

# 第 3 章

## 分支结构程序设计

### 本章将介绍的内容

#### 1. 基础部分

- 关系、逻辑运算符及其表达式。
- If 语句与 Select Case 语句;If 语句的嵌套。
- 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。

#### 2. 提高部分

- 再论常用控件：单选按钮、复选框、框架、直线和形状。
- 综合举例。

### 各例题知识要点

例 3.1 使用关系运算符及表达式。

例 3.2 使用逻辑运算符及表达式。

例 3.3 使用 If 语句设计分支结构程序,使用 If 语句格式及流程图。

例 3.4 使用无 Else 分支的 If 语句,计算分段函数值。

例 3.5 使用复选框和多行文本框;设置与文本字体格式相关的属性和颜色常量。

例 3.6 使用单选按钮,掌握求三个数中的最大(小)值算法。

例 3.7 用分支结构完善校验密码程序。

例 3.8 使用直线控件,掌握控件坐标表示、图像的循环变换和移动,使用窗体级变量。

例 3.9 使用框架、嵌套的 If 语句。

例 3.10 使用形状控件,用 Select Case 语句设计分支结构;使用语句格式和流程图。

例 3.11 合理选用 If 语句和 Select Case 语句,使用 Activate 事件。

(以下为提高部分例题)

综合实例 3 使用复选框、单选按钮、框架、If 语句、Select Case 语句。

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,本节将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 6 种关系运算符(见表 3-1)。

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,请读者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) 10-8 <= 2
- (2) True <> 1
- (3) 3 > x >= 0 (假设 x 为 Integer 类型变量,且值为 2)

- (4) "abc"="abc"
- (5) "abc"<"aBcd"
- (6) "abcd">"ab"

【解】 本例是关系表达式的运算，各表达式的值是(1) True, (2) True, (3) False, (4) True, (5) False, (6) True。

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    |
|----------------------------------------|---------------------|--------|
| <div>高</div> <div>↓</div> <div>低</div> | ()                  | 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                 |        |
|                                        | Or                  |        |

(2) 在表达式 10-8<=2 中，按照优先级应先计算 10-8，结果为 2，再计算 2<=2。对于数值型数据，按其数值大小进行比较，“2 小于等于 2”成立，所以表达式的值为 True。

(3) 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。

(4) 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。

(5) 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，若两字符串长度相等且所有字符对应相等，则两字符串相等。例如，字符串 abc 与 abc 就是完全相等的两个字符串，所以表达式 "abc"="abc" 的值为 True。

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,在第二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。对于字符串 abcd 和 ab,当比较到第三个字符时,串 ab 中已经为空,而空字符对应的 ASCII 码为 0,小于字符 c 的 ASCII 码值,所以 c 所在字符串 abcd 大于 ab,即“"abcd">"ab"”的值为 True。

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 提供以下逻辑运算符,如表 3-3 所示。

表 3-3 逻辑运算符

| 运算符 | 含义  | 举 例               | 例 子 作 用                         |
|-----|-----|-------------------|---------------------------------|
| Not | 逻辑非 | Not(0<5)          | 对 0<5 的结果取反                     |
| And | 逻辑与 | (10-8)<=2 And 5>4 | 对 (10-8)<=2 的结果和 5>4 的结果进行“与”运算 |
| Or  | 逻辑或 | 5>0 Or False=0    | 对 5>0 的结果和 False=0 的结果进行“或”运算   |

VB 6.0 还提供其他三个逻辑运算符: Xor、Eqv 和 Imp,由于篇幅有限本书不做介绍。

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”(And)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”(Or)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”(Not)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)  $3 > x \text{ And } x \geq 0$  (假设  $x$  为 Integer 类型变量,且值为 2)

(2)  $\text{True} <> 1 \text{ And False}$

(3)  $3 > 5 \text{ Or False}$

(4)  $\text{Not}(\text{True} = 0)$

**【解】** 本例是逻辑表达式的运算,各表达式的值是(1)True,(2)False,(3)False,(4)True。

说明:

(1)  $3 > x \text{ And } x \geq 0$  为逻辑表达式,其运算顺序是:先计算关系表达式  $3 > x$  和  $x \geq 0$  的值,结果均为 True;再对表达式  $\text{True And True}$  进行逻辑与运算,其结果为 True,因此表达式  $3 > x \text{ And } x \geq 0$  的值为 True。

注意,表达式  $3 > x \text{ And } x \geq 0$  与  $3 > x \geq 0$  的意义不同。 $3 > x \geq 0$  仅为一个复杂的关系表达式。在 VB 中若要表示“ $x$  的取值区间为  $[0, 3]$ ”这一数学含义,正确的表达式是  $0 \leq x \text{ And } x \leq 3$ ,而不是  $0 < x \leq 3$ ,无论  $x$  取何值,  $0 \leq x \leq 3$  的值均为 True。

(2) 在表达式  $\text{True} <> 1 \text{ And False}$  中,按优先级先计算  $\text{True} <> 1$ ,值为 True,而  $\text{True And False}$  的值为 False。逻辑与运算的规律是:只要有一个操作数是 False 或 0,结果就是 False,只有当两操作数均为 True 时,结果才为 True。

(3) 逻辑或运算的规律是:只要有一个操作数是 True 或非 0,结果就是 True,只有当两操作数均为 False 时,结果才为 False。在表达式  $3 > 5 \text{ Or False}$  中,由于逻辑或运算的第 2 个操作数已经是 False,所以  $3 > 5$  的值就至关重要了,直接决定着表达式的值。因为  $3 > 5$  的值是 False,所以  $3 > 5 \text{ Or False}$  的值就是 False。

(4) 当逻辑型数据与数值型数据进行比较时,True 对应数值 -1,False 对应数值 0。因而表达式  $\text{True} = 0$  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 If 语句的使用

**【例 3.3】** If 语句示例。在窗体上添加一个文本框、3 个标签和一个命令按钮。程序运行时,在文本框中输入一个数,单击命令按钮,在右侧标签中显示奇、偶性的判断结果,如图 3-1 所示。

**【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
```



图 3-1 判断奇、偶性

**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 的值为 0, 即 a 整除 2 后的余数为 0, 则说明 a 为偶数, 在标签中显示“偶数”, 否则显示“奇数”。该执行过程的流程图如图 3-2 所示。

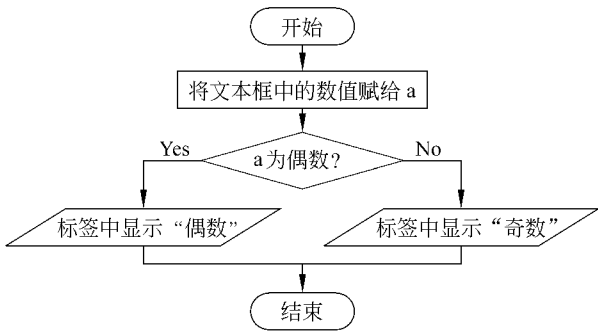


图 3-2 cmdJudge\_Click 事件的执行过程

(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 后的语句 lblValue. Caption = "偶数"; 当其值为“假”时, 执行 Else 后的语句 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 ]
```

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

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

在窗体上添加一个文本框和三个标签。程序运行时,在文本框中输入 x 值时,立刻计算出相应的函数值 y,并显示在下面的标签中,如图 3-4 所示。

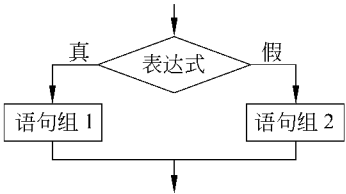


图 3-3 If 语句的执行过程

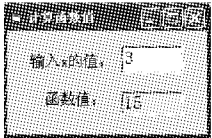


图 3-4 计算函数值

**【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 3 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。该过程的流程图如图 3-5 所示。

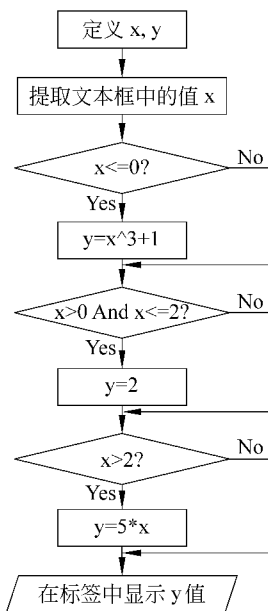


图 3-5 txtInput\_Change 事件的执行过程

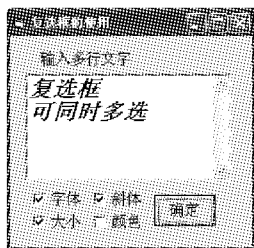


图 3-6 复选框的使用

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。

**【例 3.5】** 在窗体上添加一个标签、一个文本框、4 个复选框和一个命令按钮。程序运行时,在文本框中输入数据并根据需要选择不同复选框。单击“确定”按钮,根据复选框选中情况设置文本框中的文字字体、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
    '若选中"字体"复选框,则文本框中的字体设置为宋体,否则为黑体
    If chkFnt.Value=1 Then
        txtInput.FontName="宋体"
    Else
        txtInput.FontName="黑体"
    End If
    '若选中"斜体"复选框,则文本框中的字型设置为斜体,否则为正常字型
    If chkItc.Value=1 Then
        txtInput.FontItalic=True
    Else
        txtInput.FontItalic=False
    End If
    '若选中"大小"复选框,则文本框中的字号大小设置为 18 点,否则为 24 点
    If chkSz.Value=1 Then
        txtInput.FontSize=18
    
```

```

Else
    txtInput.FontSize=24
End If
'若选中"颜色"复选框,则文本框中的文本颜色设置为红色,否则为黑色
If chkCol.Value=1 Then
    txtInput.ForeColor=vbRed
Else
    txtInput.ForeColor=vbBlack
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 表示没有选中;1 表示选中;2 表示选中但不可用(复选框呈灰色)。本例中,在设计阶段将 4 个复选框的 Value 属性均设置成 0,使它们处于“未选中”状态。

(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) 本例题中使用了 4 个复选框,并通过 4 个 If 语句分别处理各复选框的选择情况。以第一个 If 语句为例,当 chkFont.Value 的值为 1(即选中“字体”复选框)时,执行 Then 后的语句,将文本框字体设置为“宋体”,否则执行 Else 后的语句,将文本框字体设置为“黑体”。

(4) 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、字体是否斜体、字体是否加粗以及字体的大小。其中 FontItalic 和 FontBold 的属性值为 True 或 False,值为 True 时表示倾斜或加粗。文本框还有 FontStrikethru 和 FontUnderline 属性,分别设置删除线和下划线。

(5) 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。默认值为 False。

(6) 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。

(7) vbRed 和 vbBlack 是 VB 提供的符号常量,用于表示不同的颜色。各颜色常量所对应的颜色参见表 2-31。

**【例 3.6】** 求最大数与最小数。在窗体上添加三个文本框、两个标签和两个单选按钮。程序运行时,输入 3 个整数,单击“最大数”单选按钮,则在下面的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。

**【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 属性决定:True 为选中,False 为未选中。

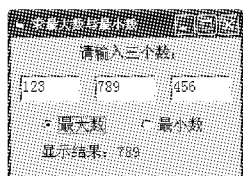


图 3-7 求最大数与最小数

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) 在开始比较前，先假设第一个数就是三个数中的最大值，将其存放到变量 max 中。

(3) 将当前找到的最大值 max(即第一个数)与第二个数进行比较。如果第二个数比 max 大，则将第二个数作为最大值放入 max 中，否则就什么也不做。此时变量 max 中存放的是前两个数中的较大值。

(4) 同理，再将变量 max 与第三个数进行比较。如果第三个数比 max 大，则将第三个数作为最大值放入 max 中，否则什么也不做。此时变量 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。程序代码如下：

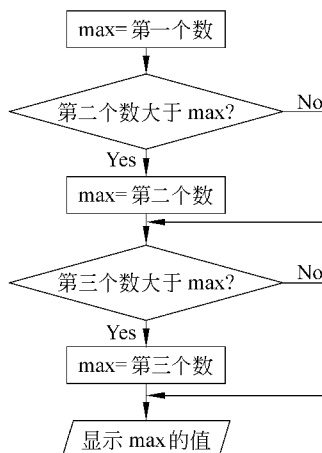


图 3-8 求解最大数的流程图

```
Private Sub optMax_Click()
    Dim max As Integer
    max=Val(txt1.Text)
    If Val(txt2.Text)>max Then
        max=Val(txt2.Text)
    End If
    If Val(txt3.Text)>max Then
        max=Val(txt3.Text)
    End If
    lblValue.Caption="显示结果：" & max
End Sub

Private Sub optMin_Click()
    Dim min As Integer
    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="显示结果：" & 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】** 编写校验密码程序，完善例 2.11。在例 2.11 的基础上进行修改，要求程序运行时，在文本框中输入密码并单击“校验密码”按钮后，若输入的密码与事先规定的密