

# 第7周 控件

## 7.1 实验目的

- 控件的一般属性；
- CommandButton, Label, TextBox；
- Frame, CheckBox, OptionButton；
- HScrollBar, VscrollBar；
- ListBox, ComboBox；
- PictureBox, Image。

## 7.2 示例学习

例 7-1 使用标签

知识点：

- 使用标签的各种属性；
- 使用鼠标事件；
- 使用字体及光标；
- 使用 shell "程序名 参数"表示调用外部程序。

程序：Label\_mouse\_shell\_263.frm

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    Label1.FontUnderline = False
    Label1.FontBold = False
    Label1.ForeColor = vbBlack
    Label1.MousePointer = vbDefault
End Sub

Private Sub Label1_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    Label1.FontUnderline = True
    Label1.FontBold = True
    Label1.ForeColor = vbBlue
    Label1.MousePointer = 14
End Sub

Private Sub Label1_Click()
    Shell "C:\Program Files\Internet Explorer\IEXPLORE.EXE " & _
        "http://www.263.net", vbNormalFocus

    'Shell "可执行程序 参数",vbNormalFocus

    'Me.MousePointer = vbHourglass
End Sub
```

程序说明：本程序中需要加入一个标签并使标签表现为超链接的样子（当鼠标移上时改变其颜色及字体）。运行结果如图 7-1。当鼠标点击时，调用 Shell 语句执行一个外部命令，Shell 语句后跟的外部命令是由外部命令的程序名以及所带的参数构成的。Shell 语句中的 vbNormalFocus 表示打开外部命令并使之成为当前活动窗口。



图 7-1 表现为超链接

## 例 7-2 使用文本框

知识点：

SetFocus 方法,  
GotFocus, LostFocus 事件  
SelStart, SelLength, SelText, Text 属性  
MultiLine, ScrollBars 属性  
PasswordChar 属性（如 "\*"）  
Locked, Enabled 属性

程序：text\_sel\_password\_enter

```
Private Sub Command1_Click()  
    MsgBox "hello"  
End Sub  
  
Private Sub Form_Load() '一开始就让文本框获得输入焦点  
    Me.Show  
    Text1.SetFocus  
End Sub  
  
Private Sub Text1_GotFocus() '获得输入焦点时就选定全部  
    Text1.SelStart = 0  
    Text1.SelLength = Len(Text1.Text)  
End Sub  
  
Private Sub Text2_KeyPress(KeyAscii As Integer) '处理回车  
    If KeyAscii = 13 Then '13 表示回车的 Ascii 码  
        Command1.SetFocus  
        Command1_Click '调用其他控件的命令  
    End If  
End Sub
```

程序说明：本程序中有两个文本框及一个命令按钮（如图 7-2 所示）。一个文本框设置

MultiLine 为 True，ScrollBars 为 3（水平、垂直方向都有滚动条）。另一个文本框，设置 PasswordChar 为\*，使得它表现为口令框的格式。程序中一开始就让文本框获得输入焦点（SetFocus），而一旦获得输入焦点（GotFocus 事件），就选定全部文本（使用 SelText 及 SelLength 属性）。而当在文本框获入回车（通过 KeyPress 事件中的按键的 ASCII 码判断出）时，则调用其他控件的命令（这里是调用 Command\_Click，也就是说在第二个文本框中输入回车时，相当于单击了 Command1）。

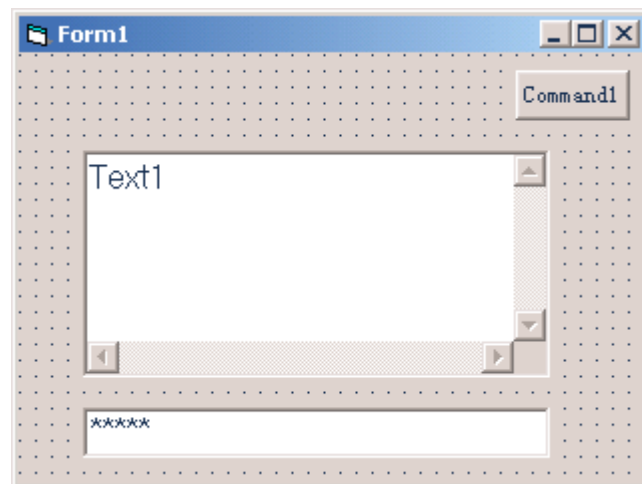


图 7-2 使用文本框

### 例 7-3 使用成组框、单选框、复选框

知识点：

使用成组框、单选框、复选框；

复选框 CheckBox 的 Value(1 或 0 或 2):vbChecked;

单选框 OptionButton 的 Value(True 或 False)。

程序：Checkbox\_option.frm

```
Private Sub Command1_Click()
    sHobby = ""
    If Check1.Value = vbChecked Then
        sHobby = sHobby & Check1.Caption
    End If
    If Check2.Value = vbChecked Then
        sHobby = sHobby & Check2.Caption
    End If
    If Check3.Value = vbChecked Then
        sHobby = sHobby & Check3.Caption
    End If
    Print "爱好有:" & sHobby
End Sub

Private Sub Command2_Click()
    If Option1.Value = True Then
        Print "男"
    ElseIf Option2.Value = True Then
        Print "女"
    ElseIf Option3.Value = True Then
```

```

        Print "性别不知道"
    End If
End Sub

Private Sub Command3_Click()
    For i = 0 To 4
        If Opt(i).Value Then Print i
    Next
End Sub

Private Sub Opt_Click(Index As Integer)
    Me.Frame3.BackColor = QBColor(Index)
End Sub

```

程序说明：程序中使用复选框来表示兴趣爱好，用单选框表示性别，另外来用一组同名的单选框来表示颜色，如图 7-3 所示。同名的一组控件又称为控件数组，有关控件数组的概念可以参见教材中的 7.6 节。



图 7-3 使用成组框、单选框、复选框

#### 例 7-4 使用形状、直线及计时器

知识点：

Shape 形状 / Line 线及其属性：

Shape 的位置 以及 Line 的 x1,y1,x2,y2

BorderStyle, BorderColor

BorderWidth

FillStyle, FillColor,

BackStyle, BackColor

Timer 定时器及其属性：

Interval, Enabled

程序：Timer\_Shape\_Line\_Clock.frm

```

Private Sub Form_Load()
    Timer1.Interval = 1000
    Line1.X1 = Shape1.Left + Shape1.Width / 2
    Line1.Y1 = Shape1.Top + Shape1.Height / 2
End Sub

```

```

Private Sub Timer1_Timer()

    t = Second(Now)
    theta = (t * 6 - 90) / 180 * 3.14159

    Line1.X2 = Line1.X1 + 1000 * Cos(theta)
    Line1.Y2 = Line1.Y1 + 1000 * Sin(theta)

    Me.Caption = Now
End Sub

```

程序说明：本程序用定时器、直线与形状形成一个简单的时钟，如图 7-4 所示。程序中在定时器对象的 Timer 事件中，计算了秒针的角度和位置，其中 60 秒代表 360 度，所以要将秒数 (t) 乘以 6，另外减去 90 是因为坐标系不同。读者可以将此程序改进，加上时针和分针。

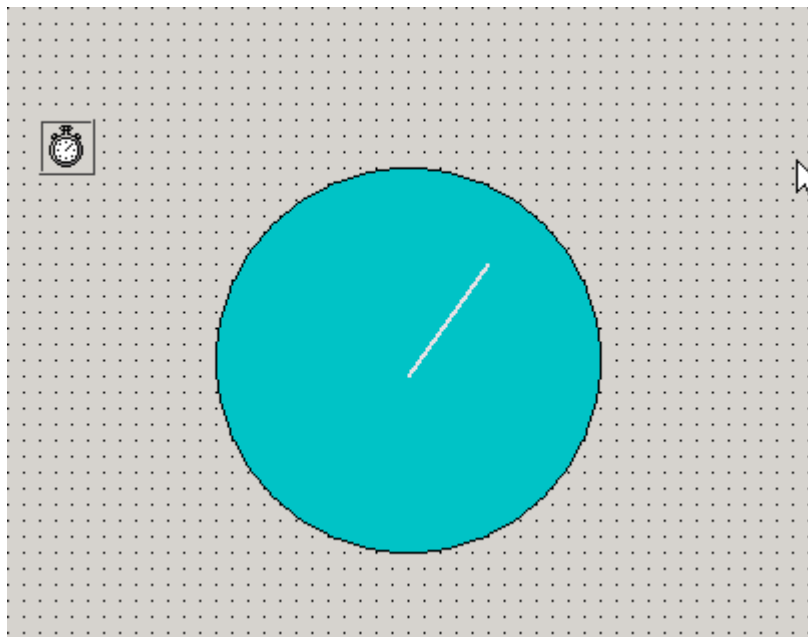


图 7-4 一个简单的秒表

#### 例 7-5 使用列表框及组合框

知识点：

ListBox (列表框) \ ComboList (组合框) 的属性和方法：

Style 属性

MultiSelect 属性

List(i), ListCount, ListIndex, Selected(i), SelCount 属性

AddItem, RemoveItem, Clear 方法

Combo 的 Text 属性

程序：Combox\_list.frm

```

Private Sub Form_Load()
    For i = 1 To 5
        Combo1.AddItem ("xxxxx" & Int(Rnd * 100))
    Next i
End Sub

```

```

Next
Combo1.Text = ""
End Sub
Private Sub Command3_Click()
    If Combo1.Text <> "" Then
        List1.AddItem (Combo1.Text)
    End If
End Sub
Private Sub Command1_Click()
    For i = 0 To List1.ListCount - 1
        If List1.Selected(i) Then
            s = List1.List(i)
            List2.AddItem (s)
        End If
    Next
    For i = List1.ListCount - 1 To 0 Step -1
        If List1.Selected(i) Then
            List1.RemoveItem (i)
        End If
    Next
End Sub

Private Sub Command2_Click()
    If List2.SelCount > 0 Then
        List1.AddItem (List2.List(List2.ListIndex))
        List2.RemoveItem (List2.ListIndex)
    End If
End Sub

```

程序说明：程序中需要加入两个列表框、一个组合框及三个按钮。程序在 Form\_Load 时向 Combo1 中加入了多条内容。当按钮 3 单击时将 Combo1 中的文本加入到 List1 中。当按钮 1 单击时，将 List1 中的所选内容移入到 List2 中，由于 List1 是允许多选的（MultiSelect 属性已置为 True），所以用了循环。当按钮 2 单击时，将 List2 中所选内容移入到 List1 中。程序中用到了列表框及组合框的多种属性和方法。运行情况如图 7-5 所示。

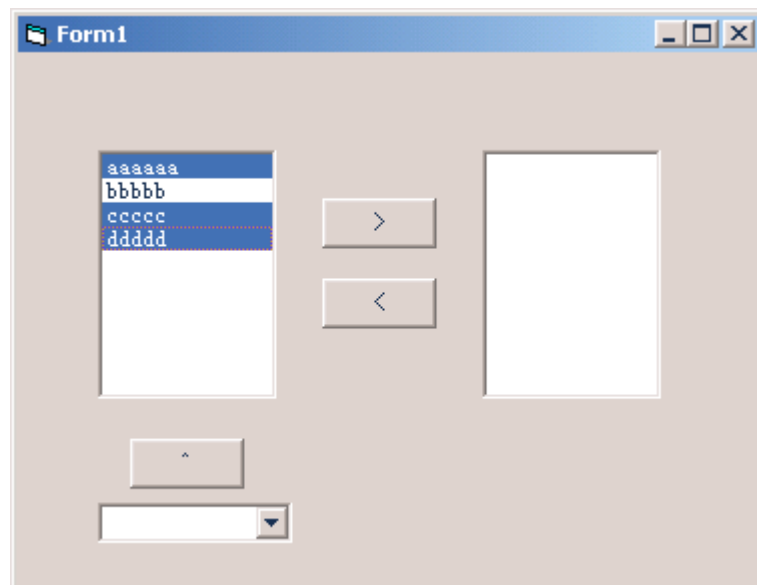


图 7-5 使用列表框及组合框

### 例 7-6 使用图片框及图像

知识点：

使用图片框及图像；  
图片框及图像的区别；  
使用 LoadPicture 载入图像。

程序：dance\_girl（跳舞的小女孩）

```
Public PicNo As Integer

Private Sub Form_Load()
    PicNo = 0
    Timer1.Interval = 300
End Sub

Private Sub Timer1_Timer()
    PicNo = PicNo + 1
    Picture1.Picture = LoadPicture( _
        App.Path & "\" & "dg0" & PicNo & ".bmp")
    If PicNo = 5 Then
        PicNo = 0
    End If
End Sub
```

程序说明：本程序中需要加入一个图片框、一个定时器控件、两个按钮，在程序所在目录需要有五个图片文件，文件名分别为 dg01.bmp、dg02.bmp、dg03.bmp、dg04.bmp 及 dg05.bmp、本程序中通过定时器对象的 Timer 事件来切换图片框的图像从而显示出动画效果（小女孩在跳舞），如图 7-6 所示。在切换图形时，为了每次取得不同的一个图片文件名这里使用了一个变量（PicNo），每次时钟事件发生时它就增加 1，如果超过了第 5 张图片，则又回到首张图片。注意变量 PicNo 不能是在子过程中定义的局部变量（可以是全局变，窗体级变量，也可以是静态变量），因为局部变量每次都重新产生，达不到在上一次变量的基础上加 1 的目的。另外，程序中用 App.Path 表示应用程序所在路径。



图 7-6 使用图片框及图像

### 例 7-7 使用进度条

知识点：

加入 ActiveX 控件；  
设置 ActiveX 控件的属性；  
编程使用 ActiveX 控件的属性、方法及事件。

程序：progress1.vbp

```
Private Sub Timer1_Timer()  
    If Me.ProgressBar1.Value + 5 <= Me.ProgressBar1.Max Then  
        Me.ProgressBar1.Value = Me.ProgressBar1.Value + 5  
    Else  
        Timer1.Enabled = False  
    End If  
End Sub
```

程序说明：进度条控件是 Microsoft Window Common Control 中的一种控件，按照上文提到的方法把该控件加入到工具箱中。然后单击进度条控件（Progress Bar），并加入到窗体中，另外再加入一个命令按钮，如图 7-7 所示。注意程序中对控件的属性的使用。

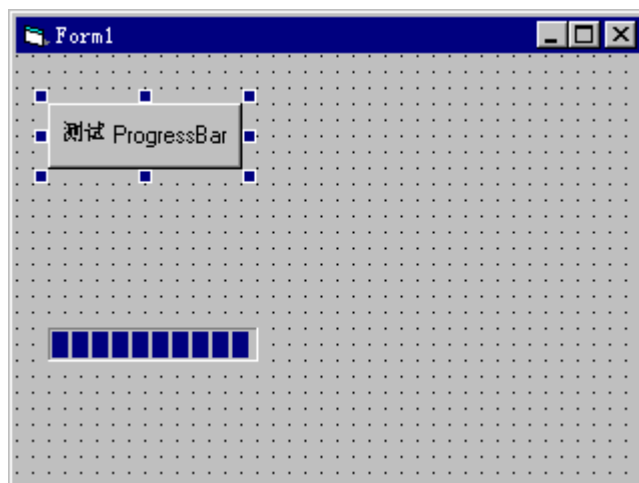


图 7-7 使用进度条控件

## 7.3 实验练习

### 练习 7-1 时钟

练习目的：

练习使用计时器控件、直线控件、形状控件；  
练习在设计时对属性的设置；  
练习在通过程序在运行时对属性的设置。

练习内容：

试利用计时器控件、直线控件、形状控件编写一个时钟，其上有一个圆盘及三个指针，指针会正确显示时间。

参考实现：时钟.frm

```
Dim cx As Integer, cy As Integer  
Private Sub Form_Load()  
    cx = Shape1.Left + Shape1.Width / 2
```



```

        cy = Shape1.Top + Shape1.Height / 2
        Timer1.Interval = 1000
        Timer1_Timer
    End Sub

```

```

Private Sub Timer1_Timer()
    Dim t As Double, theta As Double
    Const Pi As Double = 3.14159

    t = Second(Now)
    theta = (t * 6 - 90) / 180 * Pi
    Line1.X2 = cx + 1300 * Cos(theta)
    Line1.Y2 = cy + 1300 * Sin(theta)
    Line1.X1 = cx - 200 * Cos(theta)
    Line1.Y1 = cy - 200 * Sin(theta)

    t = Minute(Now)
    theta = (t * 6 - 90) / 180 * Pi
    Line2.X2 = cx + 1000 * Cos(theta)
    Line2.Y2 = cy + 1000 * Sin(theta)
    Line2.X1 = cx - 100 * Cos(theta)
    Line2.Y1 = cy - 100 * Sin(theta)

    t = Hour(Now)
    theta = (t * 30 - 90) / 180 * Pi
    Line3.X2 = cx + 800 * Cos(theta)
    Line3.Y2 = cy + 800 * Sin(theta)
    Line3.X1 = cx - 100 * Cos(theta)
    Line3.Y1 = cy - 100 * Sin(theta)

    Me.Caption = Now
End Sub

```

## 练习 7-2 电子滚动屏幕

练习目的：

练习使用标签控件及定时器控件

练习内容：

设计一个电子滚动屏幕，使“电子滚动屏幕”这几个汉字自上向下反复移动。

参考实现：电子滚动屏幕.frm

```

Private Sub Form_Load()
    Label1.Caption = "电子滚动屏幕"
    Label1.FontSize = 24
    Timer1.Interval = 200
End Sub

Private Sub Timer1_Timer()
    Label1.Top = Label1.Top + 100
    If Label1.Top > Me.Height - Label1.Height Then
        Label1.Top = 0
    End If
End Sub

```