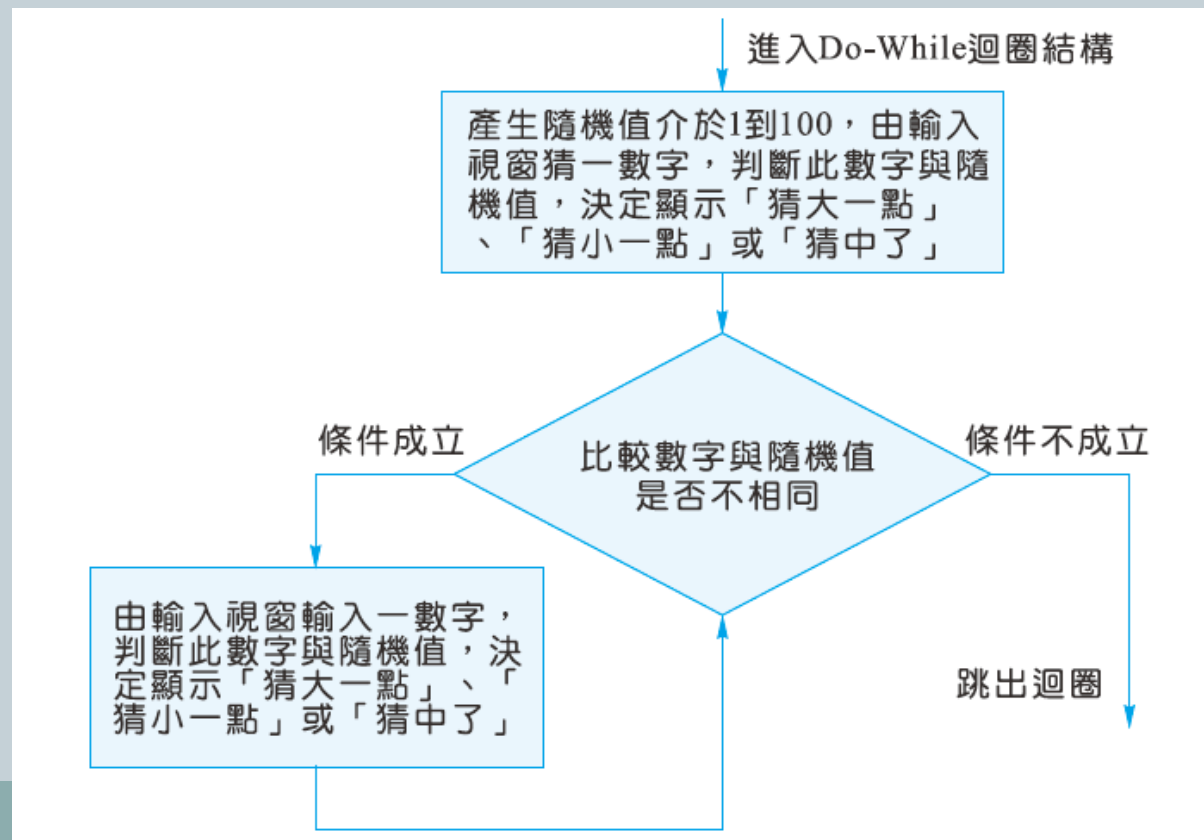


| 程式碼 | | R | A | B |
|---|--------|----|----------|--------|
| Dim A As Integer=12 Dim B As Integer=17 Dim R As Integer Do R = A Mod B A = B B = R Loop While (B <> 0) TextBox3.Text = A | 初始化 | | 12 | 17 |
| | 迴圈跑第一次 | 12 | 17 | 12 |
| | 迴圈跑第二次 | 5 | 12 | 5 |
| | 迴圈跑第三次 | 2 | 5 | 2 |
| | 迴圈跑第四次 | 1 | 2 | 1 |
| | 迴圈跑第五次 | 0 | 1=>最大公因數 | 0 =>結束 |

註：本程式可以不考慮A與B的大小關係，如本範例，A(12)小於B(17)迴圈一次後就交換。

猜數字(ch5\猜數字\)

- 大家是否玩過一個遊戲，兩人(A與B)一起玩，A心中想一數字，B猜A心中所想的數字，B每猜一次A就回答「猜大一點」、「猜小一點」或「猜中了」，當B猜到A所想的數字遊戲就結束
- 流程圖表示



猜數字(ch5\猜數字\)

- 預覽結果

- 按下「開始玩猜數字遊戲」按鈕。



- 出現以下輸入視窗，輸入一數字「50」，點選「確定」。



猜數字(ch5\猜數字\)

- 程式判斷需猜大一點，出現視窗「猜大一點」，點選「確定」。



- 輸入一數字「75」，再點選「確定」，如此一直根據電腦所回答之結果修正所猜數字，直到猜中為止。



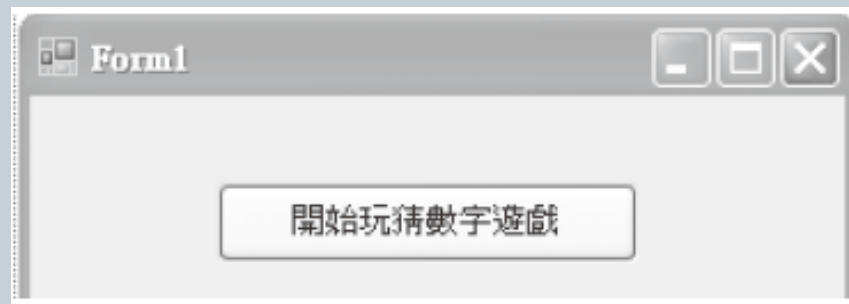
猜數字(ch5\猜數字\)

- 表單配置

- 在表單中新增一個Button。

- 屬性設定

| 物件 | 屬性 | 設定值 |
|---------|------|----------|
| Button1 | Text | 開始玩猜數字遊戲 |



猜數字(ch5\猜數字\)



InputBox函式

InputBox函式為**Visual Basic**所提供輸入視窗，允許使用者輸入字串，經由使用者按下「確定」鈕後將字串傳入**InputBox**函式回傳。例如：
num= InputBox(“請輸入一數字”)，執行到此會出現右圖。使用者輸入一數字後，按下「確定」鈕後，數字當成字串回傳**num**。

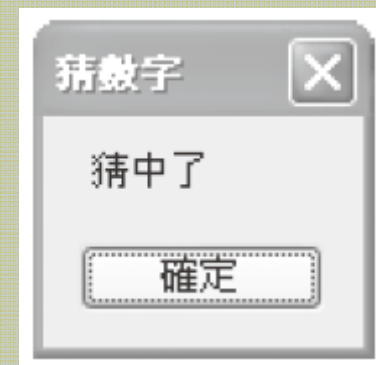
num= InputBox(“請輸入一數字”)



MsgBox函式

MsgBox函式為**Visual Basic**所提供輸出視窗，最簡單的輸出視窗為**MsgBox(“要顯示的字串”)**，例如：
MsgBox(“猜中了”)，結果於右圖。

MsgBox(“猜中了”)



猜數字 (ch5\猜數字\)

● 程式碼新增與解說

- 點選「開始玩猜數字遊戲」按鈕，新增程式碼在 **Button1_Click** 函式中，如下方程式區塊。

| 行號 | 程式碼 |
|----|--|
| 1 | Public Class Form1 |
| 2 | Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click |
| 3 | Randomize() |
| 4 | Dim target As Integer = CInt(Int((100 * Rnd()) + 1)) |
| 5 | Dim guess As Integer |
| 6 | Do |
| 7 | guess = Val(InputBox("請輸入一數字")) |
| 8 | If (target > guess) Then |
| 9 | MsgBox("猜大一點") |
| 10 | ElseIf (target < guess) Then |
| 11 | MsgBox("猜小一點") |
| 12 | Else |
| 13 | MsgBox("猜中了") |
| 14 | End If |
| 15 | Loop While (target <> guess) |
| 16 | End Sub |
| 17 | End Class |

猜數字(ch5\猜數字\)



● 解說

- 第3行：初始化隨機函式，隨機函式說明請參閱本範例之後「充電時間」說明。
- 第4行：**Rnd**函式會傳回大於等於0小於1的隨機值，所以「**(100 * Rnd()) + 1**」會產生大於等於1小於101的隨機值，**Int**與**CInt**將隨機值轉成整數，最後隨機值儲存入**target**。
- 第5行：宣告**guess**為整數。
- 第6行到第15行：使用**Do-While**迴圈，將**InputBox**元件輸入值被當成字串回傳，使用**Val**函式將字串轉換成數值儲存入變數**guess**(第7行)，比較**target**與**guess**，若**target**大於**guess**，產生視窗「猜大一點」；否則，若**target**小於**guess**，產生視窗「猜小一點」；否則，產生視窗「猜中了」(第8行到第14行)，當**target**與**guess**不相等時重複迴圈(第15行)。

猜數字(ch5\猜數字\)



● 解說

- 第3行：初始化隨機函式，隨機函式說明請參閱本範例之後「充電時間」說明。
- 第4行：**Rnd**函式會傳回大於等於0小於1的隨機值，所以「 $(100 * \text{Rnd}) + 1$ 」會產生大於等於1小於101的隨機值，**Int**與**CInt**將隨機值轉成整數，最後隨機值儲存入**target**。
- 第5行：宣告**guess**為整數。
- 第6行到第15行：使用**Do-While**迴圈，將**InputBox**元件輸入值被當成字串回傳，使用**Val**函式將字串轉換成數值儲存入變數**guess**(第7行)，比較**target**與**guess**，若**target**大於**guess**，產生視窗「猜大一點」；否則，若**target**小於**guess**，產生視窗「猜小一點」；否則，產生視窗「猜中了」(第8行到第14行)，當**target**與**guess**不相等時重複迴圈(第15行)。

隨機函式



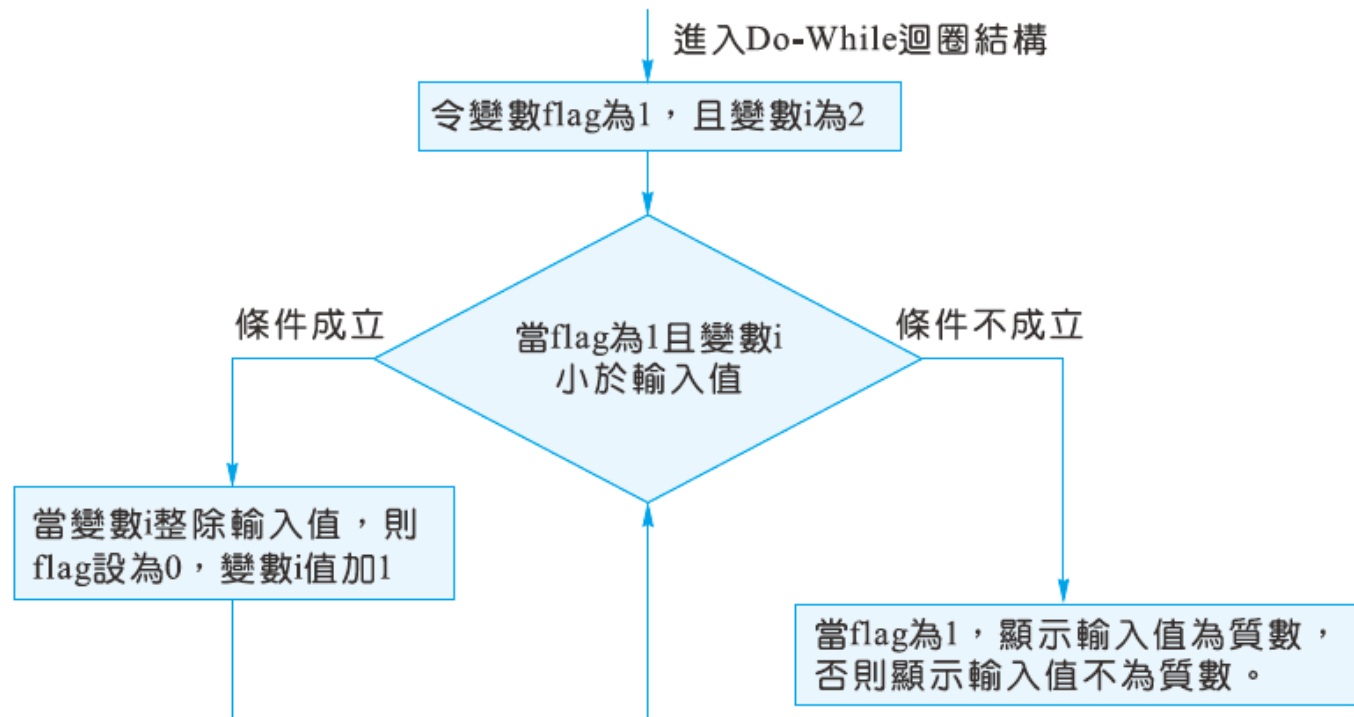
- VB的亂數函式包含**Randomize**函式與**Rnd**函式，先執行**Randomize**函式進行亂數產生器初始化，隨後執行**Rnd**函式，產生小於1且大於等於0的浮點數，將**Rnd**函式的產生值轉成上限值到下限值區間的整數，使用以下公式：
 - 「**CInt(Int((上限值 - 下限值 + 1) * Rnd() + 下限值))**」
 - 公式說明：**Int**函數為若傳入正數，則傳回整數部分，刪除小數部分。**CInt**將浮點數轉成整數。
- 亂數產生的公式如下

Randomize()

random = CInt(Int((上限值 - 下限值 + 1) * Rnd() + 下限值))

質數判斷(ch5\質數判斷\)

- 若某數的因數只有1與自己，沒有其他因數，稱此數為質數。程式中要判斷一個數是否是質數，就要判斷他的因數是否只有1與自己。
- 流程圖表示



質數判斷(ch5\質數判斷\)

- 預覽結果

- 輸入「13」，按下「質數判斷」按鈕，結果顯示在TextBox2中。



Form1

請輸入一數 13

質數判斷

判斷結果 13為質數

TextBox2

質數判斷(ch5\質數判斷\)

- 表單配置

- 在表單中新增兩個TextBox，兩個Label與一個Button。

The screenshot shows a Windows Form titled "Form1" with a standard Windows XP-style title bar. The form contains the following controls:

- Label1**: A label positioned to the left of **TextBox1**.
- TextBox1**: A text input box at the top right of the form.
- Button1**: A button labeled "Button1" located in the center of the form.
- Label2**: A label positioned to the left of **TextBox2**.
- TextBox2**: A text input box at the bottom right of the form.

Blue arrows indicate the mapping from the text labels to the text boxes: one arrow points from **Label1** to **TextBox1**, and another points from **Label2** to **TextBox2**.

質數判斷(ch5\質數判斷\)

- 屬性設定

| 物件 | 屬性 | 設定值 |
|---------|------|-------|
| Label1 | Text | 請輸入一數 |
| Label2 | Text | 判斷結果 |
| Button1 | Text | 質數判斷 |



The screenshot shows a Windows Form titled "Form1" with a standard Windows XP-style title bar (minimize, maximize, close buttons). The form contains the following elements:

- A label "請輸入一數" (Please enter a number) followed by a text input box.
- A button labeled "質數判斷" (Prime Number Judgment).
- A label "判斷結果" (Judgment Result) followed by a text input box.

質數判斷(ch5\質數判斷\)



● 旗標變數

- 在程式中可以自訂變數，並賦予變數意義，這個變數代表**程式的狀態**。
- 如本範例求質數所使用的**Flag**變數，預設為**1**，表示該數為質數，所有數都預設是質數，若發現有整除的因數，就將**Flag**變數改成**0**，表示該數為非質數。**程式最後測試Flag變數，若為1表示該數為質數，若為0表示該數為非質數**。經由**Flag**旗標變數儲存是否為**1**的狀態，程式最後藉此旗標變數決定輸入值是否為質數
- 就如同紅綠燈已經定義好了「紅燈停」與「綠燈行」的規則，所以開車就要遵守已經定義好的燈號規則。旗標變數就是程式的燈號，我們定義好了他的狀態規則，程式需遵守此規則，這就是旗標變數的功能。
- 程式解題過程中，旗標變數是個不錯的解題技巧，可以好好善用

質數判斷(ch5\質數判斷\)

- 程式碼新增與解說

- 點選「質數判斷」按鈕，新增程式碼在**Button1_Click**函式中，如下方程式區塊。

| 行號 | 程式碼 |
|----|--|
| 1 | Public Class Form1 |
| 2 | Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click |
| 3 | Dim Flag As Integer = 1 |
| 4 | Dim i As Integer = 2 |
| 5 | Do While ((Flag = 1) And (i < Val(TextBox1.Text))) |
| 6 | If ((Val(TextBox1.Text) Mod i) = 0) Then |
| 7 | Flag = 0 |
| 8 | End If |
| 9 | i = i + 1 |
| 10 | Loop |
| 11 | If (Flag = 1) Then |
| 12 | TextBox2.Text = TextBox1.Text & "為質數" |
| 13 | Else |
| 14 | TextBox2.Text = TextBox1.Text & "不為質數" |
| 15 | End If |
| 16 | End Sub |
| 17 | End Class |

質數判斷(ch5\質數判斷\)



● 解說

- 第3行：**Flag**宣告為整數變數，為本程式的旗標變數，若**Flag**為1表示為質數，若**Flag**為0表示不為質數。預設**Flag**為1，表示先視為質數。
- 第4行：**i**為整數變數，用於**While**迴圈，變數**i**從2開始依序指向所有小於輸入值的數，初始化為2。
- 第5~10行：當**Flag**為1且變數**i**小於輸入值(第5行)，繼續測試變數**i**是否可以整除輸入值(第6行)，若可以整除，則變數**i**為輸入值的因數，設定**Flag**為0，輸入值為非質數(第7行)。變數**i**值加1(第9行)，重複**While**迴圈回到第5行(第10行)。
- 第11~15行：若**Flag**為1，顯示輸入值「為質數」；否則，顯示輸入值為「不為質數」。