第3章 分支结构程序设计

本章内容

基础部分：

- 关系、逻辑运算符及其表达式。
- If 语句与 If 语句的嵌套；Select Case 语句。
- 分支结构的流程图。
- 单选按钮、复选框、框架、直线、形状等控件的使用。
- 求 3 个数的最大(小)值。

提高部分：

- 再论常用控件：单选按钮、复选框、框架、直线和形状。
- 贯穿实例。

各例题知识要点

例 3.1　关系运算符及表达式。

例 3.2　逻辑运算符及表达式。

例 3.3　用 If 语句设计分支结构程序；If 语句格式及流程图。

例 3.4　无 Else 分支的 If 语句；计算分段函数值。

例 3.5　复选框；多行文本框；与文本字体格式相关的属性；颜色常量。

例 3.6　单选按钮；求三个数中的最大(小)值算法。

例 3.7　直线控件；控件坐标表示；图像的循环变换和移动。

例 3.8　框架；嵌套的 If 语句。

例 3.9　形状控件；用 Select Case 语句设计多分支结构；Select Case 语句格式和流程图。

例 3.10　If 语句和 Select Case 语句的联合使用；Activate、Deactivate 事件。

贯穿实例　书店图书管理系统(3)。

分支结构是三种基本结构之一，因分支结构在实际应用中被广泛使用，所以本章将详细地介绍分支结构的实现方法。

3.1 关系、逻辑运算符与表达式

分支结构的特点是：首先需要对给定的条件进行判断，然后根据判断结果选择执行某一操作。在表示判断条件时，经常会用到关系表达式和逻辑表达式，下面分别介绍这两类表达式。

3.1.1 关系运算符与表达式

VB 提供了表 3-1 所示的六种关系运算符。

表 3-1 关系运算符

运 算 符	含 义	举 例	例子作用
>	大于	"ab">"aB"	字符串"ab"是否大于"aB"
>=	大于等于	a>=0	a 中的值是否大于等于 0
<	小于	a<0	a 中的值是否小于 0
<=	小于等于	−3<=0	−3 是否小于等于 0
=	等于	a−b=0	a 与 b 的差是否等于 0
<>	不等于	a<>b	a 与 b 的值是否不相等

说明：

(1) 各关系运算符的优先级相同。关系运算符隐含“是否”的含义，例如，"ab">"aB" 表示字符串"ab"是否大于"aB"。关系运算就是对运算符左右两边的表达式进行比较的运算。

(2) 由关系运算符连接表达式组成关系表达式，被连接的表达式可以是数值型、字符型和日期型。关系表达式的运算结果只能是“True”(真)或“False”(假)。

(3) VB 6.0 还提供“Like”和“Is”运算符，由于篇幅有限本书不做介绍，请查看相关书籍或帮助。

【例 3.1】 写出以下各关系表达式的值：

(1) True<>1。

(2) 3>x>=0 (假设 x 为 Integer 类型变量，其值为 2)。

(3) "abc"<"aBcd"。

【解】 本例是关系表达式的运算，各表达式的值是(1)True；(2)False；(3)False。

说明：

(1) 表达式计算遵循“从高到低、从左到右”的原则顺次执行，即在同一表达式中，首

先执行具有较高优先级的运算符，若两运算符优先级相同，则按照先左后右的顺序执行。VB 中常用运算符的优先级如表 3-2 所示。在书写表达式时应尽可能使用运算符“()”，以明确表示表达式的运算顺序。

表 3-2　常用运算符优先级

优　先　级	运　算　符	说　　明
高	()	小括号
↓	^	算术运算符
↓	－（负号）	↓
↓	*，/	↓
↓	\	↓
↓	Mod	↓
↓	＋，－	↓
↓	&	字符串运算符
↓	＞，＞＝，＜，＜＝，＝，＜＞	关系运算符
↓	Not	逻辑运算符
↓	And	↓
低	Or	↓

（2）在 VB 中，True 对应数值－1，False 对应数值 0，因此也可对 True 和 False 进行比较。表达式“True＜＞1”等价于“－1＜＞1”，所以值为 True。

（3）在表达式“3＞x＞＝0”中，由于两个关系运算符的优先级相同，所以计算应从左向右进行，其过程是：先计算“3＞x”，结果为 True；再计算“True＞＝0”，即“－1＞＝0”，结果为 False，因此表达式“3＞x＞＝0”的值为 False。由此可以看到，表达式“3＞x＞＝0”在语法上并不存在错误，但其表达的逻辑含义却已不同于代数式本身。

（4）字符型数据的比较规则是：按从左到右的顺序，将两个字符串中对应位置上的字符一一进行比较。无论两字符串长度是否相等，一旦在比较过程中出现对应位置字符不等，则以这对字符的 ASCII 码大小决定字符串的大小，即具有较大 ASCII 码值的字符，其所在的字符串较大，与后续字符的多少、大小无关。在字符串“abc”与“aBcd”中，虽然“aBcd”的长度大于“abc”，但按照字符数据的比较规则，在第 2 个字符位置出现不同字符，此时“b”的 ASCII 码为 98，“B” 的 ASCII 码为 66，所以“b”所在的字符串大，因而表达式 "abc"＜"aBcd"的值为 False。

常用字符的 ASCII 码值参见附录 A。

3.1.2　逻辑运算符与表达式

VB 提供表 3-3 所示的逻辑运算符。

表 3-3　逻辑运算符

运算符	含　义	举　　例	例 子 作 用
Not	逻辑非	Not(x<>0)	对 x<>0 的结果取反
And	逻辑与	x<=3 And x>0	对 x<=3 的结果和 x>0 的结果进行“与”运算
Or	逻辑或	x>3 Or x<-3	对 x>3 的结果和 x<-3 的结果进行“或”运算

此外，还提供其他三个逻辑运算符：“Xor”、“Eqv”和“Imp”，限于篇幅本书不做介绍。

说明：

(1) 由逻辑运算符连接表达式组成逻辑表达式，被连接的表达式可以是关系表达式或由逻辑值组成的表达式；逻辑表达式的运算结果也只能是 True 或 False。

(2) 逻辑运算是对运算符左右两边的逻辑值进行逻辑判断的运算。其中，“逻辑与”运算的特点是：只有当其两侧的逻辑值同为“真”时，运算结果才为“真”；“逻辑或”运算的特点是：其两侧的逻辑值中只要有一个为“真”，运算结果就为“真”；“逻辑非”运算，则是对当前值的取反运算。常用逻辑运算符的运算规则如表 3-4 所示，其中 a、b 均代表逻辑值。

表 3-4　逻辑运算真值表

a	b	a And b	a Or b	Not a	Not b
True	True	True	True	False	False
True	False	False	True	False	True
False	True	False	True	True	False
False	False	False	False	True	True

【例 3.2】 写出以下各逻辑表达式的值：

(1) 3>x And x>=0(假设 x 为 Integer 类型变量，且值为 2)。

(2) x>5 Or 0(假设 x 为 Integer 类型变量，且值为 0)。

(3) True<>1 And False。

(4) Not(True=0)。

【解】 (1)True；(2)False；(3)False；(4)True

说明：

(1) “3>x And x>=0”为逻辑表达式，其运算顺序是：先计算关系表达式“3>x”和“x>=0”的值，结果均为 True；再对表达式“True And True”进行逻辑与运算，其结果为 True，因此该表达式的值为 True。

注意：表达式“3>x And x>=0”与“3>x>=0”的意义不同。

(2) 在表达式“x>5 Or 0”中，“x>5”的值是 False，0 等价于逻辑值 False，所以该表达式的值是 False。

(3) 在表达式“True<>1 And False”中，按优先级先计算“True<>1”，值为 True，而“True And False”的值为 False。

(4) 由于 True 对应数值－1，因而表达式“True＝0”的值是 False；对 False 再执行 Not 运算后的结果为 True。

3.2 If 语 句

在实际应用中，经常遇到条件选择的问题。例如，人们在计划周末活动时，可能的安排是“如果天气好，就去爬香山，否则就去健身房游泳”。分支结构就是利用计算机语言描述这种分支现象，即通过比较和判断决定采取何种操作。在 VB 中，通常使用 If 语句或 Select Case 语句解决这类问题。

3.2.1 使用 If 语句处理简单分支问题

【例 3.3】 If 语句示例。在窗体上添加 1 个文本框、3 个标签和 1 个命令按钮。程序运行时，在文本框中输入一个整数后单击“判断”按钮，在黄色标签中显示其奇、偶性，如图 3-1 所示。

图 3-1 判断奇、偶性

【解】 根据题意，对文本框 txtInput 中的数据进行判断，结果显示在标签 lblValue 中。程序代码如下：

```
Private Sub cmdJudge_Click()          '单击"判断"按钮
    Dim a As Integer
    a=Val(txtInput.Text)
    If a Mod 2=0 Then                 'If 语句开始
        lblValue.Caption="偶数"
    Else
        lblValue.Caption="奇数"
    End If                            'If 语句结束
End Sub
```

程序说明：

(1) cmdJudge_Click 事件的执行过程是：先得到文本框 txtInput 中输入的数据，转换成对应的数值型数据后赋给变量 a；然后对 a 进行逻辑判断，如果 a Mod 2＝0 的值为 True，即 a 的值能被 2 整除，标签中显示“偶数”，否则显示“奇数”。该事件过程的流程图如图 3-2 所示。

(2) 本例是通过 If 语句实现分支选择的，由图 3-2 可以看出，语句 lblValue. Caption＝"偶数"和 lblValue. Caption＝"奇数"分别处于两个不同的分支中。表达式 a Mod 2＝0 为判断条件，当其值为“真”时，执行 Then 后面的语句 lblValue. Caption＝"偶数"；当其值为“假”时，执行 Else 后面的语句 lblValue. Caption＝"奇数"。由此可以看到，在分支结构中，根据判断的结果，只能执行其中的一个分支。

(3) 程序运行时应分别输入偶数值和奇数值，以判断输出结果是否正确，不能只验证

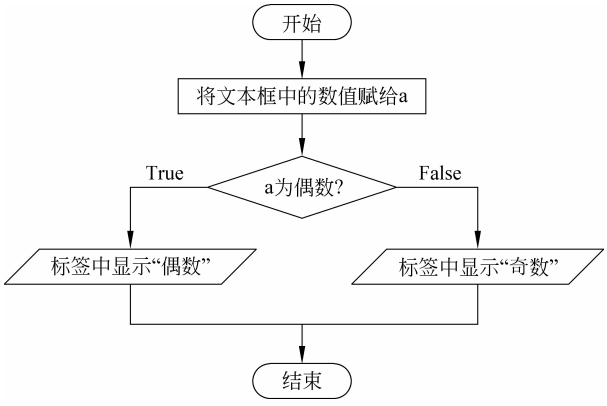


图 3-2　cmdJudge_Click 事件过程的流程图

其中一种情况(偶数或奇数)后就认为程序是正确的。

(4) If 语句的一般形式是:

```
If  表达式  Then
    语句组 1
[Else
    语句组 2 ]
End If
```

其中的语句组 1 和语句组 2 既可以由多条语句构成,也可以是单一的语句。

说明:

① If、Then、Else 是系统保留字,用“[ ]”括起来的是可省略项。

② 语句组 1 必须从新的一行开始书写,不能与 Then 在同一行,Else 和 End If 要求分别独占一行。

③ If 语句的执行流程如图 3-3 所示。如果“表达式”的值为“真”,则执行“语句组 1”,否则执行“语句组 2”。

在解决实际问题时,可根据具体情况省略 Else 分支的内容,如例 3.4。

【例 3.4】 无 Else 分支的 If 语句示例。计算以下分段函数的值。

$$y=\begin{cases} x^3+1 & (x\leqslant 0)\\ 2 & (0<x\leqslant 2)\\ 5x & (x>2)\end{cases}$$

在窗体上添加 1 个文本框和 3 个标签。程序运行时,在文本框中输入整数 x 值时,立刻计算出相应的函数值 y,并显示在黄色标签中,如图 3-4 所示。

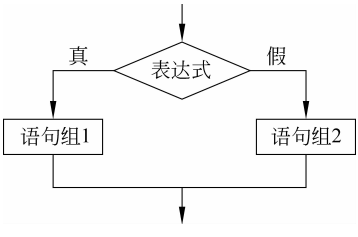


图 3-3　If 语句的执行过程

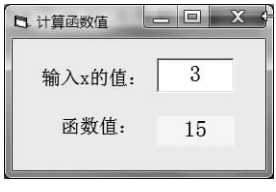


图 3-4　计算函数值

【解】 本例在文本框的 Change 事件中编写计算函数值的代码。程序代码如下：

```
Private Sub txtInput_Change()
    Dim x As Integer
    Dim y As Long
    x=Val(txtInput.Text)
    If x<=0 Then                    '根据 x 的值,计算 y 的值
        y=x ^ 3+1
    End If
    If x>0 And x<=2 Then
        y=2
    End If
    If x>2 Then
        y=5 * x
    End If
    lblValue.Caption=y
End Sub
```

程序说明：

本例使用三个 If 语句，分别处理三种不同 x 条件下的函数值。例如，在第一个 If 语句中，只给出了当“x＜＝0”为真时的具体操作，而“x＞0”，即 Else 的情况则没有给出，什么也不做。另外两个 If 语句与其结构相同。该过程的执行流程如图 3-5 所示。

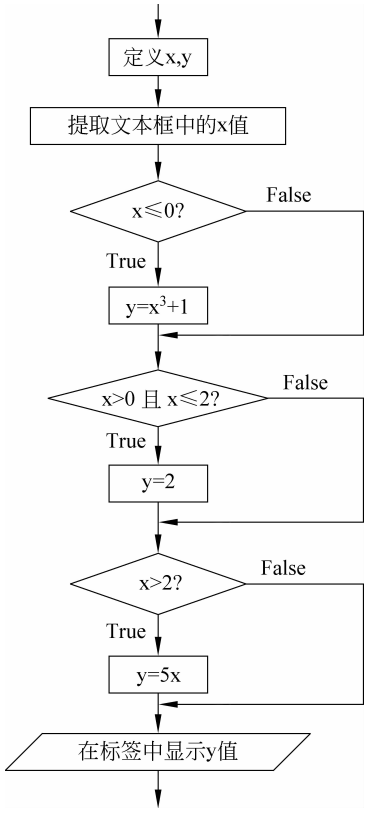


图 3-5 txtInput_Change 的执行过程

分支结构逻辑性较强，使用时有一定的难度，下面再介绍一些应用实例。

图 3-6　复选框的使用

【例 3.5】 在窗体上添加 1 个标签、1 个文本框、4 个复选框和 1 个命令按钮。程序运行时，在文本框中输入文字并根据需要选择不同复选框。单击“确定”按钮，根据复选框选择情况设置文本框的字体、字型、大小和颜色，如图 3-6 所示。

【解】 复选框控件在工具箱中的图标为☑，通过 Caption 属性设置其标题信息。为了在文本框中输入多行文本，设置其 Multiline 属性为 True，ScrollBars 属性为 2-Vertical。程序代码如下：

```
Private Sub cmdOk_Click()                          '单击"确定"按钮
    If chkFnt.Value=1 Then                         '若选中"字体"复选框,则
        txtInput.FontName="宋体"                   '文本框的字体设置为宋体
    Else                                           '否则
        txtInput.FontName="黑体"                   '文本框的字体设置为黑体
    End If
    If chkItc.Value=1 Then                         '若选中"斜体"复选框,则
        txtInput.FontItalic=True                   '文本框的字型设置为斜体
    Else                                           '否则
        txtInput.FontItalic=False                  '字型设置为正常字型
    End If
    If chkSz.Value=1 Then                          '若选中"大小"复选框,则
        txtInput.FontSize=18                       '文本框的文字大小设置为 18
    Else                                           '否则
        txtInput.FontSize=24                       '文字大小设置为 24
    End If
    If chkCol.Value=1 Then                         '若选中"颜色"复选框,则
        txtInput.ForeColor=vbRed                   '文本框的文本颜色设置为红色
    Else                                           '否则
        txtInput.ForeColor=vbBlack                 '文本颜色设置为黑色
    End If
End Sub
```

程序说明：

(1) 复选框有三种状态，其当前状态由 Value 属性表示：0—未选中；1—选中；2—选中但不可用(呈灰色)。本例中，在设计阶段将四个复选框的 Value 属性均设置成 0。

(2) 复选框控件的特点是：在多个复选框中可以同时选中多个；单击复选框时，其状态在“选中”与“未选中”之间切换。

(3) 文本框的 Multiline 属性设置为 True 时，表示在文本框中可以输入多行信息；为 False 时只能输入单行信息。

(4) 当文本框的 Multiline 属性为 True 时,可设置文本框的 ScrollBars(是否添加滚动条)属性。本例中设其属性值为 2-Vertical,表示在文本框中添加垂直滚动条。

(5) 本例题中使用了四个复选框,并通过四个 If 语句分别处理各复选框的选择情况。以第 1 个 If 语句为例,当 chkFont. Value 的值为 1(即选中"字体"复选框)时,执行 Then 后面的语句,将文本框字体设置为"宋体",否则执行 Else 后面的语句,将文本框字体设置为"黑体"。

(6) 文本框的 FontName、FontItalic、FontBold 和 FontSize 属性分别表示其显示文本的字体名称、文字是否斜体、文字是否加粗以及文字的大小。其中 FontItalic 和 FontBold 的属性值为 True 或 False,值为 True 时表示斜体或加粗。文本框还有 FontStrikethru 和 FontUnderline 属性,分别设置文字是否具有删除线和下划线。

(7) vbRed 和 vbBlack 是 VB 提供的符号常量,分别表示红色和黑色。表 3-5 列出了各颜色常量所对应的颜色。

表 3-5 颜色常量与颜色对照表

符号常量	描述颜色	值	符号常量	描述颜色	值
vbBlack	黑色	&H0	vbBlue	蓝色	&HFF0000
vbRed	红色	&HFF	vbMagenta	洋红色	&HFF00FF
vbGreen	绿色	&HFF00	vbCyan	青色	&HFFFF00
vbYellow	黄色	&HFFFF	vbWhite	白色	&HFFFFFF

【例 3.6】 求最大数与最小数。在窗体上添加 3 个文本框、3 个标签和 2 个单选按钮。程序运行时,输入 3 个整数,单击"最大数"单选按钮,在黄色标签中显示其中的最大数,如图 3-7 所示;单击"最小数"单选按钮,则显示其中的最小数。

图 3-7 求最大数与最小数

【解】 本例中用到新的控件——单选按钮,它在工具箱中的图标是。通常单选按钮成组出现,其特点是:在同组单选按钮中,只能有一个处于"选中"状态;每次单击单选按钮时,将使其处于"选中"状态,而同组中其他单选按钮则自动变成"未选中"状态。单选按钮只有两种状态,由 Value 属性表示:True—选中,False—未选中。

编程点拨:

为了在三个数中求最大数,设计如下算法:

(1) 定义变量 max,用于存放当前找到的最大值。

(2) 假设第 1 个数就是三个数中的最大值,将其存放到 max 中。

(3) 将第 2 个数与 max(即第 1 个数)进行比较。如果第 2 个数比 max 大,则将第 2 个数作为最大值放入 max 中,否则什么也不做。此时 max 中存放的是前两个数中的较大值。

(4) 将第 3 个数与 max(前两个数中的较大者)进行比较。如果第 3 个数比 max 大,

则将第 3 个数作为最大值放入 max 中，否则什么也不做。此时 max 中一定存放的是三个数中的最大值。

上述算法可用图 3-8 所示的流程图表示。

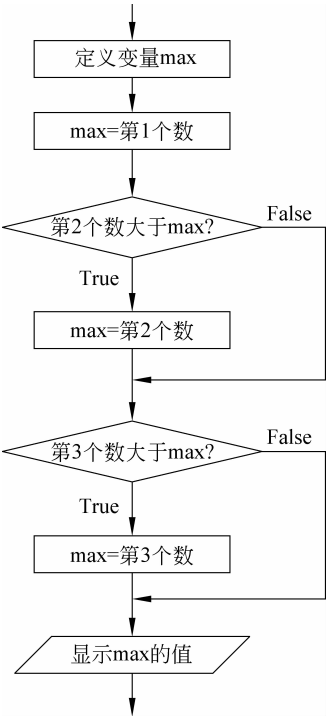


图 3-8　求解最大数的流程图

程序代码如下：

```
Private Sub optMax_Click()                          '单击"最大值"单选按钮
    Dim max As Integer                              '定义变量 max 用于存放最大值
    max=Val(txt1.Text)                              'max 中存放第 1 个数
    If Val(txt2.Text)>max Then                      'max 与第 2 个数比较
        max=Val(txt2.Text)
    End If
    If Val(txt3.Text)>max Then                      'max 与第 3 个数比较
        max=Val(txt3.Text)
    End If
    lblValue.Caption=max
End Sub

Private Sub optMin_Click()                          '单击"最小值"单选按钮
    Dim min As Integer                              '定义变量 min 用于存放最小值
    min=Val(txt1.Text)
    If Val(txt2.Text)<min Then
```

```
        min=Val(txt2.Text)
    End If
    If Val(txt3.Text)<min  Then
        min=Val(txt3.Text)
    End If
    lblValue.Caption=min
End Sub
```

程序说明：

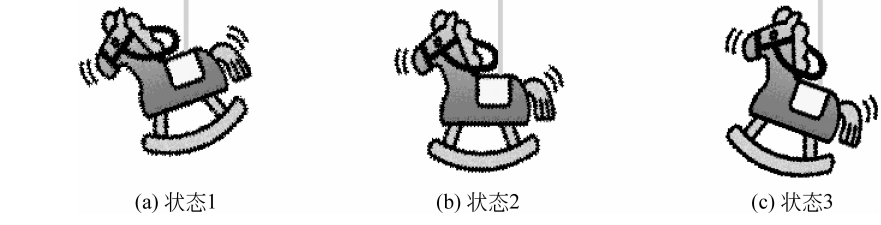
本例题中的各 If 语句均省略了 Else 分支。

【例 3.7】 旋转木马。在窗体上添加 1 个标签、1 个计时器、5 个图像框和 1 个直线控件，如图 3-9 所示。用计时器控制旋转木马在窗体内摇摆前进，程序运行时的显示效果如图 3-10 所示。

图 3-9　例 3.7 的界面设计

图 3-10　木马摇摆运动

【解】 直线控件在工具箱中的图标是 ，用于在窗体、图片框或框架中画各种直线。在窗体上添加直线后，设置其 BorderStyle 属性（直线线型）为 1-Solid（实线），BorderColor 属性（直线颜色）为红色，BorderWidth 属性（直线宽度）为 8。将 4 个小图像框的 Visible 属性设置成 False，使其不可见，仅供图片交换使用。



(a) 状态1　(b) 状态2　(c) 状态3

图 3-11　木马摇摆状态图

编程点拨：

(1) 实现木马原地摇摆。

将图 3-11 所示的(a)、(b)、(c)三幅图片按(a)，(b)，(c)，(b)的顺序轮流在大图像框中交替显示。为此，使用计时器 tmrMove 控制 5 个图像框间的图片交换，实现方法如下：

```
Private Sub tmrMove_Timer()
    img5.Picture=img1.Picture
    img1.Picture=img2.Picture
    img2.Picture=img3.Picture
    img3.Picture=img4.Picture
    img4.Picture=img5.Picture
End Sub
```

其中，在图像框 img1 至 img4 中依次放置的是对应 a，b，c，b 状态的 4 幅图片。程序运行时，需借助第 5 个图像框实现其间的图片交换。

(2) 实现木马循环移动。

木马向左前进，就是不断改变图像框 img1 的 Left 属性，使该属性值不断减小，如 img1. Left＝img1. Left－100。为了实现图像框在窗体中的循环移动，当图像框移出窗体左边界时(即图像框的右边框所在坐标小于 0)，就应该将其重新放置到窗体的右边界。具体实现方法如下：

```
Private Sub tmrMove_Timer()
    img1.Left=img1.Left-100
    If img1.Left+img1.Width<0 Then
        img1.Left=frmEx3_7.Width
    End If
End Sub
```

对于窗体中的对象，其位置坐标以窗体的左上角为基准(原点)，沿窗体水平向右为 x 坐标增加方向，沿窗体垂直向下为 y 坐标增加方向。控件的 Left 属性和 Top 属性分别标识该控件在窗体中的 x、y 坐标，而 Width 属性和 Height 属性则标识其宽度和高度，如图 3-12 所示。

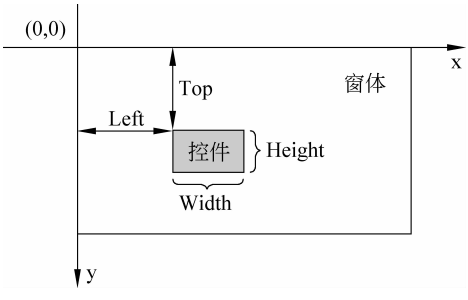


图 3-12　控件坐标表示

(3) 木马在摇摆的同时向前循环移动。

在 Timer 事件中，同时执行原地摇摆和循环前行的操作。程序代码如下：

```
Private Sub tmrMove_Timer()
    img1.Left=img1.Left - 100                    '循环向前移动
    If img1.Left+img1.Width<0 Then
```

```
        img1.Left=frmEx3_7.Width
    End If
    img5.Picture=img1.Picture                    '原地摇摆
    img1.Picture=img2.Picture
    img2.Picture=img3.Picture
    img3.Picture=img4.Picture
    img4.Picture=img5.Picture
End Sub
```

程序说明：

(1) 本例题中计时器控件的 Interval 属性为 100，即每隔 0.1 秒触发一次 Timer 事件。

(2) 对于直线控件，当线条宽度(BorderWidth 属性)大于 1 时，其 BorderStyle 属性不起作用，只能显示实心线(即 BorderStyle 属性值默认为 1-Solid)。

以上介绍了 If 语句的最常用格式，实际上 If 语句格式十分灵活，当语句组 1 和语句组 2 都是一条语句时，If 语句也可以书写成如下形式：

```
If  表达式  Then  语句 1  Else  语句 2
```

3.2.2 使用嵌套的 If 语句处理多分支问题

在 If 语句的 If 或 Else 分支中还可以再包含 If 语句，称为 If 语句的嵌套。

【例 3.8】 查看商品单价。窗体上有 2 个框架和 1 个命令按钮，其中每个框架中各含有 2 个单选按钮。程序运行时，在两个框架中分别选中一个单选按钮，单击“查看单价”按钮，以消息框形式显示被选商品的购买单价。如图 3-13 所示。

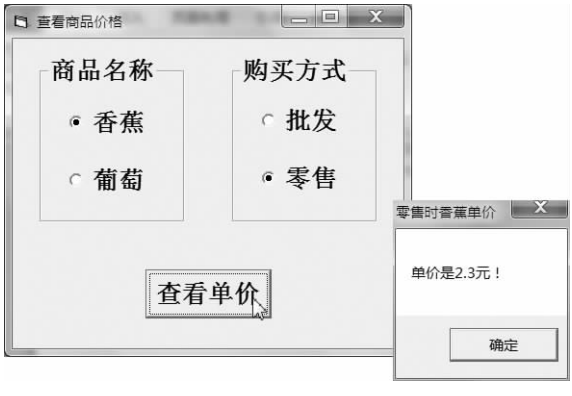


图 3-13 查看商品价格

【解】 单选按钮常成组出现，对于同一组内的单选按钮，只能有一个处于“选中”状态。通过框架控件可以对单选按钮进行分组，位于不同框架内的单选按钮属于不同的分组。本例题中使用两个框架，将四个单选按钮划分成“商品名称”和“购买方式”两组，实现

多组选择。框架控件在工具箱中的图标为  。程序代码如下：

```
Private Sub cmdDisplay_Click()
    If optBanana.Value=True Then                    '选中香蕉
        If optWhole.Value=True Then                 '选中批发
            MsgBox "单价是 1.8 元!", vbOKOnly, "批发时香蕉单价"
        Else                                        '选中零售
            MsgBox "单价是 2.3 元!", vbOKOnly, "零售时香蕉单价"
        End If
    Else                                            '选中葡萄
        If optWhole.Value=True Then
            MsgBox "单价是 1.5 元!", vbOKOnly, "批发时葡萄单价"
        Else
            MsgBox "单价是 2.0 元!", vbOKOnly, "零售时葡萄单价"
        End If
    End If
End Sub
```

程序说明：

(1) 本例题需要根据商品名称和购买方式两个条件决定商品单价。首先通过 If 语句判断所选商品名称，决定输出哪种商品的单价，用伪代码描述如下：

```
If  选中"香蕉"单选按钮  Then
    显示香蕉单价
Else
    显示葡萄单价
End If
```

但是，在执行“显示香蕉单价”的操作时，还有两种可能情况：显示香蕉的批发价格或显示香蕉的零售价格。为此，还需要通过 If 语句对“购买方式”进行判断，以决定显示香蕉的何种单价。“显示香蕉单价”的操作可用伪代码描述如下：

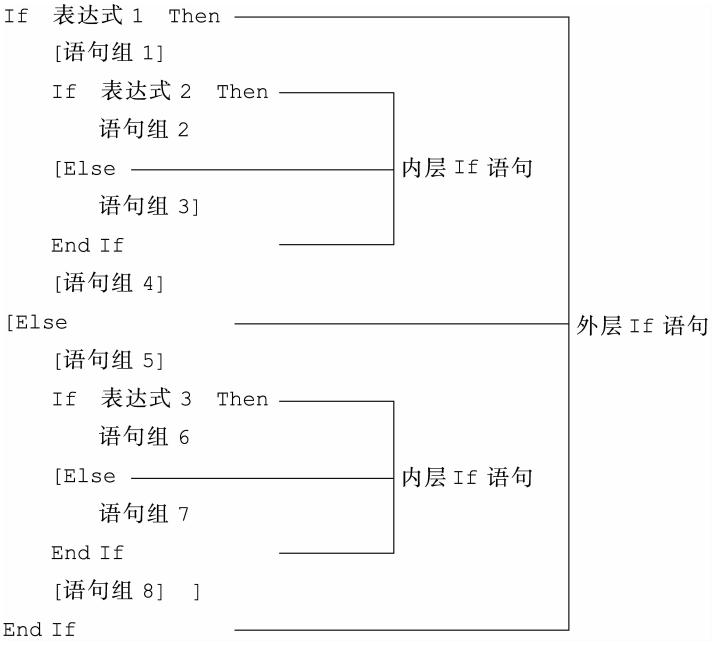
```
If  选中"批发"单选按钮 Then
    显示香蕉的批发单价
Else
    显示香蕉的零售单价
End If
```

将上述两段代码合并后的伪代码描述如下：

```
If  选中"香蕉"单选按钮 Then
    If  选中"批发"单选按钮 Then
        显示香蕉的批发单价
    Else
        显示香蕉的零售单价
    End If
```

```
Else
    显示葡萄单价
End If
```

同理,“显示葡萄单价”的操作也是通过另一个 If 语句实现。可以看到,这是一个在 If 语句的分支结构中又内含其他 If 语句的嵌套结构,称为“If 的嵌套”。嵌套 If 语句的一般形式是:



注意:只在 Then 分支或 Else 分支中含有内层 If 语句的结构也称为嵌套 If 语句。

(2) 最常见的嵌套 If 语句是在 Else 分支中含有内层 If 语句的结构,其格式可以简化为如下形式:

```
If 表达式 1 Then
    语句组 1
ElseIf 表达式 2 Then
    语句组 2
ElseIf 表达式 3 Then
    语句组 3
    ⋮
ElseIf 表达式 n Then
    语句组 n
[Else
    语句组 n+1]
End If
```

(3) 框架作为容器控件,其中可以再放置其他控件。使用框架的好处是可以将控件按类进行分组。窗体和图片框也是容器。请注意,当使用【复制】|【粘贴】的方法向框架中添加控件时,必须先选中框架,然后再执行【粘贴】操作。

3.3 使用 Select Case 语句处理多分支问题

一条 If 语句只能实现两个分支的判断操作，因而在解决多分支问题时常借助于嵌套的 If 语句。但嵌套 If 语句的格式烦琐，特别是嵌套多层后，就会大大降低程序的可读性，因而在实际应用中，更多的是使用本节将要介绍的 Select Case 语句解决多分支问题。

【例 3.9】 Select Case 语句示例。在窗体上添加 2 个标签、1 个文本框、1 个命令按钮和 1 个形状控件。程序运行时，在文本框中输入 0～5 之间的一个整数后，单击“形状”按钮，根据输入数值在形状控件上显示对应图形，同时在黄色标签中显示图形名称，如图 3-14 所示。

图 3-14 显示图形形状

【解】 形状控件在工具箱中的图标为，用于在窗体上描绘基本的图形形状。本例题中，用户在文本框中的输入情况有 7 种可能，即输入 0～5 中的任意一个值（6 种情况），以及输入错误的情况。相应地，需要判断、执行的操作也分为七种情况，因此选用 Select Case 语句实现。程序代码如下：

```
Private Sub cmdShape_Click()
    Dim a As Integer
    a=Val(txtIn.Text)
    Select Case a
        Case 0
            shpShow.Shape=0                    '设置为矩形形状
            lblShape.Caption="矩形"
        Case 1
            shpShow.Shape=1                    '设置为正方形形状
            lblShape.Caption="正方形"
        Case 2
            shpShow.Shape=2                    '设置为椭圆形形状
            lblShape.Caption="椭圆形"
        Case 3
            shpShow.Shape=3                    '设置为圆形形状
            lblShape.Caption="圆形"
        Case 4
            shpShow.Shape=4                    '设置为圆角矩形形状
            lblShape.Caption="圆角矩形"
        Case 5
            shpShow.Shape=5                    '设置为圆角正方形形状
            lblShape.Caption="圆角正方形"
        Case Else                              '对 0~5 之外的值，显示出错信息
            MsgBox "Wrong"
```

```
    End Select
End Sub
```

程序说明：

(1) 多分支结构是根据测试的条件，从不同的分支选项中选择一个满足条件的分支执行。如本例题就是根据 a 的不同取值执行不同的分支，使形状控件的 Shape 属性分别取不同的值。图 3-15 所示为 cmdShape_Click 事件的执行流程。

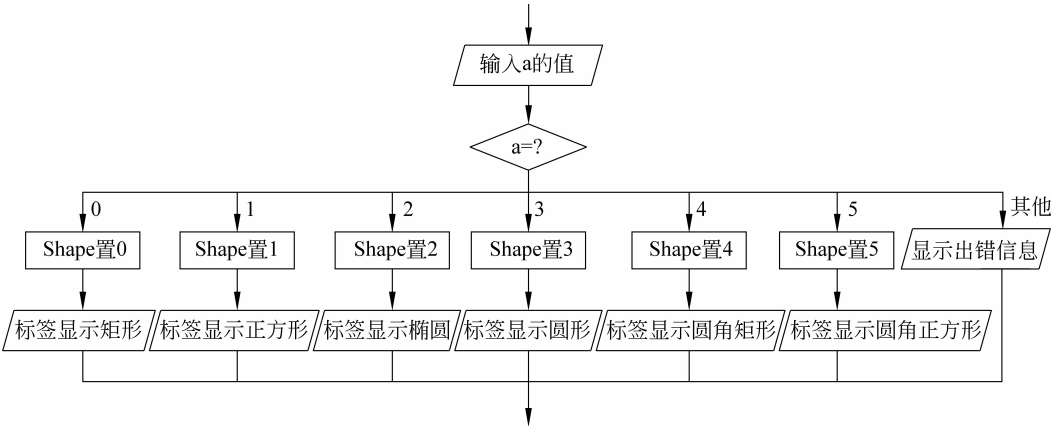


图 3-15　cmdShape_Click 流程图

(2) 形状控件的 Shape 属性决定其显示的形状形态，6 种可能的属性值 0,1,2,3,4,5 分别对应矩形、正方形、椭圆形、圆形、圆角矩形和圆角正方形。

形状控件的常用属性介绍如下：

BackColor 属性　图形的背景颜色。本例题中设置成白色。

BackStyle 属性　图形的背景样式，分为 0-Tansparent(透明)和 1-Opaque(不透明)两种。本例题中设置成 1-Opaque。

FillColor 属性　图形的填充颜色。本例题中设置为黑色。

FillStyle 属性　图形的填充样式，提供“透明”、“水平线”、“交叉线”等 8 个选项。本例中设置为 1-Transparent，表示填充区域显示成透明色。为此，尽管将 FillColor 属性设置成黑色，也仍然无法看到。

BorderColor 属性　图形的边框线颜色。本例题中设置为黑色。

BorderStyle 属性　图形的边框线样式，提供“透明”、“点线”、“点划线”等 7 个选项。本例题中选择的是 1-Solid(实心线)。

BorderWidth 属性　图形的边框线宽度。本例题中设置为 1。

(3) Select Case 语句的一般格式是

```
Select  Case  判断表达式
Case  表达式表 1
    语句组 1
Case  表达式表 2
    语句组 2
```

```
⋮
Case  表达式表 n
    语句组 n
[Case  Else
    语句组 n+1]
End  Select
```

其中，判断表达式可以是一个常量或变量，也可以是数值型表达式或字符型表达式。如 Select Case 3、Select Case x、Select Case x Mod 2 等均是合法的形式。

Case 后的“表达式表”用来判断其值是否与判断表达式相匹配，若匹配，则执行该 Case 后的语句组，然后退出 Select Case 语句；若所有 Case 后的“表达式表”均与判断表达式的值不匹配，则执行 Case Else 后的语句组。Case Else 分支可以省略，此时若判断表达式与所有 Case 后的“表达式表”均不匹配，则直接退出 Select Case 语句，不做任何操作。

Case 后的“表达式表”其形式多样，以下均是合法的形式：

```
Case  3                          判断表达式的值为 3 时匹配
Case  1 To 100                   判断表达式的值为 1~100 之间时匹配
Case  "a","A" To "Z"             判断表达式的值为"a"或"A"至"Z"之间时匹配
Case  1,3,5                      判断表达式的值为 1 或 3 或 5 时匹配
Case  Is>90                      判断表达式的值大于 90 时匹配
```

【例 3.10】 显示成绩等级。在窗体上添加 3 个标签、1 个文本框和 2 个命令按钮。程序运行时，在文本框中输入成绩并单击“判断”命令按钮，在黄色标签中显示对应的成绩等级，如图 3-16 所示。成绩在 90 ～100 为等级 A，80～89 为等级 B，70～79 为等级 C，60～69 为等级 D，0～59 为等级 E。当输入成绩大于 100 或小于 0 时，弹出出错消息框并等待用户重新输入；单击“清除”按钮，清空输入的成绩和等级。

图 3-16　显示成绩等级

【解】 编程点拨：

根据题意，若输入的成绩不在 0～100 范围内时，应显示出错信息，否则（输入正确）应进行等级分类。判断成绩是否合法只有两种可能，因此选用 If 语句，而分等级有 5 种可能，因此选用 Select Case 语句。程序代码如下：

```
Private Sub Form_Activate()
    txtScore.Text=""
    lblShow.Caption=""
    txtScore.SetFocus
End Sub

Private Sub cmdCheck_Click()
    Dim score As Integer
    score=Val(txtScore.Text)
    If score<0 Or score>100 Then
```

```
        MsgBox "数据输入错误,请重新输入!"
        Form_Activate
    Else
        Select Case score
            Case Is>=90
                lblShow.Caption="A"
            Case Is>=80
                lblShow.Caption="B"
            Case Is>=70
                lblShow.Caption="C"
            Case Is>=60
                lblShow.Caption="D"
            Case Else
                lblShow.Caption="E"
        End Select
    End If
End Sub

Private Sub cmdClear_Click()
    Form_Activate
End Sub
```

程序说明:

(1) 在窗体的 Activate 事件中将焦点设置在 txtScore 文本框上。由于设置焦点的语句不能放在 Load 事件中,因此本例改用 Activate 事件。当一个窗体成为活动窗口时触发该窗体的 Activate 事件。相反地,当一个窗体不再是活动窗口时将触发 Deactivate 事件。

(2) 使用 If 语句判断输入数据的合法性,当输入数据小于 0 或大于 100 时,提示错误信息,并清空文本框和标签,同时将光标置于文本框上。若输入数据在 0～100 之间时,使用 Select Case 语句处理 5 个等级。

(3) 在 Select Case 语句中,应特别注意各 Case 分支的排列顺序。将本例题程序代码中的 Select Case 语句修改如下,将产生错误的运行结果。

```
Select Case score
    Case Is>=60
        lblShow.Caption="D"
    Case Is>=70
        lblShow.Caption="C"
    Case Is>=80
        lblShow.Caption="B"
    Case Is>=90
        lblShow.Caption="A"
    Case Else
```

```
        lblShow.Caption="E"
End Select
```

图 3-17　错误代码运行结果

如图 3-17 所示，此时输入成绩“78”时，显示错误的等级“D”。这是因为 Select Case 语句的执行流程为自上而下依次扫描各 Case 表达式，一旦找到与判断表达式的值相匹配的 Case 分支，则执行该 Case 分支中的语句组，并在执行结束后直接跳出 Select Case 语句，即使在该 Case 分支后还存在与判断表达式相匹配的 Case 分支，也不再执行。对于成绩“78”，与第 1 个 Case 分支的条件“>=60”匹配，因而执行其后的语句，在标签中显示等级“D”，而后直接跳至 End Select，结束 Select Case 语句。

(4) 在“判断”按钮和“清除”按钮的 Click 事件中均调用了窗体的 Activate 事件过程，避免代码重复。

3.4　提 高 部 分

3.4.1　单选按钮、复选框、框架、直线和形状控件

控件是构成 VB 应用程序界面最基本的元素，掌握各控件的常用属性、事件及方法，有利于程序的设计。下面将就本章中出现的一些新控件进行综合介绍。

1. 单选按钮

单选按钮主要用于“多中选一”的情况。

(1) 属性

Alignment 属性　设置标题的显示位置。其默认值为 0，标题显示在单选按钮的右边。若 Alignment 属性值为 1，则标题显示在单选按钮的左边。

Value 属性　设置单选按钮的状态。值为 True 时按钮处于选中状态；False 时表示未选中状态。在一组单选按钮中，若一个按钮的 Value 属性为 True，则其余各单选按钮均变为 False。

Style 属性　设置单选按钮的显示方式。值为 0-Standard 时以标准方式显示按钮，即同时显示按钮和标题。值为 1-Graphical 时则以图形方式显示按钮，需进一步设置 Picture 属性为其指定图片，此时单选按钮的外观与图形化命令按钮相似。

(2) 事件

单选按钮能识别 Click、DblClick 等事件。通过这些事件，该单选按钮的 Value 属性值将变为 True，而同组内其他按钮的 Value 属性值自动变为 False。

2. 复选框

复选框主要用于在多个选项中选择其中一项或几项的情况。