

第 6 章 循环结构程序设计

在编程中常常遇到这样的情况：某一类问题的计算和处理方法完全一样，只是要求重复计算多次，而每次使用的数据都是按照一定的规律变化的。例如，需要对一个班 30 名学生的成绩进行检查，将不及格者打印出来。类似这样的问题，就要用到循环结构。

程序设计中的循环结构（简称循环）是指在程序中从某处开始有规律地反复执行某一操作块（或程序块）的现象。被重复执行的操作块（或程序块）称为循环体，循环体的执行与否及次数多少视循环类型与条件而定。当然，无论何种类型的循环结构，其共同的特点是：必须确保循环体的重复执行能被终止（即非无限循环）。

VB 中常用的循环语句有 For...Next 语句和 Do...Loop 语句。For...Next 循环用于已知循环次数的情况，而 Do...Loop 循环主要用于不知道循环次数的情况，在给定的条件满足时执行循环体。

6.1 For...Next 循环语句

For...Next 循环语句按指定次数执行循环体，它在循环体中使用一个循环变量（计数器），每重复一次循环后，循环变量的值就会自动增加或者减少。

1. For...Next 语句的语法格式

For...Next 语句的语法格式为：

```
For <循环变量> = <初值> To <终值> [Step <步长> ]  
    [ <语句组 1> ]  
    [ Exit For ]  
    [ <语句组 2> ]  
Next [ <循环变量> ]
```

【说明】

- ① <循环变量> 为必要参数，是用作循环计数器的数值变量。该变量不能是数组元素。
- ② <初值> 和 <终值> 都是必要参数。<步长> 可以是正数或负数。当步长的值为 1 时，可以省略。
- ③ 可以省略 Next 语句中的 <循环变量>，但写上 <循环变量> 将提高程序的可读性。

2. For...Next 语句的执行过程

进入 For...Next 循环后，首先把 <初值> 赋给 <循环变量>，检查 <循环变量> 的值是否超过 <终值>。如果超过就停止执行循环体，跳出循环，执行 Next 后面的语句；否则执行一次循环体，然后把 <循环变量> + <步长> 的值赋给 <循环变量>，重复上述过程。

这里所说的“超过”有两种含义，即大于或小于：

- 当〈步长〉为正值时，检查〈循环变量〉的值是否大于〈终值〉。
 - 当〈步长〉为负值时，检查〈循环变量〉的值是否小于〈终值〉。
- 可以在循环中的任何位置放置任意个 Exit For 语句，随时退出循环。

3. For...Next 循环的循环次数

For...Next 循环遵循“先检查，后执行”的原则，即先检查〈循环变量〉是否超过〈终值〉，然后决定是否执行循环体。因此，在下列两种情况下，循环体不被执行：

- 当〈步长〉为正数，〈初值〉大于〈终值〉时。
- 当〈步长〉为负数，〈初值〉小于〈终值〉时。

因此，循环的最少执行次数为 0 次。

当〈初值〉等于〈终值〉时，不管〈步长〉是正数还是负数，均执行一次循环体。循环次数由〈初值〉、〈终值〉和〈步长〉3 个因素决定，可以通过下式计算：

$$\text{循环次数} = \text{INT}((\text{终值} - \text{初值}) / \text{步长} + 1)$$

如果计算出的循环次数小于或者等于 0，循环次数为 0，这时系统将不执行循环体。

4. For...Next 语句使用示例

【例 6-1】 设计程序，用 For...Next 语句求 $1 + 2 + 3 + \dots + 100$ 的值。

【分析】 采用累加的方法，用变量 s（累加器）来存放累加和（开始为 0），用变量 n 来存放加数（加到 s 中的数）。这里 n 又称为计数器，从 1 开始，到 100 结束。

设计步骤如下。

- 1) 建立应用程序用户界面并设置对象属性，如图 6-1 所示。
- 2) 编写事件代码。编写“计算”命令按钮 Command1 的 Click 事件代码：

```
Private Sub Command1_Click()
    Dim s As Integer, n As Integer
    s = 0 ' 累加器赋初值 0
    For n = 1 To 100 ' 初值为 1，终值为 100，步长为 1（省略）
        s = s + n ' 进行累加
    Next n
    Text1.Text = s ' 输出累加结果
End Sub
```

运行程序，结果如图 6-2 所示。

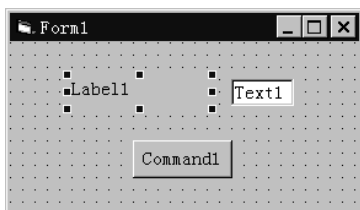


图 6-1 建立用户界面

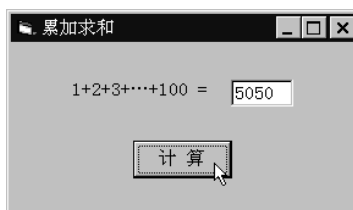


图 6-2 程序运行结果

【例 6-2】 输出 1000 以内所有能被 37 整除的自然数。

设计步骤如下。

1) 建立用户界面。在新建的窗体中增加一个文本框 Text1、一个命令按钮 Command1 和一个框架 Frame1。选中 Frame1，在其中增加一个标签 Label1，用来显示程序的说明。如图 6-3 (a) 所示。

2) 设置对象属性。设置文本框 Text1 的 MultiLine 属性值为 True，使文本框在运行时显示多行文本。设置文本框 Text1 的 ScrollBar 属性值为 2，使其具有垂直滚动条。

其他对象的属性设置如图 6-3 (b) 所示。

3) 编写事件代码。要想使每个数据换行输出，可以用一个回车加上换行符 (Chr(13) & Chr(10)) 来产生一个行断点，把所有符合要求的数都连接到变量 a 中 (见下面程序代码)。

编写“开始”命令按钮 Command1 的 Click 事件代码：

```
Private Sub Command1_Click()
```

```
    a = ""
```

```
    For n = 1 To 1000
```

```
        If n Mod 37 = 0 Then
```

```
            a = a & Str(n) & Chr(13) & Chr(10)          ' 用 Chr(13) & Chr(10)设置行断点
```

```
        End If
```

```
    Next
```

```
    Text1.Text = a
```

```
End Sub
```

运行程序，结果如图 6-3 (c) 所示。

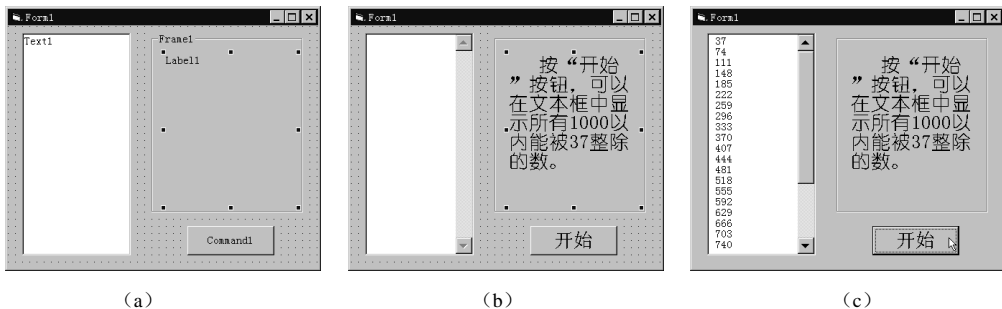


图 6-3 能被 37 整除的自然数

5. For...Next 语句的嵌套

For...Next 循环语句可以嵌套使用，嵌套层数没有具体限制，其基本要求是：

- 每个循环必须有一个唯一的变量作为控制变量。
- 内层循环必须完全放在外循环体内，内外循环不得互相交叉跨骑。

例如，下面的嵌套是错误的：

```
For a=1 To 5
```

```
    For b=3 To 9
```

```
        .....
```

```
    Next a
```

```
Next b
```

For...Next 循环的嵌套通常有以下 3 种形式。

① 一般嵌套形式。

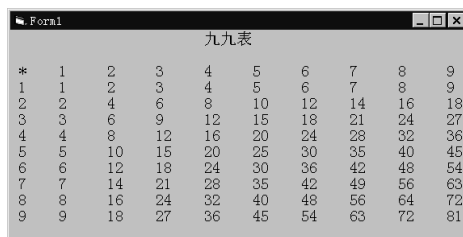
```
For a1=...  
  For a2=...  
    For a3=...  
      .....  
    Next a3  
  Next a2  
Next a1
```

② 形式 ① 中 Next 后面的 a1, a2, a3 可以省略不写。

③ 当内层循环与外层循环有相同的终点时，可公用一个 Next 语句，但是，控制变量名不能省略。例如：

```
For a=1 To 2  
  For b=2 To 3  
    For c=3 To 4  
      Print a , b , c  
    Next c , b , a
```

【例 6-3】 打印出如图 6-4 所示的乘法“九九表”。



*	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

图 6-4 打印乘法“九九表”

【分析】 “九九表”是一个 9 行 9 列的二维表，行和列都要变化，而且在变化中相互约束。这是一个二重循环问题。

直接在窗体上输出。窗体 Form 的 Load 事件代码为：

```
Private Sub Form_Load()  
  Show  
  FontSize = 12 ' 设置字号  
  Print Tab(25); "九九表" ' 输出标题  
  Print ' 输出空行  
  Print " * ";  
  For i = 1 To 9 ' 输出第一行数字 (1~9)  
    Print Tab(i * 6); i; ' 每列空 5 格，定位输出  
  Next i  
  Print ' 换行  
  For j = 1 To 9 ' 外层循环
```

```

Print j; " ";
For k = 1 To 9          ' 内层循环
    m = j * k          ' 计算乘积
    Print Tab(k * 6); m; " "; ' 定位输出
Next k
Print                  ' 换行
Next j
End Sub

```

6.2 Do...Loop 循环语句

For...Next 循环总是按指定的次数执行循环体。如果事先不知道循环次数，或循环的初值和终值不明了，则需要使用 Do...Loop 语句。Do...Loop 语句有两种语法形式：

- 前测型循环结构。
- 后测型循环结构。

6.2.1 前测型 Do...Loop 循环语句

1. 前测型 Do...Loop 的语法格式

前测型 Do...Loop 循环结构的循环特点是：先判断循环条件，根据条件决定是否执行循环体，执行循环体的最少次数为 0。

其语法格式为：

```

Do [{ While | Until } <条件> ]
    [ <语句组 1> ]
    [ Exit Do ]
    [ <语句组 2> ]

```

Loop

【说明】

- ① <条件> 是条件表达式，为循环的条件，其值为 True 或 False。
- ② <语句组> 是一条或多条命令（循环体），当（或直到）条件为真时被重复执行。

2. 前测型 Do...Loop 的执行过程

前测型 Do...Loop 循环语句先判断条件，再执行循环体。根据条件不同，可分为当型和直到型循环结构。

- 当型 Do While...Loop：当条件为真时，执行循环体；条件为假时，终止循环。
- 直到型 Do Until...Loop：当条件为假时，执行循环体；直到条件为真时，终止循环。

在 Do...Loop 循环体中，可以放置任意个数的 Exit Do 语句，随时跳出 Do...Loop 循环。Exit Do 语句通常用于条件判断之后，例如 If...Then 语句，在这种情况下，Exit Do 语句将控制权转移到紧接在 Loop 命令之后的语句。如果在嵌套的 Do...Loop 语句中使用 Exit Do，则 Exit Do 会将控制权转移到 Exit Do 所在位置的外层循环。

3. 前测型 Do...Loop 使用示例

【例 6-4】 用前测型 Do...Loop 语句，计算 $1 + 2 + 3 + \dots + 100$ 的值。
设计步骤如下。

- 1) 建立用户界面并设置对象属性，如图 6-5 (a) 所示。
- 2) 设置对象属性，参见图 6-5 (b)。

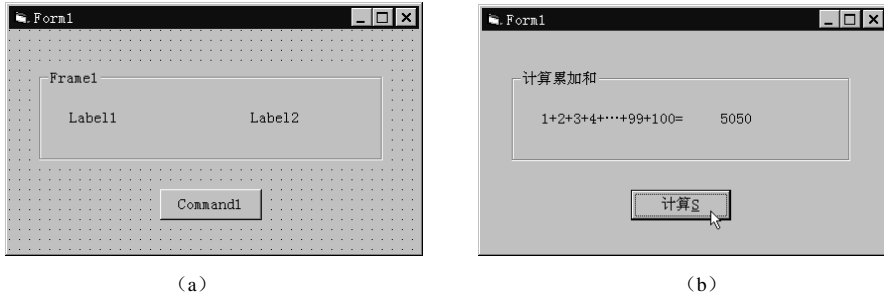


图 6-5 程序界面与运行结果

3) 编写事件代码。

采用当型循环结构编写“计算”命令按钮 Command1 的 Click 事件代码为：

```
Private Sub Command1_Click()  
    Dim S As Integer, n As Integer  
    S = 0 : n = 1           ' 累加器 S 赋初值 0，计数器 n 赋初值 1  
    Do While n <= 100     ' 当型循环  
        S = S + n         ' 累加和  
        n = n + 1        ' 计数器累加 1  
    Loop  
    Label2.Caption = S  
End Sub
```

“计算”命令按钮 Command1 的 Click 事件代码还可以改为直到型循环结构：

```
Private Sub Command1_Click()  
    Dim S As Integer, n As Integer  
    S = 0 : n = 1  
    Do Until n > 100     ' 直到型循环  
        S = S + n  
        n = n + 1  
    Loop  
    Label2.Caption = S  
End Sub
```

该事件代码还可以利用 Exit Do 语句来编写：

```
Private Sub Command1_Click()  
    Dim S As Integer, n As Integer  
    S = 0 : n = 1
```

```

Do
    S = S + n
    n = n + 1
    If n > 100 Then Exit Do           ' 如果 n>100, 则跳出循环
Loop
Label2.Caption = S

```

End Sub

运行程序，结果如图 6-5 (b) 所示。

【例 6-5】 已知 $s = 1 \times 2 \times 3 \times \dots \times n$ ，计算出 s 不大于 5000 时 n 的最大值。

【分析】 本题利用循环结构进行累乘运算。设计数器为 n ，累乘器 $s = s * n$ ，其循环条件是 $s \leq 5000$ 。由于求的是最大 n 值，输出语句应在循环体外。

设计步骤如下。

- 1) 建立用户界面并设置对象属性，如图 6-6 所示。
- 2) 编写事件代码。

编写“计算”命令按钮 Command1 的 Click 事件代码：

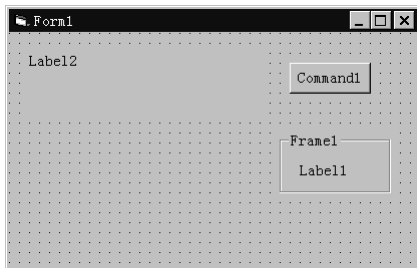
```

Private Sub Command1_Click()
    Dim n As Integer, s As Long
    CurrentY = Label2.Height + 200           ' 确定输出位置
    n = 1                                    ' 计数器赋初值 1
    s = 1                                    ' 累乘器赋初值 1
    Do While s <= 5000                       ' 循环条件
        n = n + 1                             ' 计数器累加 1
        s = s * n                             ' 累乘
        Print n, s                            ' 打印循环过程
    Loop
    Label1.Caption = "n = " & Str(n - 1)

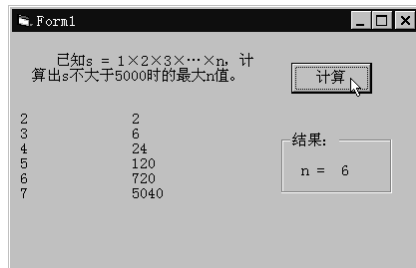
```

End Sub

运行程序，结果如图 6-6 所示。



(a)



(b)

图 6-6 程序界面与运行结果

【例 6-6】 输入一个正整数，利用 Do 循环判断是否为素数。

【分析】 所谓素数，是指除了 1 和该数本身外，不能被任何整数整除的数。判断一个

自然数 n ($n \geq 3$) 是否为素数, 只要依次用 $2 \sim \sqrt{n}$ 之间的整数作为除数去除 n , 若 n 不能被其中任何一个数整除, 则 n 即为素数。

设计步骤如下。

- 1) 建立用户界面并设置对象属性, 如图 6-7 所示。
- 2) 编写事件代码。编写“判定素数”命令按钮 Command1 的 Click 事件代码:

Private Sub Command1_Click()

```
Dim n As Long
Select Case Val(Text1.Text)
Case Is < 3
    MsgBox "请输入一个大于 2 的整数!", vbInformation + vbOKOnly, "注意"
Case Is > 2147483647
    MsgBox "此数太大!", vbInformation + vbOKOnly, "注意"
Case Else
    n = Val(Text1.Text)
    s = 0: i = 2
    Do While i <= Sqr(n) And s = 0
    If n Mod i = 0 Then
        s = 1
    Else
        i = i + 1
    End If
    Loop
    If s = 0 Then
        a = "是一个素数"
    Else
        a = "不是素数"
    End If
    Label1.Caption = Str(n) & a
End Select
Text1.SetFocus
```

End Sub

运行程序, 结果如图 6-7 所示。

【例 6-7】 输出 100~200 之间不能被 3 整除的数。

【分析】 可以用多种循环语句来实现。根据题意, 某数不能被 3 整除, 可以用 Mod 运算来完成, 即用 $x \text{ Mod } 3 \neq 0$ 来表示 x 不能被 3 整除。计数器从 100 开始, 到 200 结束。

设计步骤如下。

- 1) 建立用户界面并设置对象属性, 如图 6-8 (a) 所示。
- 2) 编写事件代码。

编写“开始”命令按钮 Command1 的 Click 事件代码:

Private Sub Command1_Click()

```
Dim x As Integer
x = 100
Do Until x > 200
    If x Mod 3 <> 0 Then
        Text1.Text = Text1.Text & Str(x) & Chr(13) & Chr(10)
    End If
    x = x + 1
Loop
```

End Sub

编写“清除”命令按钮 Command2 的 Click 事件代码：

Private Sub Command2_Click()

```
Text1.Text = ""
```

End Sub

运行程序，结果如图 6-8 (b) 所示。



图 6-7 判断素数



(a) (b)

图 6-8 用户界面与程序运行结果

6.2.2 后测型 Do...Loop 循环语句

后测型 Do...Loop 循环结构的执行特点是：先执行循环体，然后判断条件，根据条件决定是否继续执行循环，因此执行循环的最少次数为 1。

1. 后测型 Do...Loop 的语法格式

后测型 Do...Loop 循环结构的语法格式为：

Do

[<语句组 1>]

[Exit Do]

[<语句组 2>]

Loop [{While | Until} <条件>]

【说明】

① <条件> 是条件表达式，为循环的条件，其值为 True 或 False。

② <语句组> 是一条或多条命令（循环体），当（或直到）条件为真时，被重复执行。

2. 后测型 Do...Loop 的执行过程

后测型 Do...Loop 是先执行一次循环体后, 再进行条件判断, 分为当型和直到型两种:

- 当型 Do...While Loop: 当条件为真时, 继续执行循环体; 若条件为假, 则终止循环。
- 直到型 Do...Until Loop: 当条件为假时, 继续执行循环体; 直到条件为真时, 终止循环。

在 Do...Loop 循环体中, 可以放置任意个 Exit Do 语句, 随时跳出 Do...Loop 循环。

3. 后测型 Do...Loop 使用示例

【例 6-8】 输入有效数字的位数, 利用下述公式计算圆周率 π 的近似值:

$$\pi = 2 \cdot \frac{2}{\sqrt{2}} \cdot \frac{2}{\sqrt{2+\sqrt{2}}} \cdot \frac{2}{\sqrt{2+\sqrt{2+\sqrt{2}}}} \cdots$$

【分析】 首先找出公式中无穷乘积各项的规律。设第 n 项的分母为 p_n , 则第 $n+1$ 项的分母为 $p_{n+1} = \sqrt{2+p_n}$ 。若设前 n 项乘积为 S_n , 则前 n 项乘积为 $S_{n+1} = 2S_n / p_{n+1}$ 。

设计步骤如下。

- 1) 建立应用程序用户界面并设置对象属性, 如图 6-9 所示。
- 2) 编写程序代码。

根据流程图, 可以写出“计算”命令按钮 Command1 的 Click 事件代码为:

```
Private Sub Command1_Click()  
    Dim m As Integer  
    m = Val(Text1.Text)  
    p = 0#: s = 2#: e = 0.1 ^ m  
    Do  
        t = s : p = Sqr(2 + p) : s = s * 2 / p  
    Loop Until Abs(t - s) < 0.1 ^ m  
    f = String(m - 1, "#")  
    Text2.Text = Format(s, "0." & f)  
    Text1.SetFocus  
End Sub
```

文本框 Text1 的 GotFocus 事件代码如下:

```
Private Sub Text1_GotFocus()  
    Text1.SelStart = 0  
    Text1.SelLength = Len(Text1.Text)  
End Sub
```

【例 6-9】 输入两个正整数, 求它们的最大公约数。

【分析】 求最大公约数可以用“辗转相除法”, 方法如下。

- ① 以大数 m 作为被除数, 小数 n 作为除数, 相除后余数为 r 。
- ② 若 $r \neq 0$, 则 $m \leftarrow n$, $n \leftarrow r$, 继续相除得到新的 r 。若仍有 $r \neq 0$, 则重复此过程, 直到 $r=0$ 为止。
- ③ 最后的 n 就是最大公约数。

设计步骤如下。

1) 建立应用程序用户界面并设置对象属性，如图 6-10 (a) 所示。

2) 编写程序代码。根据流程图，编写“计算”命令按钮 Command1 的 Click 事件代码为：

```
Private Sub Command1_Click()
```

```
Dim m As Integer, n As Integer
```

```
m = Val(Text1.Text)
```

```
n = Val(Text2.Text)
```

```
If m < n Then
```

```
    t = m: m = n: n = t           ' 交换数据，使大数在前，小数在后
```

```
End If
```

```
Do                               ' 求最大公约数
```

```
    If n <= 0 Or m <= 0 Then     ' 检验数据范围
```

```
        MsgBox "请重新输入数据！"
```

```
        Exit Do
```

```
    End If
```

```
    r = m Mod n
```

```
    m = n
```

```
    n = r
```

```
Loop While r <> 0                ' 当 r<>0 时辗转相除
```

```
Label3.Caption = m              ' 输出结果
```

```
End Sub
```

运行程序，结果如图 6-10 (b) 所示。

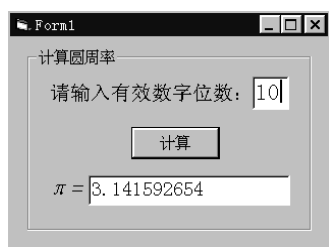


图 6-9 计算圆周率π



(a)



(b)

图 6-10 求两整数的最大公约数

【例 6-10】 设华氏温度为 h ，摄氏温度为 s ，已知将华氏温度转换为摄氏温度的公式为：

$$s = \frac{5}{9}(h - 32)$$

设计程序，实现华氏温度向摄氏温度的转换。

本例直接在窗体上载入。窗体 Form1 的 Load 事件代码为：

```
Private Sub Form_Load()
```

```
Dim h As String, s As Single, ts As String
```

```
Do
```

```
    h = InputBox("请输入华氏温度", "华氏温度")           ' 利用输入对话框输入华氏温度
```

```

If h <> "" Then
    s = Int((h - 32) * 5 / 9)          ' 计算摄氏温度
    MsgBox "摄氏温度为" & Str(s), 0 + 48 + 256, "转换为摄氏温度"
End If
Loop While h <> ""                  ' 若输入框中的值不为空，反复计算
End Sub

```

运行程序，结果如图 6-11 所示。



图 6-11 华氏温度转换为摄氏温度

6.3 列表框与组合框

如果需要向用户提供包含一些选项和信息的列表，由用户从中进行选择，可使用列表框或组合框。列表框与组合框在使用中是不相同的。

- 列表框：任何时候都能看到多个选项。
- 组合框：平时只能看到一个选项，单击组合框右端的下拉箭头按钮可以打开具有多个选项的列表。

6.3.1 列表框控件

列表框 (ListBox) 控件通过显示多个选择项，供用户选择其中一项，达到与用户对话的目的。如果有较多的选择项，超出所画的区域而不能一次全部显示时，VB 会自动加上垂直滚动条。

1. 列表框的属性

(1) 基本属性

列表框的常用基本属性有 Name, Enabled, Visible, Index 等。

(2) List 属性

该属性用于设置或返回列表中的选项。该属性是一个字符型数组，存放列表框中的项目。

注意：List 数组的下标是从 0 开始的，例如 List1.List(1) 表示列表框 List1 中第 2 项的值。

(3) Text 属性

该属性用于设置或返回列表中当前选项的文本内容。

(4) ListCount 属性

该属性用于返回列表框中项目的数量。ListCount - 1 表示列表中最后一项的序号。

(5) ListIndex 属性

该属性用于返回选中的列表项序号。如果未选中任何项，则 ListIndex 的值为 -1。

(6) Selected 属性

该属性用于在程序运行中使用代码来选定列表中的选项，例如：List1.Selected(2) = True 表示选中了 List1 中的第 3 项，如果为 False，则表示未被选中。

(7) Sorted 属性

该属性用于决定列表框中项目在程序运行期间是否按字母顺序排列显示。如果 Sorted 值为 True，则项目按字母顺序排列显示；如果 Sorted 值为 False，则按项目加入的先后顺序排列显示。

(8) MultiSelect 属性

- 0 - None: 禁止多项选择。这时在一个列表框中只能选择一项。
- 1 - Simple: 简单多项选择。单击鼠标左键或按空格键表示选定或取消选定一个选项。
- 2 - Extended: 扩展多项选择。按下 Ctrl 键同时单击鼠标左键或按空格键，表示选定或取消选定一个选项；按下 Shift 键同时单击鼠标左键，或者按下 Shift 键并且移动方向键，就可以从前一个选定的项扩展选择到当前选项，即选定多个连续项。

2. 列表框的方法

列表框中的选项可以简单地在设计状态下通过 List 属性设置，也可以在程序中使用 AddItem 方法来添加，用 Clear 或 RemoveItem 方法删除。

(1) AddItem 方法

AddItem 方法把一个项目加入列表框中，其语法格式如下：

〈对象〉. AddItem 〈字符串表达式〉 [, 〈位置〉]

【说明】

- ① AddItem 方法适用于列表框或组合框。
- ② 〈字符串表达式〉是要加入列表框或组合框中的项目。
- ③ 〈位置〉决定新增项目在列表框或组合框中的位置。如果省略，则新增项目添加在最后。对于第一个项目，位置为 0。

(2) Clear 方法

Clear 方法可清除列表框的所有内容，其语法格式如下：

〈对象〉. Clear

【说明】 Clear 方法中的对象可以是列表框、组合框或剪贴板。

(3) RemoveItem 方法

RemoveItem 方法可以从列表框中除去一个项目，其语法格式如下：

〈对象〉. RemoveItem 〈位置〉

【说明】 该方法适用于列表框或组合框。〈位置〉是被删除项目在列表框或组合框中的位置。对于第一个项目，位置为 0。

3. 列表框使用示例

【例 6-11】 在列表框中，显示 200 以内能被 6 整除的自然数。

【分析】 若某数 n 能被 6 整除，即 $n \bmod 6 = 0$ 。

设计步骤如下。

1) 建立用户界面并设置对象属性。在新建的窗体中增加一个列表框 List1，一个标签 Label1 和两个命令按钮 Command1，Command2。其中各对象的属性设置，如图 6-12 (a) 所示。

2) 编写事件代码。

编写“显示”命令按钮 Command1 的 Click 事件代码：

```
Private Sub Command1_Click()  
    List1.Clear ' 清空列表框的内容  
    For n = 1 To 200  
        If n Mod 6 = 0 Then List1.AddItem n ' 若 n 能被 6 整除，则添加到列表框中  
    Next n  
End Sub
```

编写“关闭”命令按钮 Command2 的 Click 事件代码：

```
Private Sub Command2_Click()  
    Unload Me  
End Sub
```

运行程序，结果如图 6-12 (b) 所示。

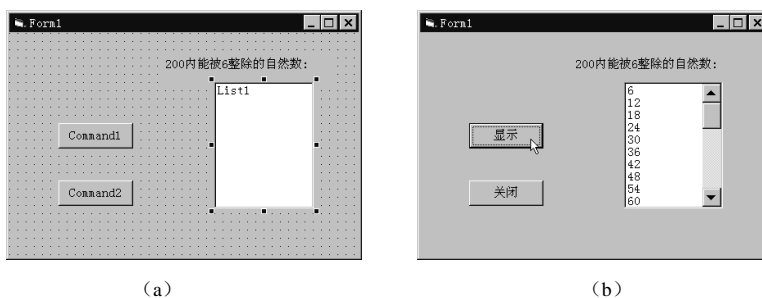


图 6-12 200 内能被 6 整除的自然数

【例 6-12】 从文本框中输入或从列表框中选择姓名，并显示出来，如图 6-13 所示。



图 6-13 从文本框中输入或从列表框中选择姓名

设计步骤如下。

1) 建立应用程序用户界面并设置对象属性。新建一个工程，进入窗体设计器，增加一

个框架 Frame1 和一个标签 Label1。激活 Frame1 后，在其中增加一个列表框 List1，一个文本框 text1 和一个命令按钮 Command1。修改对象属性参见表 6-1。

表 6-1 属性设置

对象	属性	属性值	说明
Frame1	Caption	请选择或输入姓名:	
Label1	Caption	您所选择的代表是:	
Text1	Text		清空
Command1	Caption	确定	

设置列表框 List1 的属性，在其中依次加入：“陈高阳”、“赵世杰”、“李民维”、“马英丽”、“杨广民”、“李灵君”、“陈吉至”（每输完一项，按 Ctrl+Enter 键可输入下一项）。

2) 编写程序代码。

编写列表框 List1 的 Click 事件代码：

```
Private Sub List1_Click()
```

```
Text1.Text = List1.Text
```

```
End Sub
```

编写文本框 Text1 的 Change 事件代码：

```
Private Sub Text1_Change()
```

```
Label1.Caption = "您所选择的代表是： "
```

```
End Sub
```

编写命令按钮 Command1 的 Click 事件代码：

```
Private Sub Command1_Click()
```

```
Text1.SelStart = 0
```

```
Text1.SelLength = Len(Text1.Text)
```

```
Label1.Caption = "您所选择的代表是： " + Text1.Text
```

```
End Sub
```

在上面的例子中，列表框中的各项数据是设计时在属性窗口中设置的。下面的例子说明如何在程序运行中向列表框添加新的项目，或移去列表框中的选定项。

【例 6-13】 在列表框之间移动数据，如图 6-14 所示。

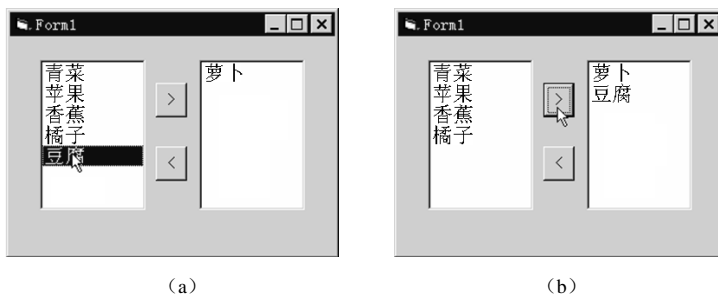


图 6-14 在列表框中移动数据

设计步骤如下。

1) 建立应用程序用户界面并设置对象属性。新建一个工程，进入窗体设计器，增加两

个列表框 List1, List2 和两个命令按钮 Command1, Command2。然后设置对象属性见表 6-2。

表 6-2 属性设置

对 象	属 性	属 性 值	说 明
Command1	Caption	>	“>” 按钮
Command2	Caption	<	“<” 按钮

2) 编写程序代码。

编写窗体的 Load 事件代码:

Private Sub Form_Load()

```
List1.AddItem "青菜"
List1.AddItem "萝卜"
List1.AddItem "豆腐"
List1.AddItem "苹果"
List1.AddItem "香蕉"
List1.AddItem "橘子"
```

End Sub

编写命令按钮 Command1 的 Click 事件代码:

Private Sub Command1_Click()

```
List2.AddItem List1.Text
List1.RemoveItem List1.ListIndex
```

End Sub

编写命令按钮 Command2 的 Click 事件代码:

Private Sub Command2_Click()

```
List1.AddItem List2.Text
List2.RemoveItem List2.ListIndex
```

End Sub

另外, 列表框还允许选择多个选项。

【例 6-14】 修改例6-13, 允许从一个列表框中将选中的多个选项移至另一个列表框中, 如图 6-15 所示。

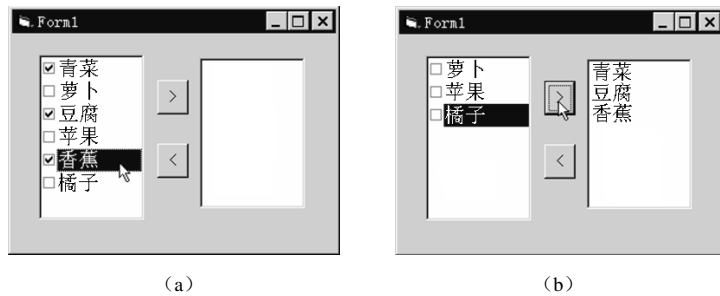


图 6-15 双列表框

设计步骤如下。

1) 修改对象属性参见表 6-3。

表 6-3 属性设置

对 象	属 性	属 性 值	说 明
List1	Style	1 - Checkbox	具有复选框风格，可以复选
List2	MultiSelect	2 - Extended	允许多项复选

2) 修改程序代码。

命令按钮 Command1 的 Click 事件代码：

```

Private Sub Command1_Click()
    i = 0
    Do While i < List1.ListCount - 1
        If List1.Selected(i) = True Then
            List2.AddItem List1.List(i)
            List1.RemoveItem i
        Else
            i = i + 1
        End If
    Loop
End Sub

```

命令按钮 Command2 的 Click 事件代码：

```

Private Sub Command2_Click()
    i = 0
    Do While i < List2.ListCount - 1
        If List2.Selected(i) = True Then
            List1.AddItem List2.List(i)
            List2.RemoveItem i
        Else
            i = i + 1
        End If
    Loop
End Sub

```

6.3.2 组合框控件

组合框（ComboBox）是组合了列表框和文本框的特性而形成的一种控件。组合框是一种独立的控件，它兼有列表框和文本框的功能，可以像列表框一样，让用户通过鼠标选择所需的项目；也可以像文本框一样，用输入的方式选择项目，但输入的内容不能自动添加到列表框中。若用户选中列表框中的某项，该项内容自动装入文本框中。组合框比列表框占用的屏幕空间要小。

列表框的属性基本上都可用于组合框，此外组合框还有一些自己的属性。

组合框有 3 种不同风格，即下拉组合框、简单组合框和下拉列表框。组合框的风格由 Style

属性值决定，其值分别为 0，1，2。3 种组合框如图 6-16 所示。

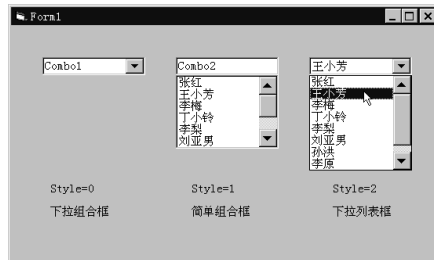


图 6-16 Style 属性的使用

1. 下拉组合框

下拉组合框的 Style 属性值为 0（默认），显示在屏幕上的仅是文本编辑框和一个下拉箭头按钮。执行时，用户可用键盘直接在文本框中输入内容，也可单击右边的下拉箭头按钮，打开列表框进行选择，选中的内容将显示在文本框中。

这种组合框允许用户输入不属于列表内的选项。当用户再次单击下拉箭头按钮时，下拉出来的列表就会消失，仅显示文本框。

2. 简单组合框

简单组合框的 Style 属性值为 1。它列出了所有的项目供用户选择，右边没有下拉箭头按钮，列表框不能被收起和拉下，与文本框一起显示在屏幕上。可以在文本框中用键盘输入列表框中没有的选项。

注意，必须用鼠标拖动滚动条才能显示全部项目。

3. 下拉列表框

下拉列表框的 Style 属性值为 2。其功能与下拉组合框类似，区别是，在下拉列表框中不能输入列表框中没有的选项。

4. 组合框使用示例

【例 6-15】 利用组合框设计“自动抽奖机”。

设计步骤如下。

1) 建立应用程序用户界面并设置对象属性。新建一个工程，进入窗体设计器，增加一个组合框 Combo1，两个标签 Label1，Label2 和两个命令按钮 Command1，Command2。将 Combo1 的 Style 属性改为 0，其他属性的设置参见图 6-17。

2) 编写事件代码。

编写组合框 Combo1 的 KeyPress 事件代码：

```
Private Sub Combo1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then           ' 按 Enter 键
        Combo1.AddItem Combo1.Text, 0   ' 接收输入的号码
        Combo1.SelStart = 0           ' 设置组合框的起始位置
        Combo1.SelLength = Len(Combo1.Text) ' 设置组合框的长度
    End If
```