

# 第 4 章 Visual Basic 常用内部控件

## 本章要点

- Visual Basic 中常用标准控件的画法
- Visual Basic 中常用标准控件的属性
- Visual Basic 中常用标准控件的事件和方法

前面已经学习了 Visual Basic 中的各种程序控制结构,可以设计出各种流程的程序。但是要想设计出一个完美的程序,单纯地依靠设计程序代码是不够的,还需要一些美观实用的交互界面,这就用到下面我们要介绍的各种控件。在这一章中,我们将学习命令按钮、标签、文本框、图片框、图像框、复选框、单选按钮、列表框、组合框、滚动条、计时器和框架,帮助我们实现完善的程序设计。

## 课前体验 成绩录入

前面我们已经学习了 Visual Basic 的基础知识和各种控制结构的程序设计,可以完成很多应用程序了,但是读者是否感觉到设计的界面不够漂亮,用起来不太方便呢?其实,Visual Basic 是一个面向对象的开发工具,它提供了许多内部控件,可以方便地进行用户界面设计。下面我们先练习使用 Visual Basic 内部控件来开发一个学生成绩输入与合计的界面。

要求输入各门课的成绩,如果输入的成绩在 0~100 之间,则可以继续输入,否则清除原数据,并且要求重新输入。单击“合计”按钮,计算总成绩并显示在文本框中。

设计步骤如下:

### 1. 建立用户界面

- (1)在窗体上建立 6 个标签。
- (2)在窗体上建立 6 个文本框,分别用来显示数学、物理、化学、语文、外语和合计成绩。
- (3)在窗体上建立 5 个复选框,用来设置参加合计的课程成绩。
- (4)在窗体上建立 1 个命令按钮。
- (5)参考图 4-1,将各个控件排列整齐。

### 2. 设置控件属性

按图 4-1 设置各个控件的属性如下:

- 6 个标签标题分别为:数学、物理、化学、语文、外语、合计。
- 6 个文本框的 Text 属性为空。
- 5 个复选框的标题分别为:数学、物理、化学、语文、外语。
- 命令按钮标题为:合计。
- 窗体标题为:成绩录入。

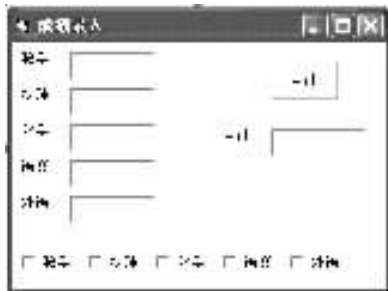


图 4-1 成绩录入

### 3. 编写事件代码

(1) 假设用来输入数学成绩的文本框名称为 Text1, 该文本框的 LostFocus 事件过程如下:

```
Private Sub Text1_LostFocus()
    If Val(Text1.Text) < 0 Or Val(Text1.Text) > 100 Then
        Text1.Text = ""
        Text1.SetFocus
    End If
End Sub
```

(2) 其他文本框的 LostFocus 事件类似。

(3) 假设按钮名称为 Command1, 该按钮的 Click 事件过程如下:

```
Private Sub Command1_Click()
    If Check1.Value = 1 Then Sum = Sum + Val(Text1.Text)
    If Check2.Value = 1 Then Sum = Sum + Val(Text2.Text)
    If Check3.Value = 1 Then Sum = Sum + Val(Text3.Text)
    If Check4.Value = 1 Then Sum = Sum + Val(Text4.Text)
    If Check5.Value = 1 Then Sum = Sum + Val(Text5.Text)
    Text6.Text = Sum
End Sub
```

说明: 如果复选框被选中, 则计算总成绩时要计算这门课程的成绩。

### 4. 运行程序

输入成绩, 选择需要计算总成绩的课程, 单击“合计”按钮, 计算总成绩。程序运行结果如图 4-2 所示。

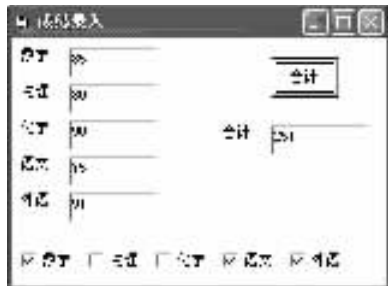


图 4-2 成绩录入运行结果

从上面的实例中我们可以看出,使用 Visual Basic 的内部控件建立用户界面是一件非常容易的事,我们只要轻点鼠标,一个漂亮的用户界面就马上建立起来了。本章为大家介绍的就是 Visual Basic 中的常用控件及其使用。

## 4.1 控件概述

控件是构成用户界面的基本元素,只有掌握了控件的属性、事件和方法,才能编写界面美观、具有实用价值的应用程序。

Visual Basic 中的控件分为两类,一类是标准控件(或称内部控件),另一类是 ActiveX 控件。启动 Visual Basic 后,工具箱中只有标准控件,共有 20 个。本章将介绍一些常用的标准控件的用法,包括标签、文本框、图片框、图像框、命令按钮、复选框、单选按钮、列表框、组合框、水平滚动条、垂直滚动条、计时器、框架。

## 4.2 命令按钮

### 4.2.1 命令按钮的常用属性

Visual Basic 中的按钮控件是命令按钮,它是 Visual Basic 应用程序中最常用的控件,它提供了用户与程序交互最简便的方法。其默认名称和标题为 Commandx(其中 x 为 1,2,3...)。

在应用程序中,命令按钮常用来在单击时执行指定的操作。其大多数属性与窗体相同,下面我们将介绍命令按钮的常用属性。

#### 1. Caption(标题)属性

标题属性用来设置按钮的标题,即在按钮上显示的文本,可以在“属性”窗口中设置其值,如图 4-3 所示。课前体验中我们用到的命令按钮的标题“合计”就是在“属性”窗口中设置的。



图 4-3 标题属性

按钮的标题属性也可以通过程序代码设置,也就是通过语句设置属性值。例如,设置按钮的标题属性为“确定”,可以在“代码”窗口编写如图 4-4 所示的程序。程序运行后,单击命

令按钮,按钮显示结果如图 4-5 所示。

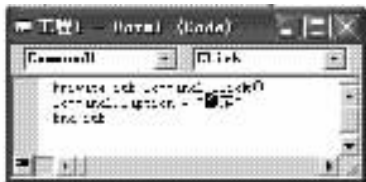


图 4-4 通过程序代码设置属性



图 4-5 标题属性设置效果

## 2. Enabled(允许)属性

Enabled 属性用来确定按钮是否能够对用户产生的事件作出反应,值为 Boolean 类型。我们在 Windows 中经常会看见有的按钮是灰色的,不能使用,就是将该按钮的 Enabled 属性设为 False。它同样可以通过“属性”窗口或程序代码设置,如 `Command1.Enabled = True`。当该值为 True 时,允许按钮对事件作出反应;如果为 False,则禁止对事件作出反应。

提示:在应用程序中,有时为防止误操作,就可以利用 Enabled 属性让命令按钮暂时失去作用,在合适的时候再将其属性设为 True,恢复按钮的功能。

## 3. FontBold(加粗)

该属性的值为 Boolean 类型,用来确定按钮标题文本是否加粗,可以通过“属性”窗口的 Font 属性设置或通过程序代码设置。例如,`Command1.FontBold = True`,则按钮文本加粗显示。

双击命令按钮的 Font 属性,会弹出如图 4-6 所示的对话框,这和我们在其他应用程序中设置字体加粗是一样的。

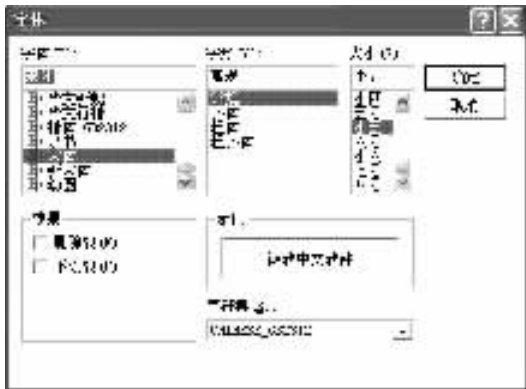


图 4-6 字体设置

## 4. FontItalic(倾斜)

该属性用来设置按钮标题文本是否倾斜,可以通过“属性”窗口的 Font 属性设置或通过程序代码设置,该属性的值为 Boolean 类型。

## 5. FontName(字体)

该属性用来设置按钮标题文本的字体类型,可以通过“属性”窗口的 Font 属性设置或通过程序代码设置,例如 `Command1.FontName = “黑体”`。

## 6. FontSize( 字号)

该属性用来设置按钮标题文本的大小即字号,可以通过“属性”窗口的 Font 属性设置或通过程序代码设置,例如 `Command1.FontSize=40`。

## 7. FontUnderline( 下划线)

该属性用来设置按钮标题文本是否加下划线,可以通过“属性”窗口的 Font 属性设置或通过程序代码设置,该属性的值为 Boolean 类型。

试一试:在工具箱中,单击命令按钮,在窗体中画一个命令按钮,然后在“属性”窗口设置该按钮前面讲到的 7 个属性,看看按钮有什么变化?

## 8. Height( 高度)

该属性用来确定按钮的高度,可以通过“属性”窗口或程序代码设置。

## 9. Left( 左边位置)

该属性用来设置按钮左边的坐标值,可以通过“属性”窗口或程序代码设置。

## 10. Top( 顶边位置)

该属性用来设置按钮上边的坐标值,可以通过“属性”窗口或程序代码设置。

## 11. Visible( 可见性)

该属性用来设置按钮是否可见,该属性的值为 Boolean 类型。

想一想:Enabled 属性和 Visible 属性有什么不同(这两个属性很容易混淆)?

## 12. Width( 宽度)

该属性用来设置按钮的宽度,可以通过“属性”窗口或程序代码设置。

## 13. Cancel

当一个命令按钮的 Cancel 属性被设置为 True 时,按【Esc】键与单击该命令按钮的作用相同。

注意:在一个窗体中,只能有一个命令按钮的 Cancel 属性被设置为 True。

## 14. Default

当一个命令按钮的 Default 属性被设置为 True 时,按回车键和单击该命令按钮的效果相同。

注意:在一个窗体中,只能有一个命令按钮的 Default 属性被设置为 True。

## 15. Style

该属性用来指定按钮的显示类型和操作。该属性在运行期间是只读的。Style 属性可用于多种控件,如复选框、组合框、列表框、单选按钮和命令按钮等。当用于命令按钮时,可以取以下两种值:

(1)0(符号常量 `vbButtonStandard`):标准样式。在命令按钮中只显示文本(Caption 属性),没有相关的图形。此为默认设置。

(2)1(符号常量 `vbButtonGraphical`):图形格式。控件用图形样式显示,在命令按钮中不仅显示文本(Caption 属性),而且可以显示图形(Picture)。

## 16. Picture

用该属性可以给命令按钮指定一个图形。

注意:为了使用这个属性,必须把 Style 属性设置为 1(图形格式),否则 Picture 属性无效。

#### 17. DownPicture

该属性用来设置当控件被单击并处于按下状态时在控件中显示的图形,可用于复选框、单选按钮和命令按钮。

注意:为了使用这个属性,同样要把 Style 属性设置为 1(图形格式),否则 DownPicture 属性将被忽略。

#### 18. DisabledPicture

该属性用来设置对一个图形的引用,当命令按钮禁止使用(即 Enabled 属性被设置为 False)时在按钮中显示该图形。

注意:和前两个属性一样,必须把 Style 属性设置为 1 才能使 DisabledPicture 属性生效。

试一试:命令按钮有 3 个关于图形的属性,分别用于在不同的时刻显示的图形。你来设置一下这 3 个属性,体会它们的区别。别忘了把 Style 属性设置为 1。

### 4.2.2 命令按钮的常用事件

在 Windows 应用程序中,我们单击某个按钮,就会引入一个新的操作,这就是命令按钮的单击事件。命令按钮最常用的事件是单击(Click)事件,当单击一个命令按钮时,触发 Click 事件。

注意:命令按钮不支持双击(DblClick)事件。

【例 4-1】设计一个命令按钮,单击该命令按钮后,在窗体上会显示如图 4-7 所示的图形。

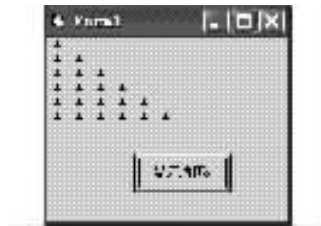


图 4-7 单击命令按钮显示图形

首先在窗体上建立一个命令按钮,并将 Caption 属性设置为“显示图形”,然后编写命令按钮的单击事件。



编写的事件过程如下:

```
Private Sub Command1_Click()           '在其单击事件中编程
    For i = 1 To 6                     '外循环,控制输出几行
        For j = 1 To i                 '内循环,控制输出几列
            Print " * ";
        Next j
        Print                           '换行
    Next i
End Sub
```

运行程序,单击命令按钮,窗体上显示出三角形的 \* 图案。加了一个命令按钮的感觉是

不是更有意思?

## 4.3 文本控件

在 Visual Basic 中,和文本相关的标准控件有两个:标签和文本框。在标签中只能显示文本,不能进行编辑,而在文本框中既可显示文本,又可输入文本。在 Visual Basic 工具箱中,标签的图标为 ,其默认名称(Name)和标题(Caption)为 Labelx(x 为 1,2,3,…) ;文本框的图标为 ,标签的默认名称和标题为 Textx(x 为 1,2,3,…)。

### 4.3.1 标签

标签主要用来显示文本信息,它所显示的内容只能用 Caption 属性来设置或修改,不能直接编辑。

说明:标签不能直接编辑,一般用来为其他控件附加一些描述信息。

#### 1. 标签的属性

标签的部分属性和命令按钮相同,除了 FontBold、FontItalic、FontName、FontSize、FontUnderline、Height、Left、Name、Top、Visible 外,还有其他属性,下面分别进行介绍。

##### (1) Alignment

该属性用来确定标签中标题的对齐方式,可以设置为:

- 0: 标题靠左对齐(默认)。
- 1: 标题靠右对齐。
- 2: 标题居中显示。

##### (2) AutoSize

该属性决定控件是否自动调整大小。如果把该属性设置为 True,则可根据 Caption 属性指定的标题自动调整标签的大小;如果把 AutoSize 属性设置为 False,则标签将保持设计时定义的大小,在这种情况下,如果标题太长,则只能显示其中的一部分。

##### (3) BorderStyle

该属性用来设置标签的边框,可以取 0 和 1 两个值。在默认情况下,该属性为 0,标签无边框;如果需要为标签加上边框,则应改变该属性的值为 1,如图 4-8 所示。



图 4-8 标签的 BorderStyle 属性

##### (4) BackStyle

该属性用来设置标签是否为透明的。当该属性值为 1 时,标签“不透明”,会覆盖背景;为 0,则标签为“透明”的,默认值为 1。

##### (5) WordWrap

该属性用于设置标签标题(Caption)的显示方式。该属性的值为 True 或 False,默认为

False。如果设置为 True,则标签将在垂直方向变化大小以与标题文本相适应,水平方向的大小与原来所画的标签相同;如果设置为 False,则标签将在水平方向上扩展到标题中最长的一行,在垂直方向上显示标题的所有行。

注意:为了使 WordWrap 起作用,应把 AutoSize 属性设置为 True。

## 2. 标签的常用事件

标签可触发 Click 和 DblClick 事件。此外,标签主要用来显示一小段文本,可以通过 Caption 属性定义,一般不使用标签事件,也不需要其他方法。

## 4.3.2 文本框

文本框是一个文本编辑区域,用户可以在这个区域中输入、编辑和显示文本,类似于一个简单的文本编辑器。

### 1. 文本框属性

文本框也具有前面介绍的命令按钮和标签的基本属性,这些属性包括 BorderStyle、Enabled、FontBold、FontItalic、FontName、FontSize、FontUnderline、Height、Left、Name、Visible、Width,此外文本框还具有如下属性:

#### (1)Text

该属性用来设置文本框中显示的内容,可以在“属性”窗口或通过程序代码设置,例如:

```
Text1.Text = "Visual Basic"
```

该语句表示将在文本框 Text1 中显示“Visual Basic”。

想一想:命令按钮和标签中显示的内容是通过哪个属性设置的?

#### (2)MaxLength

该属性用来设置允许在文本框中输入的最大字符数。如果该属性被设置为 0,则在文本框中输入的字符数不能超过 32K(多行文本)。在一般情况下,该属性使用默认值(0)。

#### (3)MultiLine

该属性用来设置文本框是否可以显示多行文本。如果把该属性设置为 False,则在文本框中只能输入单行文本;当属性 MultiLine 被设置为 True 时,可以使用多行文本,即在文本框中输入或输出文本时可以换行,并在下一行接着输入或输出。按【Ctrl+Enter】组合键可以插入一个空行。

#### (4>PasswordChar

该属性用于口令输入。在默认状态下,该属性被设置为空字符串(不是空格),用户输入的每个字符都会在文本框中显示出来。如果把 PasswordChar 属性设置为一个字符,如星号(\*),则在文本框中输入字符时,显示的不是输入的字符,而是被设置的字符(如\*)。不过文本框中的实际内容仍是输入的文本,只是显示结果被改变了。

说明:一般利用 PasswordChar 属性来设置口令。

#### (5)ScrollBars

该属性用来设置文本框中是否有滚动条,可以取 0,1,2,3 四个值,其含义分别为:

- 0:文本框中没有滚动条。
- 1:只有水平滚动条。
- 2:只有垂直滚动条。



- 3:同时具有水平和垂直滚动条。

注意:只有当 MultiLine 属性被设置为 True 时,才能用 ScrollBars 属性在文本框中设置滚动条。此外,当在文本框中加入水平滚动条(或同时加入水平和垂直滚动条)后,文本框中文本的自动换行功能将不起作用,只能通过回车键换行。

#### (6) SelLength

用来确定当前选中的字符数。当在文本框中选择文本时,该属性值会随着选择字符的多少而改变,也可以在程序代码中把该属性设置为一个整数值,由程序来改变选择。如果 SelLength 属性值为 0,则表示未选中任何字符。

注意:该属性及下面将要介绍的 SelStart、SelText 属性,只能通过程序代码设置。

#### (7) SelStart

该属性用于定义当前选择的文本的起始位置。0 表示选择的开始位置在第 1 个字符之前,1 表示从第 2 个字符之前开始选择,2 表示从第 3 个字符之前开始选择,依此类推。

#### (8) SelText

该属性值为当前所选择的文本字符串,如果没有选择文本,则该属性含有一个空字符串,如果在程序中设置 SelText 属性,则用该值代替文本框中选中的文本。例如,假设文本框 Text1 中有这样一行文本:2008 年奥运会。

选择文本“奥运会”后,如果执行语句 Text1.SelText="奥林匹克运动会"后,上述文本将变成:2008 年奥林匹克运动会。

#### (9) Locked

该属性用于设置文本框是否可以被编辑。当设置为 False(默认值)时,可以编辑文本框中的文本;当设置值为 True 时,可以滚动和选择控件中的文本,但不能编辑,即文本框被锁定。

## 2. 文本框事件和方法

文本框支持 Click、DbClick 等鼠标事件,同时支持 Chang、GotFocus、LostFocus 等事件。

#### (1) Change

当文本框中的文本发生变化时,将触发 Change 事件。该变化包括用户向文本框中输入新信息,或在程序中将 Text 属性设置为新值。程序运行时,在文本框中每输入一个字符,就会引发一次 Change 事件。

**【例 4-2】**用 Change 事件改变文本框的 Text 属性。

在窗体上建立 3 个文本框,其 Name 属性分别为 Text1、Text2、Text3,然后编写如下的事件过程:

```
Private Sub Text1_Change()  
    Text2.Text = LCase(Text1.Text)  
    Text3.Text = UCase(Text1.Text)  
End Sub
```

程序运行后,在第 1 个文本框中输入“This is the first textbox!”,输入文本时将引发第一个文本框的 Change 事件,执行 Text1\_Change 事件过程,从而在第 2、第 3 个文本框中分别用小写字母和大写字母显示文本框 Text1 中的内容,如图 4-9 所示。

#### (2) GotFocus

当文本框具有焦点(即处于活动状态)时,从键盘上输入的每个字符都将在该文本框中显示出来。



图 4-9 Change 事件示例

说明:只有当一个文本框被激活并且可见性为 True 时才能接收到焦点。

### (3) LostFocus

当按下【Tab】键使光标离开当前文本框,或者用鼠标选择窗体中的其他对象时触发该事件,即光标不在文本框中时触发该事件。

### (4) SetFocus

SetFocus 是文本框中较常用的方法,用于把输入光标(焦点)移到指定的文本框,其格式为:

[对象].SetFocus


例如,执行语句 Text2.SetFocus 后,光标将在文本框 Text2 中显示。

在课前体验中,当输入的分不符合要求时,我们使用了 SetFocus 要求重新输入分数。

技巧:如果窗体上建立了多个文本框,需要依次输入内容时,可以用该方法把光标置于所需要的文本框。

## 4.4 单选按钮、复选框和框架

### 4.4.1 单选按钮

单选按钮用于在一组选项中选择其中的一种情况,类似于我们考试时做的单项选择题。在工具箱中,单选按钮的图标为 ,其默认名称为 Optionx(其中 x 为 1,2,3...)。在一组单选按钮中,只能选择其中的一个,当选中某个单选按钮时,其他单选按钮都处于关闭状态。

#### 1. 单选按钮的常用属性

单选按钮也有 Name、Caption、Enabled、Font、Visible、Left、Top、Height、Width 等常用属性,其含义和其他控件一样。单选按钮也可以使用 Pictrue、DownPicture 和 DisabledPicture 属性,含义和命令按钮一样。此外,单选按钮还具有下列属性:

##### (1) Value

Value 属性用于设置单选按钮的状态,其值可以为 True 或 False。设置为 True 时表示该按钮被选定;设置为 False 时表示单选按钮未被选定。

##### (2) Alignment 属性

Alignment 属性用来设置单选按钮控件标题的位置。设置为 0 时,标题在按钮的右侧显示,如图 4-10(a)所示。设置为 1 时,标题在按钮的左侧显示,如图 4-10(b)所示。



(a)为0时 (b)为1时

图 4-10 单选按钮的 Alignment 属性

### (3) Style

Style 属性用来设置单选按钮的显示方式,只能在设计时使用,有 0 和 1 两个值。

- 0: 标准方式,同时显示控件和标题。
- 1: 图形方式,其外观和命令按钮类似。

### 2. 事件

单选按钮接收 Click 事件,但一般不需要编写 Click 事件过程。因为当用户单击单选按钮时,它们自动改变状态。

说明:用户单击单选按钮时,Value 值会变化,在程序中可以通过 Value 属性的值判断单击的是哪个按钮。

## 4.4.2 复选框

复选框用于在一组选项中选择多个,类似于试卷中的多项选择题。在工具箱中,复选框的图标为 ,其默认名称为 Checkx(其中 x 为 1,2,3...)。复选框用“√”表示被选中,可以同时选择多个。

### 1. 复选框常用属性

复选框的大多数属性和单选按钮相同,只是 Value 属性与单选按钮有所不同。

Value 属性用来设置复选框的状态,其值可以是 0,1,2。为 0 时,表示复选框没有被选中;为 1 时,表示复选框被选中;为 2 时,表示复选框被禁止(灰色)。

### 2. 常用事件

复选框和单选按钮一样,可以接收 Click 事件。

**【例 4-3】**用单选按钮和复选框设置文本框中显示不同的字体。

在窗体上建立两个单选按钮、两个复选框和 1 个文本框,各控件属性见表 4-1。

表 4-1 程序中使用的控件

控 件	名称(Name)	标题(Caption)	文本(Text)
文本框	Text1	无	VB 程序设计
单选按钮 1	Option1	黑体	无
单选按钮 2	Option2	宋体	无
复选框 1	Check1	加下划线	无
复选框 2	Check2	斜体显示	无

两个单选按钮和复选框的事件过程代码如下:

```
Private Sub Check1_Click()
    Text1.FontUnderline = Not Text1.FontUnderline
End Sub
```

```
Private Sub Check2_Click()
```

```

Text1.FontItalic = Not Text1.FontItalic
End Sub

Private Sub Option1_Click()
    Text1.Font = "黑体"
End Sub

Private Sub Option2_Click()
    Text1.Font = "宋体"
End Sub

```

程序运行后,可以使文本框中的文本按照单选按钮和复选框的设置显示出来,如图 4-11 所示。



图 4-11 单选按钮和复选框的使用

想一想:将上面程序中单选按钮的单击事件去掉,改为使用单选按钮的 Value 属性实现相同的功能,应该做哪些改动?

### 4.4.3 框架

框架的作用是对窗体上的控件进行分组。例如,在程序中如果需要在同一窗体中建立几组相互独立的单选按钮时,就需要用框架将每一组单选按钮框起来,这样在一个框架内的单选按钮就成为一组,它们的操作不影响框外其他组的单选按钮。使用框架对窗体中的对象进行分组,可提供视觉上的区分和总体的激活或屏蔽特性。

操作时,为把指定的控件放到框架中,必须先画框架,然后在框架内画出需要成为一组的控件,这样才能使框架内的控件成为一个整体,和框架一起移动。如果在框架外画一个控件,然后把它拖到框架内,则该控件不是框架的一部分,当移动框架时,该控件不会移动。如果要用框架对已有控件分组,可先选中所有需要框入的控件,将它们剪切(Ctrl+X)到剪贴板,然后选中框架并将剪贴板上的控件粘贴(Ctrl+V)到框架上。

#### 1. 框架属性

框架的属性包括 Enabled、FontBold、FontName、FontUnderline、Height、Left、Top、Visible、Width。框架内的所有控件将随框架一起移动、显示、消失和屏蔽。当将框架的 Enabled 属性设置为 False,程序运行时该框架在窗体中的标题正文为灰色,表示框架内的所有对象均被屏蔽,不允许用户对其进行操作。若框架的 Visible 属性设置为 False,则在程序运行期间,框架及其所有控件全部被隐藏起来。也就是说,对框架的操作也是对其内部的控件的操作。

## 2. 事件

框架可以响应 Click 和 DblClick 事件。但是,在应用程序中一般不需要编写有关框架的事件过程。

**【例 4-4】**编写程序,通过单选按钮设置字体类型和字号。

如图 4-12 所示,在窗体上建立两组单选按钮,分别放在两个框架中。



图 4-12 框架举例

两个命令按钮的事件过程如下:


```
Private Sub Command1_Click()  
    If Option1 Then  
        Text1.FontName = "宋体"  
    Else  
        Text1.FontName = "黑体"  
    End If  
    If Option3 Then  
        Text1.FontSize = 8  
    Else  
        Text1.FontSize = 10  
    End If  
End Sub  
Private Sub Command2_Click()  
    End  
End Sub
```

程序运行后,选择所需要的字体类型和字号,再单击“确定”按钮,即可改变文本框架中的字体和字号。

## 4.5 列表框和组合框

### 4.5.1 列表框

列表框可以列出多个项目供用户进行选择。而组合框可以把一个文本框和列表框组合

为单个控制窗口。在工具箱中,列表框的图标为,其默认名称为 Listx(x 为 1,2,3...)。

### 1. 列表框属性

除基本属性外,列表框还有以下特殊属性:

#### (1) Columns 属性

该属性用来确定列表框的列数。当该属性设置为 0(默认)时,所有的项目呈单列显示;如果该属性等于 1,则列表框呈多行多列显示;如果大于 1 且小于列表框中的项目数,则列表框呈单行多列显示。默认设置(0)时,如果表项的总高度超过了列表框中的项目数,则列表框的右边加上一个垂直滚动条,可以通过它上下移动列表。当 Columns 的设置值不为 0 时,如果表项的总高度超过了列表框的高度,将把部分表项移到右边一列或几列显示。当各列的宽度之和超过列表框宽度时,将自动在底部增加一个水平滚动条。

#### (2) List 属性

该属性用来列出表项的内容。List 属性保存了列表框中所有值的数组,可以通过下标访问数组中的值,其语法格式如下:

$S\$ = [\text{列表框}.]List(\text{下标})$

例如: $S\$ = List1.list(6)$ 将列出列表框 List1 第 7 项的内容。

注意:下标值是从 0 开始的。

该属性也可以改变数组中某一项已有的值,其语法格式如下:

$[\text{列表框}.]List(\text{下标}) = s\$$

例如: $List1.list(3) = \text{"内存"}$ 将把列表框 List1 第 4 项(下标值为 3)的内容设置为“内存”。

说明:在“属性”窗口输入 List 属性值时,每输入一行,按【Ctrl+Enter】组合键,接着输入下一行。

#### (3) ListCount

该属性列出列表框中表项的数量。列表框中表项的排列从 0 开始,最后一项的序号为 ListCount-1。如果列表框中共有 7 项,则最后一项的序号为 6。例如,执行  $x = List1.ListCount$  后,x 的值为列表框 List1 中的总项数。

#### (4) ListIndex

该属性值是已选中的表项的位置。表项位置由索引值(下标值)指定,第一项的索引值为 0,第二项为 1,依此类推。如果没有选中任何项,ListIndex 的值将设置为-1。在程序中设置 ListIndex 后,被选中的条目反相显示。例如,在列表框中如果选中了第 2 项(其下标值为 1),则执行语句  $x = List1.ListIndex$  后, $x = 1$ 。

#### (5) MultiSelect

该属性用来设置一次可以选择的表项数。对于一个标准列表框,该属性的设置值决定了用户是否可以在列表框中选择多个表项。MultiSelect 属性的值可以设置成以下 3 种:

- 0—None:每次只能选择一项,如果选择另一项则会取消对前一项的选择。
- 1—Simple:可以同时选择多个项,后续的选择不会取消前面所选择的项。可以用鼠标或空格键选择。
- 2—Extended:可以选择指定范围内的表项。其方法是单击所要选择的范围的第一项,然后按下【Shift】键不放,并单击所要选择的范围的最后一项。如果按住【Ctrl】键,并单击列表框中的项目,则可不连续地选择多个表项。

注意:如果选择了多个表项, ListIndex 和 Text 的属性只表示最后一次的选择值。为了确定所选择的表项, 必须检查每个元素的 Selected 属性。

#### (6) Selected

该属性实际上是一个数组, 各个元素的值为 True 或 False, 每个元素与列表框中的一项相对应。当元素的值为 True 时, 表明选择了该项; 如果为 False 则表示未选择。用下面的语句可以检查指定的表项是否被选择:

列表框.Selected(索引值)

“索引值”从 0 开始, 它实际上是数组的下标。上面的语句返回一个逻辑值(True 或 False)。用下面的语句可以选择指定的表项或取消已选择的表项:

列表框名.Selected(索引值)=True|False

注意:索引值从“0”开始。

#### (7) SelCount

如果 MultiSelect 属性设置为 1(Simple)或 2(Extended), 则该属性用于读取列表框中所选项的数目。通常它和 Selected 一起使用, 以处理控件中的所选项目。

#### (8) Sorted

该属性用来确定列表框中的项目是否按字母、数字升序排列。如果 Sorted 的属性设置为 True, 则表项按字母、数字升序排列; 如果把它设置为 False(默认), 则表项按加入列表框的先后次序排列。

#### (9) Style

这个属性用于确定控件外观, 只能在设计时确定。其取值可以设置为 0(标准形式)和 1(复选框形式)。

#### (10) Text

该属性的值为最后一次选中的表项的文本, 不能直接修改 Text 属性。

技巧:在列表框中, 如果选中了某一项, 可以使用 Text 属性得到所选项的值。这是列表框常用的一个属性。

### 2. 列表框事件

列表框接收 Click 和 DblClick 事件, 但有时不用编写 Click 事件过程代码, 而是当单击一个命令按钮或发生 DblClick 事件时, 读取 Text 属性。

### 3. 列表框方法

#### (1) AddItem

该方法用来在列表框中插入一行文本, 其格式为:

列表框.AddItem 项目字符串[, 索引值]

AddItem 方法把“项目字符串”的文本内容放入列表框中。如果省略“索引值”, 则文本被放在列表框的尾部。“索引值”可以指定插入项在列表框中的位置, 表中项目从 0 开始计数, “索引值”不能大于表中项数-1。该方法只能单个地向表中添加项目。

说明:可以使用 AddItem 方法在程序运行过程中向列表框中添加项目。

#### (2) Clear

该方法用来清除列表框中的全部内容, 其格式如下:

列表框.Clear

执行 Clear 方法后, ListCount 属性重新被设置为 0。

### (3) RemoveItem

该方法用来删除列表框中指定的项目,其格式如下:

列表框.RemoveItem 索引值

RemoveItem 方法从列表框中删除以“索引值”为地址的项目,该方法每次只能删除一个项目。

假设在窗体上建立了一个列表框 List1 和两个命令按钮 Command1、Command2,则如下过程:

```
Sub Command1_Click()
    List1.AddItem "Test:",0
End Sub
```

```
Sub Command2_Click()
    List1.RemoveItem,0
End Sub
```

可以分别向列表框中增加和删除项目。单击命令按钮 Command1,可以把字符串“Test:”加到列表框 List1 的开头;而单击命令按钮 Command2,则可删除列表框开头的一项。

**【例 4-5】**编写一个能对列表框进行项目添加、修改、删除操作的应用程序,如图 4-13 所示。



图 4-13 列表框应用举例

因为不能直接对列表框中的项目进行添加、修改和删除操作,所以用了一个文本框。列表框(ListBooks)的项目在 Form\_Load 中用 AddItem 方法添加。“添加”(cmdAdd)按钮的功能是将文本框中的内容添加到列表框,“删除”(cmdDelete)按钮的功能是删除列表框中选定的项目。如果要修改列表框,则先选定项目,然后单击“修改”(cmdModify)按钮,使所选的项目显示在文本框中,当在文本框中修改完之后再单击“修改确定”(cmdModify\_OK)按钮,更新列表框。初始时,“修改确定”按钮是不可用的,即它的 Enabled 属性为 False。

各事件过程如下:

```
Private Sub Form_Load()
```


在窗体的 Load 事件中输入列表框的各个项目



```
lstBooks.AddItem "计算机应用基础"  
lstBooks.AddItem "操作系统"  
lstBooks.AddItem "数据结构"  
lstBooks.AddItem "网络技术基础"  
End Sub  
  
Private Sub cmdAdd_Click()  
'单击添加命令按钮时将文本框中输入的内容添加到列表框中  
    lstBooks.AddItem txtItem  
    txtItem = ""  
End Sub  
  
Private Sub cmdDelete_Click()  
'删除列表框中选中的项目  
    lstBooks.RemoveItem lstBooks.ListIndex  
End Sub  
  
Private Sub cmdModify_Click()  
'所选项目显示在文本框中,等待修改  
    txtItem.Text = lstBooks.Text  
    cmdAdd.Enabled = False  
    cmdDelete.Enabled = False  
    cmdModify.Enabled = False  
    cmdModify_OK.Enabled = True  
End Sub  
  
Private Sub cmdModify_OK_Click()  
'所选项目显示在文本框中修改完成后,单击下“修改确定”按钮,更新列表框内容  
    lstBooks.List(lstBooks.ListIndex) = txtItem  
    cmdModify_OK.Enabled = True  
    cmdAdd.Enabled = True  
    cmdDelete.Enabled = True  
    cmdModify.Enabled = True  
    txtItem = ""  
End Sub
```

#### 4.5.2 组合框

组合框(ComboBox)是组合列表框和文本框的特性而成的控件。也就是说,组合框是一种独立的控件,但它兼有列表框和文本框的功能。它可以像列表框一样,让用户通过鼠标选择所需要的项目,也可以像文本框一样,用键入的方式选择项目。在工具箱中,组合框的图

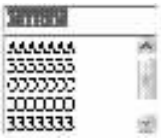
标为 。下面介绍其属性。

### 1. 组合框属性

列表框的属性基本上都可用于组合框,此外它还有自己的一些属性。

#### (1) Style 属性

这是组合框的一个重要属性,其取值可以为 0,1,2,它决定了组合框 3 种不同的类型,如图 4-14 所示。



(a) Style=0



(b) Style=1



(c) Style=2

图 4-14 组合框的 3 种类型

- 0: 取该值时,组合框为“下拉式组合框”。它看起来像一个下拉列表框,但可以输入文本或从下拉列表中选择表项。单击右端的下拉按钮可以下拉显示表项,并允许用户选择,可识别 DropDown 事件。
- 1: 取该值时,组合框为“简单组合框”,它由可输入文本的编辑区和一个标准列表框组成。列表不是下拉式的,而是一直显示在屏幕上,可以选择表项,也可以在编辑区中输入文本,它识别 DblClick 事件。在运行时,如果项目的总高度比组合框的高度大,则自动添加垂直滚动条。
- 2: 取该值时,组合框为“下拉式列表框”。和下拉式组合框一样,它的右端也有个下拉按钮,可供“拉下”或“收起”列表框,可以选择列表框中的项目。它不能识别 DblClick、Change 事件,但可识别 Dropdown 事件。

#### (2) Text 属性

该属性是用户所选择的项目的文本或直接从编辑区输入的文本。

### 2. 组合框的事件

组合框所响应的事件依赖于其 Style 属性。例如,只有简单组合框才能接收 DblClick 事件,其他两种组合框可以接收 Click 事件和 DropDown 事件。对于下拉式组合框和简单组合框,可以在编辑区输入文本,当输入文本时可以接收 Change 事件。一般情况下,用户选择项目之后,只需要读取组合框的 Text 属性。



当用户单击组合框的下拉按钮时,将触发 DropDown 事件,该事件实际上对应于单击(Click)事件。

注意:不同的组合框所能接收的事件不同。

### 3. 方法

前面介绍的 AddItem、Clear 和 RemoveItem 方法也适用于组合框,其用法与下拉列表框的相同,在此不再复述。

## 4.6 图片框和图像框

图像框和图片框主要用于在应用程序中创建图形效果。在工具箱中图片框的图标为, 图像框控件的图标为。其默认的名称分别为 Picturex 和 Imagex(x 为 1,2,3...)。

### 4.6.1 图片框

#### 1. 图片框属性

前面介绍的部分窗体属性,如 Enabled、Name、Visible、FontBold、FontItalic、FontName、FontSize、FontUnderline 等,完全适用于图片框和图像框,其用法也相同。除此之外,图片框还有一些特有的属性。

##### (1)Picture 属性

该属性用于把图形放入图片框中,可以通过“属性”窗口设置,也可以在程序运行期间进行设置。

##### (2)Autosize 属性

该属性用于调整图片框的大小以适应图形尺寸。当 Autosize 属性设置为 True 时,图形框能自动调整大小与显示的图片匹配。如果将 Autosize 属性设置为 False 时,则图片框不能自动改变大小来适应其中的图形,如果图形比控件大,则超过的部分被剪裁掉,只显示图形的一部分。

##### (3)CurrentX 和 CurrentY 属性

当使用 Print 方法向窗体或图片框中输出数据时,可以用这两个属性设置输出位置的水平(CurrentX)或垂直(CurrentY)坐标。这两个属性只能在运行期间使用,格式如下:

```
[对象].CurrentX[=x]
```

```
[对象].CurrentY[=y]
```

其中“对象”可以是窗体、图片框和打印机,x 和 y 表示横坐标值和纵坐标值,默认时以 twip 为单位。如果省略“=x”或“=y”,则显示当前的坐标值。如果省略“对象”,则指的是当前窗体。

#### 2. 图片框事件

和窗体一样,图片框可以接收 Click(单击)、DbClick(双击)事件,可以在图片框中使用(Cls)清屏和 Print 方法。

#### 3. 相关函数

可以在程序运行期间使用 LoadPicture 函数为图片框装入图形,其结果与在“属性”窗口设置图片框的 Picture 属性效果相同。其一般格式如下:

```
[对象].Picture=LoadPicture("文件名")
```

例如,Picture1.Picture=LoadPicture("c:\vb60\Graphics\1.ico"),该语句可以把存放在 C 盘的文件夹 vb60\Graphics 中的图标文件 1.ico 装入图片框中。

图片框中的图形也可以拷贝到另一个图片框中。例如,Picture2.Picture=Picture1.Picture,可以把图片框 Picture1 中的图形拷贝到图片框 Picture2 中。

注意:如果文件名省略,则删除图片框中的图形,例如,Picture1. Picture=LoadPicture()。

Visual Basic 6.0 支持以下格式的图形文件:

- Bitmap(位图):也称“绘图类型”图形,将图形定义为由点(像素)组成的图案,其文件扩展名为. BMP 或. DIB。
- Icon(图标):是一种特殊类型的位图,其最大尺寸为 32×32 像素,也可以为 16×16 像素,其文件扩展名为. ICOAK 或. CUR。
- Metafile(图元文件):也称为“绘图类型”图形,它将图形定义为编码的线段和图形。普通图元文件的扩展名为. WMF,增强型图元文件的扩展名为. EMF。

注意:在窗体、图片框或图像框中只能装入 Microsoft Windows 兼容的图元文件。

- JPEG(Joint Photographics Expert Group):是一种支持 8 位和 24 位颜色的压缩位图格式,也是 Internet 上流行的文件格式,其文件扩展名为. JPG。
- GIF(Graphics Interchange Format)是最初由 CompuServe 开发的一种压缩位图格式,支持 256 种颜色,是 Internet 上流行的文件格式,其扩展名为. GIF。

【例 4-6】编写程序,交换两个图片框中的图形。

首先在窗体上建立 3 个图片框 Picture1、Picture2、Picture3,其中 Picture1 和 Picture2 两个控件在设计阶段分别通过 Picture 属性装入了图形。



```
Private Sub Form_Click()
    Picture3. Picture=Picture1. Picture
    Picture1. Picture=Picture2. Picture
    Picture2. Picture=Picture3. Picture
    Picture3. Picture=LoadPicture(" ") '把第 3 个图片框设置为空
End Sub
```

## 4.6.2 图像框

在窗体上使用图像框的步骤和图片框相同,都可以装入多种格式的图形文件,其主要区别有以下几点:

- 图像框比图片框占用更少的内存,显示速度更快。在用图片框和图像框都能满足需要的情况下,应优先考虑使用图像框。
- 图片框是“容器”控件,可以作为父控件,而图像框不能作为父控件。
- 图片框可以通过 Print 方法接收文本,并可接收由像素组成的图形,而图像框不能接收用 Print 方法输入的信息,也不能用绘图方法在图像框上绘制图形。

## 4.7 滚动条

滚动条通常用来附在窗体上协助观察数据或确定位置,也可以用来作为数据输入的工具。滚动条有水平和垂直两种。在工具箱中,水平滚动条的图标为,垂直滚动条的图标为。其默认名称分别为 HScrollx 和 VScrollx(x 为 1,2,3...)。

除方向不同外,水平滚动条和垂直滚动条的结构和操作是一样的。滚动条的两端各有

一个滚动箭头,在滚动箭头之间有一个滚动框,如图 4-15 所示。

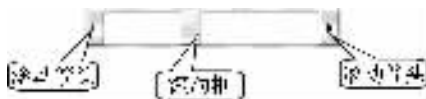


图 4-15 滚动条结构

#### 4.7.1 滚动条的常用属性

滚动条的属性用来标识滚动条的状态,除支持 Enabled、Height、Left、Caption、Top、Visible、Width 等标准属性外,还具有一些特有的属性。

##### 1. Max

该属性表示当滚动条中的滑块位于最右端或最上端时所代表的值(-32768~32767)。

##### 2. Min

该属性表示当滚动条中的滑块位于最左端或最下端时所代表的值。

##### 3. LargeChange

单击滚动条中的空白处时,滚动条 Value 属性增加或减小的增量值。

##### 4. SmallChange

单击滚动条两端的箭头时,滚动条 Value 属性增加或减小的增量值。

##### 5. Value

该属性表示滑块在滚动条上的当前位置所代表的值。如果在程序中设置该值,则把滚动框移到相应的位置。

注意:不能把 Value 属性设置为 Max 和 Min 范围之外的值。

#### 4.7.2 滚动条的常用事件

与滚动条有关的事件主要有 Scroll 和 Change。当拖动滚动框时会触发 Scroll 事件(单击滚动箭头或滚动条时不发生 Scroll 事件);而改变滚动框的位置后会触发 Change 事件。Scroll 事件用于跟踪滚动条中的动态变化,Change 事件用来获取滚动条的最后值。

【例 4-7】建立一个水平滚动条,其 Max 属性为 100,Min 属性为 0,Value 属性初始值为 50。另有一个文本框 Text1,用来显示滚动框当前位置所代表的值,如图 4-16 所示。

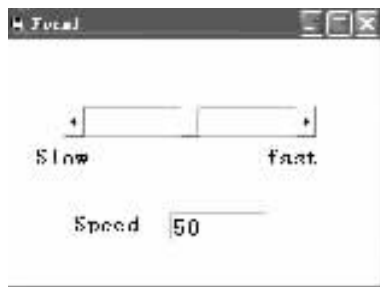



图 4-16 滚动条实例

为了让界面中的文本框能够即时显示滚动框当前位置所代表的值,编写下面事件过程:

```
Private Sub HScroll1_Change()  
    Text1.Text = HScroll1.Value  
End Sub
```

## 4.8 计时器

Visual Basic 提供的计时器控件,能有规律地以一定的时间间隔激发计时器事件而执行相应的程序代码。在工具箱中,计时器的图标为 。

### 1. 计时器常用属性

#### (1) Interval 属性

该属性用来设置计时器事件之间的间隔,单位是毫秒,其取值范围为 0~65535,因此其最大时间间隔不能超过 65 秒。如果把 Interval 属性的值设置为 1 000,则表明每秒钟发生一个计时器事件,设为 60 000 毫秒则每分钟发生 1 个计时器事件。

#### (2) Enabled 属性

该属性用于设置计时器是否可用。

### 2. Timer 事件

对于一个含有计时器控件的窗体,每经过一段由属性 Interval 指定的时间间隔,就产生一个 Timer 事件。

注意:计时器控件的大小和位置无关紧要,因为它只是在设计阶段出现在窗体上,程序运行时会自动消失。

**【例 4-8】**在窗体上建立数字时钟。

首先在窗体上建立一个计时器控件和一个标签控件,如图 4-17 所示,计时器控件 Timer1 的 Interval 属性设置为 1000;标签控件的 BorderStyle 属性设置为 1。

编写如下事件过程:

```
Private Timer1_Timer()  
    Label.Caption=Time $  
End Sub
```

运行程序,结果如图 4-18 所示。

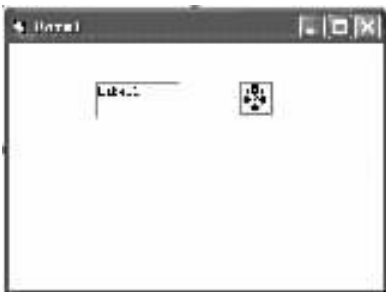


图 4-17 数字时钟界面设计



图 4-18 数字时钟运行效果

## 实 训

### 一、实训目的

1. 掌握标签、文本框、单选按钮、复选框、列表框、组合框、滚动条的常用属性、事件和方法。
2. 熟练掌握窗体和控件的事件过程代码的编写。
3. 学会利用各种控件设计友好的数据输入界面。

### 二、实训内容

设计一个用于输入教师信息的应用程序,如图 4-19 所示。当用户单击“确定”按钮后,能够在立即窗口上输出这些信息。

### 三、实训步骤

(1)建立用户界面如图 4-19 所示。民族和政治面貌是下拉式列表框,其中的项目请读者自己设置,职称是简单组合框。水平滚动条用于输入外语熟练程度(0~100)。

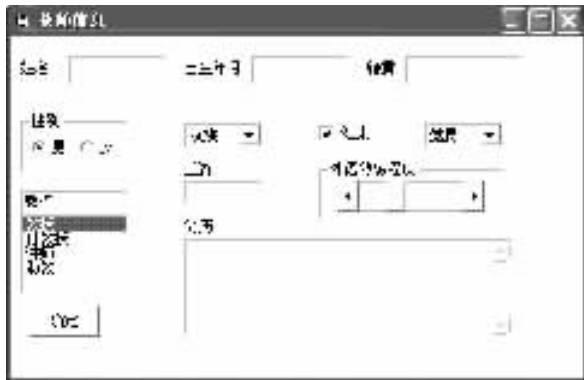


图 4-19 教师信息界面设计

(2)参照图 4-19 设置各控件的属性。

(3)编写程序代码。

### 四、实训代码

程序代码如下:

```
Private Sub Command1_Click()
    Debug.Print "姓名:"; Text1.Text
    Debug.Print "出生年月:"; Text2.Text
    Debug.Print "籍贯:"; Text3.Text
    If Option1.Value Then Debug.Print "性别:"; "男"
    If Option2.Value Then Debug.Print "性别:"; "女"
    Debug.Print "民族:"; Combo1.Text
    If Check1.Value = 1 Then Debug.Print "健康" Else Debug.Print "不健康"
    Debug.Print "政治面貌:"; Combo2.Text
    Debug.Print "职称:"; Combo3.Text
    Debug.Print "工资:"; Text4.Text

```

```
Debug.Print "外语熟练程度"; HScroll11.Value
```

```
Debug.Print "简历: "; Text5.Text
```

```
End Sub
```

程序运行后,输入各项内容,结果如图 4-20 所示,单击“命令”按钮,Debug 窗口如图 4-21 所示。



图 4-20 教师信息录入窗口



图 4-21 教师信息在立即窗口显示

## 本章小结

本章对设计 Windows 应用程序界面过程中所需要使用的界面元素进行了深入细致地介绍。在 Visual Basic 中,控件是构造应用程序用户界面的图形化工具,在程序设计中扮演着重要的角色。本章对常用内部控件介绍时结合了具体的应用实例,读者应上机自己动手实践,这样才能加深对控件的理解,从而灵活运用。在学习本章时,各控件的属性、事件和方法是重点,应该重点掌握。

## 习题 4

### 一、选择题

- 下列控件中,没有 Caption 属性的是( )。
  - 框架
  - 列表框
  - 复选框
  - 单选按钮
- 将数据项 China 添加到列表框(List1)中成为第一项应使用( )语句。
  - List1.AddItem "China",0
  - List1.AddItem "China",1
  - List1.AddItem 0,"China"
  - List1.AddItem 1,"China"
- 文本框(Text1)中有选定的文本,执行 Text1.SelText="Hello"的结果是( )。
  - “Hello”将替换掉原来选定的文本
  - “Hello”将插入到原来选定的文本之前
  - Text1.SelLength 为 5



D. 文本框中只有“Hello”

4. 在程序运行期间,如果拖动滚动条上的滚动框,则触发的滚动事件是( )。

A. Move                      B. Change                      C. Scroll                      D. GetFocus

5. 能改变窗体边框线类型的属性是( )。

A. FontStyle                  B. BorderStyle                  C. BackStyle                  D. Border

6. 用户在组合框中输入或选择的数据可以通过一个属性获得,这个属性是( )。

A. List                          B. ListIndex                      C. Text                          D. ListCount

7. 设置复选框或单选按钮标题对齐方式的属性是( )。

A. Align                          B. Alignment                      C. Sorted                          D. Value

## 二、填空题

1. 图像框中的图形能与图像框的大小相适应,应该把图像框的 Stretch 属性设置为\_\_\_\_\_。

2. 计时器事件之间的间隔通过\_\_\_\_\_属性设置。

3. 为了使标签能自动调整大小以显示全部文本内容,应把标签的\_\_\_\_\_属性设置为 True。

4. 要想在文本框中显示垂直滚动条,必须把\_\_\_\_\_属性设置为 2,同时还应把\_\_\_\_\_属性设置为\_\_\_\_\_。

5. 在窗体上画一个文本框和一个图片框,然后编写如下两个事件过程:

```
Private Sub Form_Click()
    Text1.Text = "VB 程序设计"
End Sub
```

```
Private Sub Text1_Change()
    Picture1.Print "VB Programming"
End Sub
```

程序运行后,单击窗体,则在文本框中显示的内容是\_\_\_\_\_,在图片框中显示的内容是\_\_\_\_\_。

6. 表示滚动条控件取值范围最大值的属性是\_\_\_\_\_。

7. \_\_\_\_\_属性设置为 1,单选按钮和复选框的标题显示在左边。

## 三、简答题

1. 框架的作用是什么? 如何在框架中建立控件?

2. 在列表框中添加项目有哪几种方法?

3. 组合框有哪几种类型? 能否用文本框加列表框实现组合框的功能?

4. 列表框的 Columns 属性有什么作用? 在程序运行时,是否可以修改该属性?

## 四、实验题

1. 在名称为 Form1 的窗体上画一个图片框(名称为 Picture1)、一个垂直滚动条(名称为 VScroll1)和一个命令按钮(名称为 Command1,标题为“设置属性”),通过“属性”窗口在图片框中装入一个图形,图片框的宽度与图形的宽度相同,图片框的高度任意(如图 4-22 所示)。编写适当的事件过程。程序运行后,如果单击命令按钮,则设置垂直滚动条的如下属性:

Max	2400
LargeChange	200
SmallChange	20

之后就可以通过移动滚动条上的滚动块来放大或缩小图片框的高度。运行后的窗体如图 4-22 所示。



图 4-22 运行结果

2. 在窗体上画两个命令按钮(其名称分别为 Command1 和 Command2,标题分别为“添加项目”和“删除项目”),再画一个列表框(名称为 List1)和一个文本框(名称为 Text1),如图 4-23 所示。编写适当的事件过程。程序运行后,如果单击“添加项目”命令按钮,则从键盘输入要添加到列表框中的项目;如果单击“删除项目”命令按钮,则从键盘输入要删除的项目,将其从列表框中删除。程序的运行情况如图 4-24 所示。

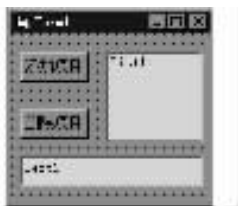


图 4-23 添加控件



图 4-24 运行结果

3. 编写如图 4-25 所示的倒计时程序。要求:

- (1)通过 InputBox 对话框输入倒计时时间(以分钟为单位)。
- (2)当倒计时时间到达 00:00:00 时,停止计时并且提示时间到。



图 4-25 计时器程序