

# 第 5 章 常用标准控件

## 本章学习目标

- 了解什么是 VB 程序设计中的标准控件和使用目的。
- 学会直线控件、形状控件和框架控件的基本使用范围和方法。
- 重点掌握单选按钮、复选框、列表框、组合框、滚动条、计时器、驱动器列表框、目录列表框和文件列表框的使用方法和技巧。
- 了解图像框和图片框控件的基本使用方法。

本章将通过实例介绍 VB 界面设计所用的各种标准控件,以期让读者快速了解和掌握这些控件的各种属性、方法和事件。

通过前面几章的学习,读者已经对 VB 面向对象程序设计方法有了初步的认识,对象是 VB 程序设计的核心。对象具有各自的属性、方法和事件,不同的对象有着不同的用途,用户可根据应用目的的不同,选择不同的对象。

在第 1 章,我们为读者介绍了几个基本控件的使用,除此之外,VB 中还经常使用复选框、单选按钮、列表框和组合框等标准控件。本章将逐步介绍这些控件的使用。

## 5.1 图形类控件


在 VB 中,与图形有关的标准控件有 4 种,即直线(Line)、形状(Shape)、图片框(PictureBox)和图像框(Image)等控件。

直线控件和形状控件,也是图形控件。直线控件可以显示水平线、垂直线或者对角线,通过属性的设置可以改变直线的粗细、颜色和样式。形状控件预定义了 6 种形状,可以显示矩形、正方形、椭圆形、圆形、圆角矩形或者圆角正方形,同时可以设置形状的颜色和填充图案。图片框控件和图像框控件主要用来为用户显示图片(包括位图、图标、图元文件、JPEG 或 GIF 文件格式的图像)。

本节将介绍直线控件、形状控件、图片框和图像框等控件的用法。

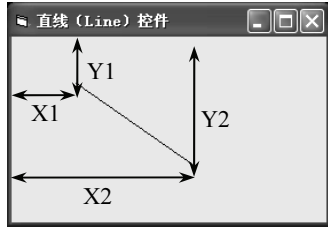
### 5.1.1 直线(Line)控件

#### 1. 常用属性

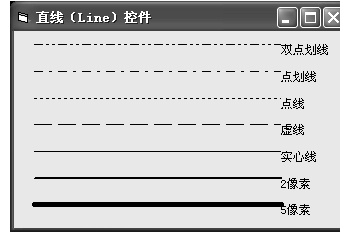
直线控件在工具箱上的图标是。直线控件除了具有 Name、Index、Visible 等常见属性外,还具有以下属性:

- (1) X1、Y1、X2、Y2 属性。返回或设置直线控件的起始点(X1,Y1)和终止点(X2,Y2)的坐标。两端点的水平坐标是 X1 和 X2;垂直坐标是 Y1 和 Y2,如图 5-1(a)所示。
- (2) BorderColor 属性。返回或设置直线的颜色。
- (3) BorderStyle 属性。返回或设置直线的样式,取值如表 5-1 所示。

(4) **BorderWidth** 属性。返回或设置直线的宽度，默认单位为像素。直线控件 **BorderColor** 和 **BorderStyle** 属性的图形化表示，如图 5-1 (b) 所示。



(a) X1、X2、Y1、Y2 属性



(b) **BorderStyle** 和 **BorderWidth** 属性

图 5-1 直线 (Line) 控件

表 5-1 **BorderStyle** 属性值

符号常数	取值	说明
vbTransparent	0	透明
vbBSSolid	1	(缺省值) 实线。边框处于形状边缘的中心
vbBSDash	2	虚线
vbBSDot	3	点线
vbBSDashDot	4	点划线
vbBSDashDotDot	5	双点划线
vbBSInsideSolid	6	内收实线

(5) **DrawMode** 属性。表示画线模式，决定 Line (Shape) 控件的外观 (也可决定以图形方法的输出外观)。DrawMode 属性值共有 16 种，取值如表 5-2 所示。

表 5-2 **DrawMode** 属性值

常数	设置值	描述
VbBlackness	1	黑色。忽略画笔色和显示色
VbNotMergePen	2	非或笔—Not((画笔色) Or (背景色))
VbMaskNotPen	3	与非笔—Not(画笔色) And (背景色)
VbNotCopyPen	4	非复制笔—Not(画笔色)
VbMaskPenNot	5	与笔非—(画笔色) And (Not (背景色))
VbInvert	6	反转—Not (背景色)
VbXorPen	7	异或笔—(画笔色) Xor (背景色)
VbNotMaskPen	8	非与笔—Not((画笔色) And (背景色))
VbMaskPen	9	与笔—(画笔色) And (背景色)
VbNotXorPen	10	非异或笔—Not((画笔色) Xor (背景色))
VbNop	11	无操作—输出保持不变。该设置实际上关闭画图
VbMergeNotPen	12	或非笔—Not(画笔色) Or (背景色)
VbCopyPen	13	复制笔 (缺省值) —由 ForeColor 属性指定的颜色
VbMergePenNot	14	或笔非—(画笔色) Or Not (背景色)
VbMergePen	15	或笔—(画笔色) Or (背景色)
VbWhiteness	16	白色。忽略画笔色和背景色

例如,窗体的背景色 (BackColor 属性) 为红色 (&H000000FF&), 在窗体上添加一个 BorderColor 属性为蓝色 (Blue) 的线条, 然后再将线条的 DrawMode 属性设置为 10, 这时线条显示的颜色为绿色。这是因为:

$$\text{Not}(\text{Blue} \text{ Xor } \text{Red}) = \text{Not}(\text{\&H00FF0000\& Xor } \text{\&H000000FF\&}) = \text{Not}(\text{\&H00FF00FF\&}) = \text{\&HFF00FF00\&}$$

即表示绿色。

(6) Tag 属性。存储程序所需的附加数据。返回或设置一个表达式用来存储程序中需要的额外数据。与其他属性不同, Tag 属性值不被 Visual Basic 使用; 可以用该属性来标识对象。利用该属性可以给对象赋予一个标识字符串, 而不会影响任何其他属性设置值或引起副作用。

## 2. 常用方法

直线控件的方法有 ZOrder 方法, 使用该方法可以决定控件放置在其图层的 Z-顺序。在设计时, ZOrder 方法的作用等同于选择“编辑”菜单中的“置前”或“置后”菜单命令。

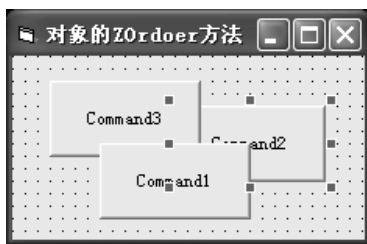
ZOrder 方法使用语法格式如下:

**对象名称.ZOrder(<位置顺序>)**

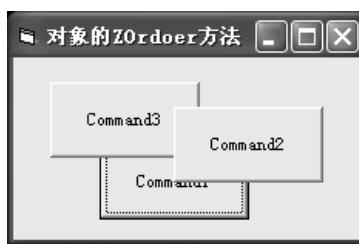
其中: <位置顺序>是一个可选项, 表示一个整数, 它用以指示同类对象相对的位置。如果 <位置顺序>为 0 或被省略, 则对象定位在 Z-顺序前面。如果 <位置顺序>为 1, 则对象定位在 Z-顺序后面。

例如, 在新建窗体中添加三个命令按钮 Command1~3, 选择“编辑”菜单中的“置前”或“置后”菜单命令, 改变三个命令按钮的顺序, 如图 5-2 (a) 所示。执行下面程序后, 观察到的按钮顺序, 如图 5-2 (b) 所示。

```
Private Sub Form_Load()
    Command1.ZOrder (1)
    Command2.ZOrder (0)
End Sub
```




(a) Command2 在图层的下方



(b) Command2 在图层的上方

图 5-2 对象 ZOrder 方法的使用

## 5.1.2 形状 (Shape) 控件

形状控件 (Shape) 在工具箱上的图标是 。使用形状控件可在窗体或其他控件容器中画出矩形、正方形、圆形、椭圆形、圆角矩形或圆角正方形。形状控件的 Shape 属性决定了它的图形样式; 形状控件没有 Enabled 属性, 也不支持任何的事件, 但有 ZOrder 和 Move 方法。

### 1. 常用属性

(1) 基本属性。和其他控件一样, 形状控件有 Name、Left、Top、Width、Height、BorderColor、BackColor、BackStyle、Index、Visible、DrawMode、Tag 等基本属性。

(2) **BorderStyle** 和 **BorderWidth** 属性。这两个属性决定了形状控件的边框样式和边框宽度。与直线控件一样，图形的边框样式有 7 种 (0~6)，分别是：透明、实线、虚线、点划线、双点划线、内部实线。**BorderWidth** 属性的单位为像素。

(3) **FillColor** 属性。填充颜色。如果要为图形填充颜色 (背景) 时，首先应该将属性 **FillStyle** (填充方式) 设置成 1 (为透明)，否则 **FillColor** (前景色) 的颜色会遮盖了背景色，达不到预期的目的。

例如，将图形的 **FillStyle** 属性的值设为 0 (实心，不透明)，**FillColor** (填充颜色) 设为蓝色，此时，不论 **BackColor** 为何种颜色，都被蓝色所遮盖，如果将 **FillStyle** 改为 1 (透明)，则图形内显示背景色。

(4) **FillStyle** 属性。填充方式，该属性用来设置图像填充的线形 (风格和样式)。使用语法格式如下：

**对象名称.FillStyle = [<数值>]**

其中：<数值>的大小，如表 5-3 所示。

表 5-3 FillStyle 属性设置值

常数	设置值	描述
VbFSSolid	0	实线
VbFSTransparent	1	(缺省值) 透明
VbHorizontalLine	2	水平直线
VbVerticalLine	3	垂直直线
VbUpwardDiagonal	4	上斜对角线
VbDownwardDiagonal	5	下斜对角线
VbCross	6	十字线
VbDiagonalCross	7	交叉对角线

(5) **Shape** 属性。该属性决定图形控件以什么形状显示 (0~5)：矩形、正方形、椭圆形、圆形、圆角矩形、圆角正方形。该属性的使用语法格式如下：

**对象名称.Shape = [<数值>]**

其中：<数值>的大小，如表 5-4 所示。

表 5-4 Shape 属性设置值

常数	设置值	描述
VbShapeRectangle	0	(缺省值) 矩形
VbShapeSquare	1	正方形
VbShapeOval	2	椭圆形
VbShapeOval	3	圆形
VbShapeRoundedRectangle	4	圆角矩形
VbShapeRoundedSquare	5	圆角正方形

【例 5-1】用 **Shape** 控件的 **Shape** 属性显示 **Shape** 控件的 6 种形状，并填充不同的图案，如图 5-3 所示。

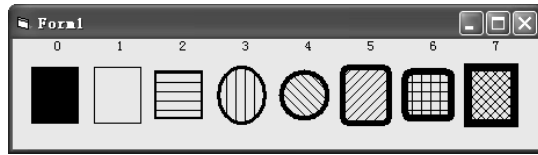


图 5-3 不同形状、填充样式的形状 (Shape) 控件

分析：在窗体上放置一个 Shape 控件，设置其 Index 属性为 0。在循环中使用 Load 方法按行的顺序产生 7 个 Shape 控件数组对象，通过改变 Shape 属性和 FillStyle 属性，形成各种形状，并填充不同的图案。

在新建工程的窗体上设置单击事件，编写代码如下：

```
Private Sub Form_Activate()
    Dim i As Integer
    Print Space(5); i; Space(3);
    Shape1(0).Shape = 0 : Shape1(0).FillStyle = 0
    For i = 1 To 7
        Print Space(3); i; Space(3);
        Load Shape1(i)           '装入数组控件
        Shape1(i).Left = Shape1(i - 1).Left + 805 '确定控件 Left 属性
        Shape1(i).Visible = True   '显示该控件
        Shape1(i).BorderWidth = i  '显示边框宽度
        If i <= 7 Then Shape1(i).Shape = i - 1 '确定所需要的几何形状
        Shape1(i).FillStyle = i    '填充不同的图案
    Next i
End Sub
```

## 2. 常用方法

形状控件支持 ZOrder 和 Move 方法。ZOrder 方法的使用与线条相同，这里介绍 Move 方法的使用。语法格式如下：



**对象名称.Move left, top, width, height**

其中：left 是必选项，表示对象左边的水平坐标 (x-轴)。而 top、width 和 height 是可选项，分别表示对象顶边的垂直坐标 (y-轴)，新的宽度和新的高度。但是，要指定任何其他的参数，必须先指定出现在语法中该参数前面的全部参数。例如，如果不先指定 left 和 top，则无法指定 width。任何没有指定的尾部的参数则保持不变。例如：

```
Shape1.Move 500, 300, 600, 700
```

## 5.2 图片框和图像框

### 5.2.1 图片框和图像框

图片框控件 (PictureBox) 和图像框控件 (Image) 在工具箱中的图标分别是  和 。图片框和图像框都能加载显示图像文件 (包括位图、图标、图元文件、JPEG 或 GIF 文件格式的图像)。但图片框功能更强，它不仅可以显示图像，还可以用绘图方法绘制图形、用 Print 方法显示文本，图片框也能作为其他控件的容器。

图片框控件不能伸缩图形以适应控件的大小，但可以自动调整控件的大小以便完整显示图形。

图像框控件占用内存比图片框少，显示速度快。可以通过属性设置伸展图片的大小使之适应控件的大小。

#### 1. 图片框的常用属性

图片框的属性除 Name、Left、Top、Height、Width、Enabled、Visible、FontBold、FontItalic、FontName、FontSize、FontUnderline、AutoRedraw 等通用属性外，还有下列主要属性：

(1) Picture 属性。默认属性，该属性用于设置控件要显示的图形。该属性不论是在属性窗口中设置还是在运行时由程序代码设置，均需要完整的路径和文件名。

(2) AutoSize 属性。返回或设置控件是否自动调整大小，以完整显示图形。如果设置为 True，控件则自动调整以适应加载的图形；如果设置为 False，若图形的原始大小比控件尺寸大，则超出部分自动被裁掉。

(3) CurrentX 和 CurrentY 属性。用来返回或设置下一个输出的水平 (CurrentX) 或垂直 (CurrentY) 坐标。这两个属性只能在运行时使用，使用格式如下：

[<对象名称>].CurrentX [= x]

[<对象名称>].CurrentY [= y]

其中，<对象名称>可以是窗体、图片框或打印机。默认单位为“缇 (Twip)”。如果省略 <对象名称>，则指当前窗体；如果省略 “=x”、“=y”，则返回当前的坐标值。

#### 2. 图像框的常用属性

图片框的大部分属性都适用于图像框。另外，图像框还具有 Stretch 属性。

Stretch 属性返回或设置一个值，该值用来指定由 Picture 属性 (默认) 设定的图形是否要自动调整大小，以适应图像框控件的大小。Stretch 属性设置为 False 时，图像框可自动改变大小以适应其中的图形；Stretch 属性设置为 True 时，加载到图像框的图形可自动调整尺寸以适应图像框的大小。图形的伸缩可能会导致图形质量的降低。

#### 3. 图片框、图像框的常用事件和方法

图片框控件和图像框控件都支持 Click 和 DblClick 等事件。

图片框控件与窗体一样，支持 Print 方法、Cls 方法以及 Circle (画圆)、Line (画直线) 和 Point (画点) 等图形方法。

### 5.2.2 图形文件的装入和保存

窗体、图片框和图像框都可以通过在设计阶段装入图形文件或在运行期间装入图形文件两种方式把 VB 所能接受的图形文件装入其中。

#### 1. 在设计阶段装入图形文件

在设计阶段装入图形有两种方法：

(1) 使用对象的 Picture 属性。选择窗体上要显示图像的图片框或图像框，然后在对象的“属性窗口”的属性列表中选择 Picture 属性，VB 弹出一个“加载图片”对话框，从中可以选择图形文件并把它加载到窗体、图片框或图像框中。

(2) 使用剪贴板。把图形从另一个应用程序 (如 Windows 的画笔) 复制到剪贴板上，然后返回 VB 环境，把它粘贴到窗体、图片框或图像框中。

#### 2. 在运行期间装入图形文件

图片框或图像框，除了在设计阶段可以通过属性窗口装载图形外，也可以在运行时用程序代码加载图形。运行时装入图形有以下两种方法。

(1) 使用 LoadPicture 函数装入图形文件。使用 LoadPicture 函数可以将指定的图形文件装入，并将该图形赋给对象的 Picture 属性。其一般格式为：

**[<对象名称>.]Picture = LoadPicture ("文件名")**

其中，<对象名称>为窗体、图片框或图像框的 Name 属性。“文件名”为要装入图形文件的全称，包括驱动器和路径。例如，在窗体上建立了一个名为 Picture1 的图片框，则运行下边的语句：

```
Picture1.Picture = LoadPicture("c:\Windows\Web\Wallpaper\Wind.jpg")
```

可以把一个图标文件 Wind.jpg 装入该图片框中，如果该图片框中已有图形，则新的图形将覆盖原有的图形。

如果省略“文件名”，即括号中的参数为空串("")时，将清除该对象所显示的图形。例如：

```
Picture1.Picture = LoadPicture()
```

清除使用 Picture 属性输入的图形时，只能通过这种方法，而不能使用 Cls 方法。

(2) 使用 Picture 属性在对象间相互复制。图形一旦被加载或粘贴到窗体、图片框或图像框控件以后，运行时就可以把它赋值给另一窗体、图片框或图像框。例如，下列语句把名为 Picture1 的图片框中的图形复制到名为 Image1 的图像框中。

```
Picture1.Picture=Image1.Picture
```

**【例 5-2】**在窗体上添加一个图片框和两个命令按钮，运行时单击“显示”按钮，程序会将一张指定的图片装入图片框，并在图片框中输出一行文字。

分析：将一张指定的图片装入图片框，可以用 LoadPicture 函数；在图片框中输出一行文字则用 Print 方法。

程序设计步骤如下：

- ①新建一工程，然后在窗体上添加一个图片框 Picture1 和两个命令按钮 Command1~2。
- ②设置命令按钮 Command1~2 的 Caption 属性值分别为“显示”和“退出”。
- ③为命令按钮 Command1~2 编写单击事件 Click 代码如下：

- 命令按钮 Command1（显示）的 Click 事件代码

```
Private Sub Command1_Click()
    Picture1.Picture = LoadPicture("c:\Windows\Web\Wallpaper\Wind.jpg") '加载图片
    Picture1.FontSize = 20
    Picture1.FontName = "隶书"
    Picture1.CurrentX = 200
    Picture1.CurrentY = 600
    Picture1.ForeColor = RGB(255, 255, 255)
    Picture1.Print "来自太空的风" '在图片框上输出文字
End Sub
```

- 命令按钮 Command2（退出）的 Click 事件代码

```
Private Sub Command2_Click()
    End
End Sub
```

运行程序的结果，如图 5-4 所示。

### 3. 图形文件的保存

可以使用语句 SavePicture 将图形文件进行保存，该语句的使用格式如下：

**SavePicture 图形文件控件名称.Image, 保存的图形文件名称**

其中“图形文件控件名称”为产生图形文件的 PictureBox 控件或 Image 控件；“保存的图



图 5-4 例 5-2 程序运行界面



形文件名称”为文件名。一般以 BMP 格式进行保存。但若图片框或图像框原来的格式为 ICON 或 Meta（元）格式，则以原格式进行保存。例如，以下语句可将图片框中的图形保存到当前工程所在文件夹中。

```
SavePicture Picture1.Image, App.Path & "\TEST1.BMP" '将图片保存到文件
```

## 5.3 选择类控件

在程序设计过程中，有时需要从多个选项中选择一个，各选项间的关系是互斥的，单选按钮（OptionButton）就能完成这个功能。复选框（CheckBox）的功能类似于单选按钮，也允许在多个选项中做出选择。但不同的是，多个单选按钮中只允许选择其中的一个，而复选框既可以选择其中的一项，也可以同时选择多项或全部不选。

### 5.3.1 单选按钮和复选框

单选按钮和复选框在工具箱中的图标分别是  和 .

#### 1. 常用属性

单选按钮和复选框的属性除了 Name、Caption、Enable、FontName、FontSize、FontBold、FontItalic、FontUnderline、Top、Height、Left、Width、BackColor、ForeColor、Visible 等通用属性外，其主要属性如下：

（1）Style 属性。用来设置单选按钮或复选框控件的显示方式，以改善视觉效果。该属性的取值范围，如表 5-5 所示。

表 5-5 Style 属性取值表（复选框、单选按钮）

符号常数	取值	说明
VbButtonStandard	0	（缺省值）标准方式，同时显示控件和标题
VbButtonGraphical	1	图形方式，控件用图形样式显示，即单选按钮和复选框的外观和命令按钮相似

Style 属性是只读属性，只能在设计时使用。使用时应注意，属性设置为不同的值（0 或 1），外观不一样，设为 1 时，控件的外观和命令按钮相似，但作用和命令按钮不相同。还可以用 Picture、DownPicture 和 DisablePicture 属性分别设置不同的图标或位图，以表示未选定、选定、和禁用。

（2）Value 属性。用来返回或设置控件的状态。对于单选按钮来说，Value 属性取值有两个：False（缺省值），表示没有选中该按钮，是关闭的，按钮是一个圆圈；True，表示该按钮被选中，按钮是打开的，中心有一个圆点。

对于复选框来说，Value 属性取值有三个：0（缺省值），没有选中该复选框；1，选中该复选框；2，该复选框不可用（变灰）。

（3）Alignment 属性。用来设置单选按钮或复选框控件的标题对齐方式。可以在设计时设置，也可以在程序运行期间设置，使用格式如下：

```
[<对象名称>].Alignment [= 值]
```

这里的<对象名称>指单选按钮或复选框，“值”取值如表 5-6 所示。



表 5-6 Alignment 属性取值表（复选框、单选按钮）

符号常数	取值	说明
VbLeftJustify	0	（缺省值）控件居左，标题在控件右侧
VbRightJustify	1	控件居右，标题在控件左侧

## 2. 常用事件和方法

单选按钮和复选框支持 Click 事件，无论何时单击复选框控件或单选按钮都将触发 Click 事件。单选按钮同时还支持 DblClick 事件。

复选框也称检查框。在执行程序时单击复选框可以使“选”和“不选”交替起作用。即每单击一次复选框都产生一个 Click 事件，以“选”和“不选”响应。

【例 5-3】设计一个应用程序，窗体上有一个标签、三个单选按钮 Option1~3 和一个由三个复选框组成的复选框数组 Check1()。程序功能是利用单选按钮和复选框改变文字的颜色和字型，程序运行界面如图 5-5 所示。

分析：判断文字的颜色和字型的改变，可利用单选按钮和复选框的 Value 属性。

程序设计步骤如下：

①新建一工程，然后在窗体上添加一个标签控件 Label1 和三个单选按钮控件 Option1~3 和一个由三个复选框组成的控件数组 Check1(2)。

②依照如图 5-5 所示的样式，修改三个单选按钮 Option1~3 和一个由三个复选框组成的控件数组 Check1(2)各元素的 Caption 属性。

③修改窗体 Form1 的 Caption 属性为“单选按钮和复选框”；设置标签 Label1 的 Caption、Alignment、BorderStyle、Font 的属性分别为“让城市生活更美好!”、“2-Center”、“1-Fixed Single”和“楷体、常规、二号”。

④编写各个控件的代码如下：

- 复选框控件数组 Check1()的 Click 事件代码

```
Private Sub Check1_Click(Index As Integer)
    Select Case Index
        Case 0
            If Check1(Index).Value = 1 Then '或 If Check1(0).Value = 1 Then
                Label1.FontBold = True
            Else
                Label1.FontBold = False
            End If
        Case 1
            If Check1(Index).Value = 1 Then '或 If Check1(1).Value = 1 Then
                Label1.FontItalic = True
            Else
                Label1.FontItalic = False
            End If
        Case 2
            If Check1(Index).Value = 1 Then '或 If Check1(2).Value = 1 Then
                Label1.FontUnderline = True
            End If
    End Select
End Sub
```



图 5-5 例 5-3 程序运行界面

```
Else
    Label1.FontUnderline = False
End If
```

```
End Select
```

```
End Sub
```

- 单选按钮 Option1（红色）的 Click 事件代码

```
Private Sub Option1_Click()
    Label1.ForeColor = RGB(255, 0, 0)
End Sub
```

- 单选按钮 Option2（绿色）的 Click 事件代码


```
Private Sub Option2_Click()
    Label1.ForeColor = RGB(0, 255, 0)
End Sub
```

- 单选按钮 Option3（蓝色）的 Click 事件代码

```
Private Sub Option3_Click()
    Label1.ForeColor = RGB(0, 0, 255)
End Sub
```

- ⑤最后运行程序，观察效果是否和预想的一致。

### 5.3.2 框架

框架控件（Frame）在工具箱上的图标是。

框架控件的主要作用是将控件进行分组。该控件是一个容器控件，它可以把不同的对象放在一个框架中，当框架设置为不能操作或不可见时，框架中的控件也变得不能操作或不可见。

为了将控件分组，首先需要绘制框架控件，然后绘制框架控件里面的控件，这样就可以把框架和里面的控件同时移动。如果在框架控件外绘制了一个控件并把它移到框架内部，那么该控件将在框架的上部，不属于框架的一部分，这时移动框架控件，该控件不会随框架一起移动。如果希望把已经存在的若干控件放在某个框架中，可以先选择所有控件，将它们剪贴到剪贴板上，再把它们粘贴到框架上。

为了选择框架内的控件，必须在框架处于非活动状态时，按住 Ctrl 键，然后用单击鼠标或框选，框住要选择的控件，选择一个或多个控件。

#### 1. 常用属性

框架的主要属性除 Name、BackColor、Font、Height、Index、Left、Top、Tag、Visible、Width 等外，常用的主要属性有两个。

（1）Caption 属性。框架的标题，即框架的可见文字部分。

（2）Enabled 属性。一般设为 True，即框架内的对象是“活动”的。如果设为 False，则框架标题变灰，框架中的所有对象均被屏蔽。

#### 2. 常用事件和方法

框架控件支持 Click 和 DblClick 事件，通常不编写这样的事件过程。

【例 5-4】在例 5-3 的基础上，将三个单选按钮控件 Option1~3 和三个复选框控件 Check1()进行分组，程序运行界面如图 5-6 所示。

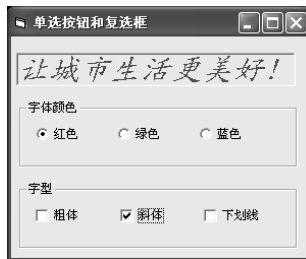





图 5-6 例 5-4 程序运行界面

将已有控件进行分组的过程如下:

(1) 按住 Ctrl 键, 点选“红色”、“绿色”和“蓝色”三个单选按钮 Option1~3, 然后按下 Ctrl+X 组合键 (或标准工具栏上的“剪切”按钮 )

(2) 点选工具箱上的框架控件 (Frame) 图标 , 然后, 在窗体上画出一个适当大小的框架 Frame1。按下 Ctrl+V 组合键 (或标准工具栏上的“粘贴”按钮 ) , 将“红色”、“绿色”和“蓝色”三个单选按钮粘贴到框架 Frame1 中。

(3) 同样的方法, 将复选框控件数组“粗体”、“斜体”和“下划线”复制到框架 Frame2 中。

(4) 修改两个框架 Frame1~2 的 Caption 属性值分别为“字体颜色”和“字型”。

(5) 各控件的程序代码同例 5-3, 在此不再列出。

## 5.4 列表框和组合框

当仅需要用户从少量选项中做出选择时, 单选按钮与复选框完全可以满足要求, 但当需要在有限空间里为用户提供大量选项时, 单选按钮和复选框就不适合了。这时, 使用 VB 为用户提供的列表框与组合框, 就可以解决此问题了。

列表框 (ListBox) 以列表形式显示一系列数据, 利用列表框, 可以接受用户对其中需要的项目的选择。组合框 (ComboBox) 则是文本框和列表框组合。

### 5.4.1 列表框

列表框控件在工具箱中的图标是 。

列表框控件是一个可提供选择的列表, 用户可以从中选择一项或多项。缺省时, 选项以垂直单列方式显示, 也可以设置成多列方式。如果待选项目超过列表框长度, 将会自动提供滚动条, 以供查看。

#### 1. 常用属性

列表框支持的属性除 Name、Left、Top、Height、Width、FontBold、FontItalic、FontName、FontSize、FontUnderline、BackColor、ForeColor、Enabled、Visible 等通用属性之外, 还具有以下主要属性:

(1) Columns 属性。该属性决定列表框控件是水平还是垂直滚动以及如何显示列中的项目。如果水平滚动, 则 Columns 属性决定显示多少列, 取值如表 5-7 所示。

表 5-7 Columns 属性取值表

取值	说明
0	(缺省值) 垂直单列列表
大于等于 1	ListBox 水平滚动并显示指定数目的列。项目安排在多个列中, 先填第一列, 再填第二列, 等等

(2) List 属性。字符串数组, 数组中的每个元素都是列表框的一个列表项内容。利用 List 属性可以输入列表项目, 在代码中可以引用该数组。List 属性数组的项目, 可以通过下标进行访问 (下标值从 0 开始, 最后一个项目的值是 ListCount-1), 语法格式为:

列表框名称. List (下标)

例如, Str=List1.List(6)

而语句:

List1.List(6)="中国"

则是将列表框中的第七项的内容设置为“中国”。

用户也可以利用该属性在设计阶段添加项目, 方法是在属性窗口中, 打开该属性, 添加一个项目后, 按下 Ctrl+Enter 组合键, 再添加第二个项目, 依次进行。在 List 属性窗口中, 添加项目的界面, 如图 5-7 所示。

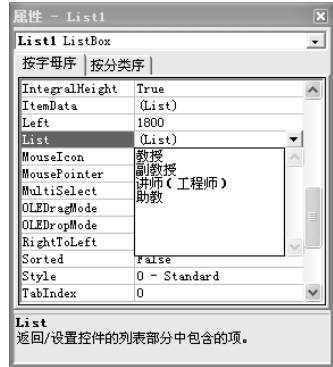


图 5-7 利用 List 属性添加项目

(3) ListCount 属性。返回列表框中项目的个数。该属性是只读的, 不能在属性窗口中设置, 只能在运行时访问它。

(4) ListIndex 属性。返回或设置列表框中已选定项目的位置。该属性也是只读的, 不能在属性窗口设置。如果未选定项目, 则 ListIndex 属性值是-1, 列表框中的第一项是 0, ListCount 属性值总是比最大的 ListIndex 值大 1。

例如, 下面语句的作用把当前选定的项目赋给标签 Label1 的标题 Caption。

Label1.Caption=List1.List(List1.ListIndex)

(5) MultiSelect 属性。只读属性, 返回或设置一个值, 该值指示是否能够在列表框控件中选择多项以及如何进行选择, 取值如表 5-8 所示。

表 5-8 MultiSelect 属性取值表

取值	说明
0-None	(缺省值) 每次只能选择一项, 不允许选择多项
1-Simple	可以同时选择多项, 鼠标单击或按下 SpaceBar (空格键) 可在列表选中或取消选中项
2-Extended	可以选择指定范围内的表项。按下 Shift 键并单击鼠标或按下 Shift 键以及一个箭头键 (上箭头、下箭头、左箭头和右箭头) 将在以前选中项的基础上扩展选择到当前选中项。按下 Ctrl 键并单击鼠标可在列表选中或取消选中项

注: 如果选择了多个表项, 则 ListIndex 和 Text 属性只表示最后一次的选择值。

(6) Selected 属性。只读属性, 返回或设置在列表框控件中的一个项的选择状态。该属性是一个布尔型数组, 数组中的每个元素与列表框中的一项相对应。当元素为 True 时, 表明选择了该项; 如果为 False, 则表示未选择。

Selected 属性的使用语法格式如下:

**列表框名称.Selected(下标)**

其中, 如果返回 True, 说明指定表项被选择; 返回 False, 说明指定表项未选择。

还可以通过该属性选择指定的表项或者取消已选择的表项, 使用语法如下:

**列表框名称.Selected(下标) [= True|False]**

(7) SelCount 属性。返回在列表框控件中被选中项的数目。如果没有项被选中, 那么 SelCount 属性将返回 0 值; 否则, 它返回当前被选中的列表项的数目。该属性对能够做多项选择是尤其有用的。

(8) Sorted 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; 如果设置为 False (默认值), 则表项按添加到列表框中的先后顺序排列。

(9) Style 属性。此属性决定控件外观，在设计时使用。Style 属性的取值如表 5-9 所示。

表 5-9 Style 属性取值表（列表框）

符号常数	取值	说明
VbListBoxStandard	0	缺省值为标准的。ListBox 控件按它在 Visual Basic 老版本中的样子显示
VbListBoxCheckbox	1	复选框。在 ListBox 控件中，每一个文本项的边上都有一个复选框。在 ListBox 中可以选择多项

(10) Text 属性。返回列表框最后一次选中项目的文本，是字符串类型。在列表框中该属性为只读属性，界面设计阶段不能修改。列表框的 Text 属性是默认属性，因此，如果把在列表框中选定列表项的值赋给某个控件，如文本框 Text 控件时，可省略不写，即可写成如下形式：

Text1=List1

## 2. 常用事件和方法

列表框支持 Click、DblClick、GotFocus、LostFocus 等事件，代码中常用 DblClick 事件过程。

列表框可以使用的方法有 AddItem、RemoveItem 和 Clear。利用它们可以在程序运行期间修改列表框的内容。

(1) AddItem 方法。向列表框添加一个新项目，格式是：

**列表框名称.AddItem 项目字符串 [序号]**

其中，“项目字符串”是要添加到列表框中的新项目；“序号”是要添加的位置，其值为整数，若省略，则表示将新项目添加到最后。对于列表框或后面介绍的组合框来讲，项目首项序号为 0。

(2) RemoveItem 方法。从列表框中删除一个项目，格式是：

**列表框名称.RemoveItem 序号**

其中，序号是要删除的项目的顺序号，其值为整数，范围为 0~ListCount-1。

(3) Clear 方法。删除列表框中的所有项目，格式是：

**列表框名称.Clear**

【例 5-5】编写一个程序，判断一个数是否为“同构数”。“同构数”是指该数出现在其平方数的右端，如 5，其平方为 25，则 5 出现在平方数的右侧。

要求：在窗体中添加一个列表框，并将 10~999 添加到列表框项目中，选择一个数，单击“判别”按钮，将判别的信息显示在文本框中，程序运行界面如图 5-8 所示。

分析：设  $n$  表示 10~999 的数，将数添加到列表框中可使用 `List1.AddItem n` 语句；2 位同构数  $n$  应满足的条件是： $n=n^2 \text{ Mod } 100$ ；3 位同构数  $n$  应满足的条件是： $n=n^2 \text{ Mod } 1000$ 。

程序设计步骤如下：

①新建一工程，然后在窗体上添加一个列表框控件 List1、一个文本框控件 Text1 和一个命令按钮控件 Command1（判别）。文本框控件 Text1 的 MultiLine 属性值为 True。

②编写窗体和“判别”命令按钮 Command1 的事件代码如下：

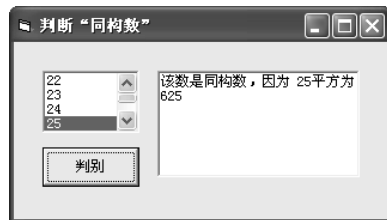


图 5-8 例 5-5 程序运行界面

- “判别”按钮 Command1 的 Click 事件代码

```
Dim n%
Private Sub Command1_Click()
    n = List1.Text
    If n = n ^ 2 Mod 100 Or n = n ^ 2 Mod 1000 Then
        Text1 = "该数是同构数, 因为" & Str(n) & "平方为" & Str(n ^ 2)
    Else
        Text1 = "该数不是同构数"
    End If
End Sub
```

- 窗体 Form1 的 Load 事件代码

```
Private Sub Form_Load()
    Text1 = ""
    For n = 10 To 999
        List1.AddItem n
    Next
End Sub
```

**思考题：**如果不用命令按钮 Command1（判别），而使用列表框的 Click 或 DblClick 事件，代码如何编写。

**【例 5-6】**设计一个应用程序，该程序可对流行歌曲进行排序。程序运行界面如图 5-9 所示。



图 5-9 例 5-6 程序运行界面

要求如下：

①在列表框中选择一首歌后，单击“↑”按钮该歌曲的排名上升一位，单击“↓”按钮该歌曲的排名下降一位，单击“→”按钮，此歌下榜。

②在文本框中输入新歌曲名字和排名名次后，单击“←”按钮可将该首歌加入列表框指定的位次。

③单击列表框中的一项，在文本框中显示歌曲名和排名。

④单击“清除”按钮将清除列表框中所有内容。

⑤单击“结束”按钮将结束程序。

**分析：**对一首歌曲的排名上升或下降，首先要确定该歌曲的位置，然后利用删除和添加的方法进行位置的调整。在程序，用 Music 表示已选择或输入的歌曲名，Indexn 表示该歌曲的位置或名次。

单击列表框中的一项，在文本框中显示歌曲名和排名，需要用列表框的 Click 事件。

程序设计步骤如下：

①新建一工程，然后在窗体上添加一个列表框控件 List1、两个文本框控件 Text1~2、三

个标签控件 Label1~3 和六个命令按钮控件 Command1~6。

②依照图 5-10 所示的样式，修改窗体与各控件的属性和位置布局。

③编写各个控件的相关事件代码如下：

- “↑”按钮 Command1 的 Click 事件代码

```
Dim music$, indexn% '分别表示歌曲名和排名位置
Private Sub Command1_Click()
    indexn = List1.ListIndex
    '判断已选择的歌曲是否逐步形成排名第一
    If indexn <> -1 And indexn <> 0 Then
        music = List1.List(indexn - 1)
        List1.RemoveItem indexn - 1
        List1.AddItem music, indexn
    End If
End Sub
```

- “↓”按钮 Command2 的 Click 事件代码

```
Private Sub Command2_Click()
    indexn = List1.ListIndex
    '判断已选择的歌曲是否逐步形成排名最后
    If indexn <> -1 And indexn <> List1.ListCount - 1 Then
        music = List1.List(indexn + 1)
        List1.RemoveItem indexn + 1
        List1.AddItem music, indexn
    End If
End Sub
```

- “←”按钮 Command3 的 Click 事件代码

```
Private Sub Command3_Click()
    music = Text1
    indexn = Text2
    '判断是否已输入歌曲名和有效的排名
    If music <> "" And indexn >= 0 Then 'And indexn <= List1.ListCount - 1 Then
        List1.AddItem music, indexn - 1
    End If
End Sub
```

- “→”按钮 Command4 的 Click 事件代码

```
Private Sub Command4_Click()
    indexn = List1.ListIndex
    If indexn <> -1 Then
        List1.RemoveItem indexn
    End If
End Sub
```

- “清除”按钮 Command5 的 Click 事件代码

```
Private Sub Command5_Click()
    List1.Clear
End Sub
```


- “结束”按钮 Command6 的 Click 事件代码

```
Private Sub Command6_Click()
    End
End Sub
```

- 列表框 List1 的 Click 事件代码


```
Private Sub List1_Click()
    Text1 = List1.Text
    Text2 = List1.ListIndex + 1
End Sub
```


### 5.4.2 组合框

组合框控件在工具箱中的图标是。

组合框控件是列表框与文本框组合而成的控件，但它兼有列表框和文本框的功能。其功能既可以像列表框一样，让用户通过鼠标选择需要的项目，也可以像文本框一样，用键入的方式输入项目。

组合框有三种样式，改变组合框的 Style 属性值，可以得到组合框的三种样式，如图 5-10 所示。

(1) 0-Dropdown Combo (下拉式组合框)：按下右侧的下三角箭头按钮后就会出现一个列表供用户选择其中一个项目，也可直接在文本框中输入文本，在 DropDown 等事件过程中把文本框内的文本添加到组合框中。

(2) 1-Simple Combo (简单组合框)：没有下三角箭头按钮，列表始终都显示出来。用户可以选择表项，也可以在文本框中直接输入。

(3) 2-Dropdown List (下拉式列表框)：用户不能在文本框中进行输入，只能从列表中选择一项。

对组合框来说，Text 属性只有在 Style 为 0 或 1 时可以输入文本。

组合框的 Style 属性取值不同，能响应的事件也有差异，只有简单组合框能响应 DblClick 事件，下拉式组合框和下拉式列表框能响应 DropDown 事件，所有组合框都能响应 Click 事件和 Change 事件。

组合框的方法与列表框基本相同，在此不再叙述。

【例 5-7】在文本框中输入数据，按回车键添加到下拉式组合框中，在列表框中选定项目，双击鼠标可移去选定项，如图 5-11 所示。



图 5-10 组合框的样式

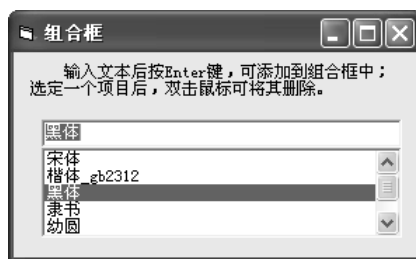


图 5-11 例 5-7 运行界面

分析：当在文本框输入一个项目，按下回车键，将发生 KeyPress 事件，要添加一个项目，可使用组合框的添加方法 AddItem，双击鼠标将引发 DblClick 事件，同时用 RemoveItem 方法删除一个项目。

设计步骤如下：

- ① 在新建工程窗体中，添加一个标签控件 Label1 和一个组合框控件 Combo1。标签控件



Label1 的 Caption 属性值设置为“输入文本后按 Enter 键，可添加到组合框中；选定一个项目后，双击鼠标可将其删除。”；设置 Combo1 控件的 Style 属性值为 1-Simple Combo（简单组合框）。安排好各控件的位置。

②编写简单组合框 Combo1 控件各事件代码如下：

- 简单组合框 Combo1 的 DblClick 事件代码



```
Private Sub Combo1_DblClick()  
    Combo1.RemoveItem Combo1.ListIndex  
End Sub
```

- 简单组合框 Combo1 的 KeyPress 事件代码

```
Private Sub Combo1_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 13 Then  
        If Combo1.Text <> "" Then '文本框内容不为空可进行添加  
            Combo1.AddItem Combo1.Text  
        End If  
        Combo1.SelStart = 0  
        Combo1.SelLength = Len(Trim(Combo1.Text))  
    End If  
End Sub
```

## 5.5 滚动条和计时器

### 5.5.1 滚动条

滚动条控件在工具箱中的图标是  和 。

滚动条可以作为数据输入工具，也可以用于辅助浏览显示内容、确定位置。主要用在列有较长项目或者大量信息的地方，这样用户就可以在小区域中巡视到所有的列表项目或者信息等。

滚动条分为两种：水平滚动条（HScrollBar）和垂直滚动条（VScrollBar）。除名称和方向不同以外，水平滚动条和垂直滚动条的结构和操作是一样的。

#### 1. 滚动条的常用属性

滚动条的属性用来识别滚动条的状态，除支持 Name、Enabled、Visible、Top、Left、Height、Width、Parent、Tag 等属性外，还具有以下属性：

(1) Min 属性。返回或设置滚动条滚动范围的下界，即滚动块位于 HScrollBar 控件的最左端或 VScrollBar 控件的最上端时所代表的值，取值范围是-32 768~32 767。

(2) Max 属性。返回或设置滚动条滚动范围的上界，即滚动块位于 HScrollBar 控件的最右端或 VScrollBar 控件的最下端时所代表的值，取值范围是-32 768~32 767。

滚动条的 Max 和 Min 属性值可以在设计阶段设定，也可以在程序中对它们赋值。一般习惯设置 Max>Min，如果 Max 的值设置得比 Min 小，那么最大值将被置于水平滚动条的最左端或垂直滚动条的最上端。

(3) LargeChange 属性。返回和设置当用户单击滚动条和滚动箭头之间的区域时，滚动条控件的 Value 属性值的增量或减量。

(4) SmallChange 属性。返回或设置当用户单击滚动箭头时，滚动条控件的 Value 属性值

的改变量。

以上两个属性，值都可以指定在 1~32767 之间（包括 1 和 32767 的整数值，默认值都设为 1）。一般在设计时设置这两个属性的值。当滚动条的状态必须改变时，也可以在运行应用程序时对其进行重新赋值。

（5）Value 属性。返回或设置滚动条的当前值（位置），其返回值始终介于 Max 和 Min 属性值之间，包括这两个值。

## 2. 滚动条的常用事件

滚动条控件最常用的事件是 Change 事件和 Scroll 事件。

在应用程序运行时，当单击滚动条两端箭头或单击滚动条的空白位置时，就会触发 Change 事件；当在滚动条内拖动滚动块时会触发 Scroll 事件，单击滚动条箭头或滚动条时不发生该事件。

【例 5-8】如图 5-12 所示，设计一个应用程序，功能可对图片进行展开操作。

分析：为实现对图像展开的上下和左右控制，可使用滚动条。

设计步骤如下：

① 在新建工程窗体中，添加一个图像控件 Image1、一个水平滚动条控件 HScroll1 和一个垂直滚动条控件 VScroll1。为窗体 Form1 和图像控件 Image1 的 Picture 属性设置一幅图片。修改窗体 Form1 的 Caption 属性为“滚动条与图像”，并安排好各控件的位置。

② 水平滚动条控件 HScroll1 和垂直滚动条控件 VScroll1 相关事件代码。

- 水平滚动条控件 HScroll1 的 Change 事件代码

```
Private Sub HScroll1_Change()  
    Image1.Width = HScroll1.Value  
End Sub
```

- 水平滚动条控件 HScroll1 的 Scroll 事件代码

```
Private Sub HScroll1_Scroll()  
    Image1.Width = HScroll1.Value  
End Sub
```

- 垂直滚动条控件 VScroll1 的 Change 事件代码

```
Private Sub VScroll1_Change()  
    Image1.Height = VScroll1.Value  
End Sub
```

- 垂直滚动条控件 VScroll1 的 Scroll 事件代码

```
Private Sub VScroll1_Scroll()  
    Image1.Height = VScroll1.Value  
End Sub
```

③ 运行程序，拖动水平和垂直滚动条，观察图片的控制情况。



图 5-12 例 5-8 程序运行界面

## 5.5.2 计时器

计时器（又称为定时器）控件在工具箱上的图标是.

计时器控件与其他控件有点不同，计时器只是在设计界面时才会显示出来，而在窗体运行时是不可见的，所以无论把计时器放在窗体的什么地方，都不会影响窗体的界面。计时器能

周期性地产生 Timer 事件, 可用来处理反复发生的动作。

### 1. 计时器的主要属性

计时器控件的基本属性有 Enabled、Index、Left、Name、Tag、Top 等, 其用法也和前面介绍的控件相同。与其他控件不同的重要属性是 Interval 属性。

计时器的作用是, 每隔一定的时间间隔 Interval 就会自动地产生一个动作 (事件) Timer。Interval 属性就是两个动作时间间隔的毫秒数, 取值范围为 0~65535, 因此其最大时间间隔不能超过 65 秒。如果该属性设置为 0 (缺省值), 则计时器控件失效。

如果把 Interval 属性设置为 1000, 则表明每秒钟发生一个计时器的动作。如果希望每秒产生 n 个事件, 则属性 Interval 的值为 1000/n。

### 2. 计时器的事件

计时器控件支持的唯一事件是 Timer 事件, 是计时器在间隔一个 Interval 时间后所触发的事件。

计时器使用的要点如下:

(1) 计时器每当计时间隔一到时就会产生一个 Timer 事件, 因此需将执行的代码编写在 Timer 事件中。

(2) 定时间隔通过 Interval 属性设置, 定时单位为毫秒。

(3) 启动计时器工作通过 Enabled 属性设置, 如果希望计时器在加载窗体时就开始工作, 需将 Enabled 属性设计为 True; 否则将 Enabled 属性设计为 False。

【例 5-9】设计一个窗体, 通过计时器控制上面的动态字符串在窗体中逐渐变大 (在大到窗体无法容纳时, 又会缩小到原来的大小), 下面的字符串从左到右滚动, 在滚动到窗体右侧边框时, 再返回原始位置继续向右滚动, 运行界面如图 5-13 所示。



图 5-13 例 5-9 程序运行界面

分析: 本题需要两个计时器控件 Timer1~2, 一个计时器用于控件动态字符串的变化, 一个计时器用于控件字符串从左到右的移动。

程序设计步骤如下:

① 创建一个新工程, 将新建工程窗体的 Caption 属性设置为 “计时器控件的使用”。

② 在窗体上添加两个标签 Label1~2, 其 Caption 属性分别设置为 “小小又变大了” 和 “快来追我啊!”。

③ 添加两个计时器, 并设置 Interval 属性值为 500。

④ 编写两个计时器的 Timer 事件。

● 计时器 Timer1 的 Timer 事件代码

```
Private Sub Timer1_Timer()
```

```

If Label1.Width > Me.Width Then
    Label1.FontSize = 9
Else
    Label1.FontSize = Label1.FontSize + 1
End If
End Sub

```

- 计时器 Timer2 的 Timer 事件代码

```

Private Sub Timer2_Timer()
    If Label2.Left > Me.Width Then
        Label2.Caption = "快来追我啊！"
    Else
        Label2.Caption = Space(2) + Label2.Caption
    End If
End Sub

```

思考题：如何设计一个“数字时钟”显示器，如图 5-14 所示。



图 5-14 “数字时钟”显示器

## 5.6 文件系统控件

为了管理计算机中的文件，VB 提供了文件系统控件，包括：驱动器列表框 (DriverListBox)、目录列表框 (DirListBox) 和文件列表框 (FileListBox)，如图 5-15 所示。



图 5-15 文件系统控件

文件系统控件能自动执行文件数据获取任务，用户也可以编写代码，自定义控件的外观并指定显示的信息。用户可以单独使用，也可以组合起来使用文件系统控件，生成操作方便的文件系统对话框。本节将介绍这些控件的功能和用法。

### 5.6.1 驱动器列表框和目录列表框

#### 1. 驱动器列表框控件

驱动器列表框控件的外观与组合框相似，用来显示驱动器名称。缺省时驱动器列表框显示当前驱动器名称。用户可以输入任何有效的驱动器名称，也可以单击右侧的下拉箭头，把系统中所有的驱动器全部下拉显示出来，如图 5-16 所示。从中选定一个驱动器，即可把它变为当前驱动器，在列表框的顶部显示。

驱动器列表框的基本属性有 Name、Enable、Top、Left、Width、Height、Visible、FontName、FontBold、FontItalic、

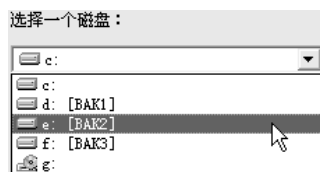


图 5-16 驱动器列表框控件的使用

FontSize 等。除此之外，驱动器列表框还有一个 Drive 属性，该属性用来返回或设置所选择的驱动器名。Drive 属性只能用程序代码设置，不能在设计时通过属性窗口设置。其格式为：

**对象名称.Drive [=驱动器名]**

例如，应用程序可通过下述简单赋值语句指定出现在驱动器列表框顶端的驱动器：

```
Drive1.Drive = "D:\\"
```

这里的“驱动器名”是指定的驱动器，如果省略，则 Drive 属性默认的是当前驱动器。如果所选择的驱动器在当前的系统中不存在，则产生错误。

驱动器列表框显示可用的有效驱动器。从列表框中选择驱动器并不能自动地变更当前工作驱动器。然而，可用 Drive 属性在操作系统级变更驱动器，这时只需将它作为 ChDrive 语句的参数：

```
ChDrive Drive.Drive
```

每次重新设置驱动器列表框的 Drive 属性时，都将触发驱动器列表框的 Change 事件。例如，要实现驱动器列表框（Drive1）与目录列表框（Dir1）同步，就要在该事件过程中写入如下代码：

```
Private Sub Drive1_Change()  
    Dir1.Path = Drive1.Drive '驱动器与目录（文件夹）同步  
End Sub
```

## 2. 目录列表框控件

目录（或文件夹）列表框控件用来显示当前驱动器上的目录和路径。外观与列表框相似，用来显示用户系统当前驱动器的目录结构（目录树），如图 5-17 所示。

程序执行时，目录列表框将显示当前目录名及其下一级目录名，如果用户选中某一个目录名，并双击它，将打开该目录，显示其子目录的结构。如果目录列表较多，将自动添加一个滚动条。

目录列表框的基本属性有 Name、Enable、Top、Left、Width、Height、Visible、FontName、FontBold、FontItalic、FontSize 等。除此之外，目录列表框还有一个 Path 属性，该属性用来返回或设置当前路径。

Path 属性适用于目录列表框和文件列表框，用来返回或设置当前驱动器的路径。Path 属性也只能用程序代码设置，不能在设计时通过属性窗口设置。其格式为：

**对象名称.Path [=路径字符串]**

其中，“路径字符串”的格式与 Dos 下相同，如果省略“=路径字符串”，则显示当前路径。例如，下面的语句将重新设置路径，目录列表框将显示 D 盘 VB 目录下的目录结构。

```
Dir1.Path = "D:\VB6.0\例子"
```

另外，Path 属性也可以直接设置限定的网络路径，如，\\网络计算机名\共享目录名\path。

如果要在程序中对指定目录及其下级目录进行操作，就要用到 List、ListCount 和 ListIndex 等属性，这些属性与列表框（ListBox）控件基本相同。

目录列表框中的当前目录的 ListIndex 值为-1，紧邻其上的目录的 ListIndex 值为-2，再上一个的 ListIndex 值为-3，依次类推。当前目录（Dir1.Path）中的第一个子目录的 ListIndex 值为 0。若当前目录下有多个子目录，则从第二个子目录起，其 ListIndex 值依次是 1、2、3、……，如图 5-18 所示。



图 5-17 目录列表框控件的使用



图 5-18 目录列表框的 ListIndex 属性值

例如，图 5-18 中 Dir1.Path 属性和 Dir1.List(Dir1.ListIndex) 的值相同。

单击目录列表框中的某个项目时，将突出显示该项目，而双击目录时则把该路径赋值给 Path 属性，同时将其 ListIndex 属性设置为 -1，然后重绘目录列表框以显示直接相邻的下级子目录。

目录列表框的重要事件是 Change 事件。当 Path 属性值改变时，将触发该事件。例如，要实现目录列表框 (Dir1) 与文件列表框 (File1) 同步，就要在目录列表框的 Change 事件过程中添加如下代码：

```
Private Sub Dir1_Change()
    File1.Path = Dir1.Path
End Sub
```

### 5.6.2 文件列表框

文件列表框控件以列表的形式显示指定目录中所包含的指定类型的文件，外观与列表框形似。

#### 1. 常用属性

文件列表框的基本属性有 Name、Enable、Top、Left、Width、Height、Visible、FontName、FontBold、FontItalic、FontSize 等。除此之外，还有以下几个常用的重要属性。

(1) FileName 属性。返回或设置所选文件的文件名。该属性只能在运行时通过代码设置，不能在设计时用属性窗口设置。

(2) ListCount、ListIndex、List、MultiSelect 属性。ListCount 属性返回控件内所列项目的总和；ListIndex 属性返回或设置控件上所选择的项目的“索引值”（即下标）；List 属性中存有文件列表框中所有项目的数组，用来返回或设置某一项目；MultiSelect 属性返回或设置一个值，该值指示是否能够在 FileListBox 控件中进行复选以及如何复选。

(3) Path 属性。用于返回或设置文件列表框的当前目录，设计时不可用。使用格式与目录列表框的 Path 属性相同。当 Path 值改变时，会引发一个 PathChange 事件。

(4) Pattern 属性。用来设置在运行时显示在 FileListBox 控件中的文件类型。该属性可以在设计时用属性窗口设计，也可以在运行时通过代码设置。在默认情况下，Pattern 的属性值为 \*.\*，即所有文件。例如，执行下面语句，将在文件列表框中只显示扩展名为 .exe 的文件。

```
File1.Pattern = "*.exe"
```

当 Pattern 属性改变时，将触发 PatternChange 事件。

在设置 Pattern 属性值时，除使用通配符外，还能够使用分号 (;) 分隔的多种模式。例如， "\*.doc; \*.docx \*.xls;\*.xlsx"，表示显示所有 Word 文件和所有 Excel 文件的列表。

(5) 使用文件属性。文件列表框中当前选定文件的属性包括 Archive、Normal、System、Hidden 和 ReadOnly。可在文件列表框中用这些属性指定要显示的文件类型。System 和 Hidden 属性的缺省值为 False，Normal、Archive 和 ReadOnly 属性的缺省值为 True。

例如，为了在列表框中只显示只读文件，直接将 ReadOnly 属性设置为 True 并把其他属性设置为 False：

```
File1.ReadOnly = True
File1.Archive = False
File1.Normal = False
File1.System = False
File1.Hidden = False
```

当 Normal = True 时，将显示无 System 或 Hidden 属性的文件；当 Normal = False 时，仍然可显示具有 ReadOnly 和/或 Archive 属性的文件，只需将这些属性设置为 True。

## 2. 主要事件

文件列表框最常用的事件有下面几个：

(1) PathChange 事件。当路径被代码中的 FileName 或 Path 值改变时，会引发一个 PathChange 事件。

说明：可使用 PathChange 事件过程引发文件列表框中路径的改变。当用包含新路径的字符串给 FileName 属性赋值时，文件列表框控件就调用了 PathChange 事件。

(2) PatternChange 事件。当 Pattern 属性改变时，将触发 PatternChange 事件。

(3) Click 和 DblClick 事件。在文件列表框中单击选中的文件，将改变 ListIndex 属性值，并将 FileName 的值设置为所单击的文件名字符串。

例如，下面语句的功能是单击可输入选中的文件名。

```
Private Sub File1_Click()
    MsgBox File1.FileName, "选中的文件名称"
End Sub
```

利用 DblClick 事件，可以执行文件列表框中的某个可执行文件。也就是说，只要双击文件列表框中的某个可执行文件，即可通过 Shell 函数来实现该文件的执行。例如：

```
Private Sub File1_DblClick()
    x = Shell(File1.FileName, 1)
End Sub
```

【例 5-10】设计一个应用界面，当单击文件列表框中的一个文件时，可显示该文件的详细信息，如图 5-19 所示。

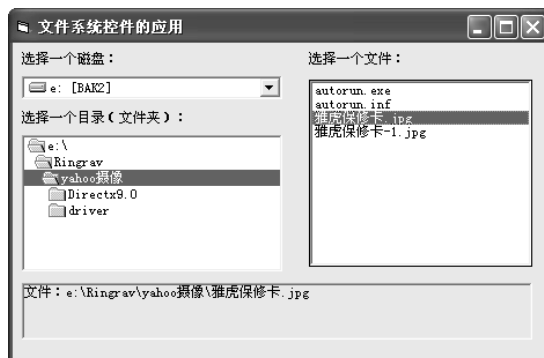


图 5-19 例 5-10 程序运行界面

分析：为了要在标签控件中显示选择文件的路径，可使用文件列表框控件的 Path 属性。程序设计步骤如下：

①创建一个新工程，将新建工程窗体的 Caption 属性设置为“文件系统控件的应用”。

②在窗体上添加四个标签 Label1~4、一个驱动器列表框 Drive1、一个目录列表框 Dir1 和一个文件列表框 File1。标签控件 Label1~3 的 Caption 属性分别设置为“选择一个磁盘：”、“选择一个目录（文件夹）：”和“选择一个文件：”；标签控件 Label14 的 Caption 和 WordWrap 属性分别设置为“”和 True。

③编写驱动器 Drive1、目录列表框 Dir1 和文件列表框 File1 相关的事件代码。

- 目录列表框 Dir1 的 Change 事件代码

```
Private Sub Dir1_Change()  
    File1.Path = Dir1.Path      '文件列表框与目录（文件夹）同步  
End Sub
```

- 驱动器列表框 Drive1 的 Change 事件代码

```
Private Sub Drive1_Change()  
    Dir1.Path = Drive1.Drive    '驱动器与目录（文件夹）同步  
End Sub
```

- 文件列表框 File1 的 Click 事件代码

```
Private Sub File1_Click()  
    Label4.Caption = "文件： " & File1.Path & "\" & File1.FileName '选中的文件名称"  
End Sub
```

## 习题五

### 一、单选题

- 下列控件中没有 Caption 属性的是（ ）。  
A) 框架                      B) 列表框                      C) 复选框                      D) 单选按钮
- 复选框的 Value 属性为 1 时，表示（ ）。  
A) 复选框未被选中                      B) 复选框被选中  
C) 复选框内有灰色的勾                      D) 复选框操作有误
- 将数据项"China"添加到列表框 List1 中成为第二项应使用（ ）语句。  
A) List1.AddItem "China",1                      B) List1.AddItem "China", 2  
C) List1.AddItem 1, "China"                      D) List1.AddItem 2, "China"
- 引用列表框 List1 最后一个数据项，应使用（ ）语句。  
A) List1.List(List1.ListCount)                      B) List1.List(ListCount)  
C) List1.List(List1.ListCount-1)                      D) List1.List(ListCount-1)
- 假如列表框 List1 有四个数据项，那么把数据项"China"添加到列表框的最后，应使用（ ）语句。  
A) List1.AddItem 3, "China"                      B) List1.AddItem "China", List1.ListCount-1  
C) List1.AddItem "China", 3                      D) List1.AddItem "China", List1.ListCount
- 执行了下面的程序后，列表框中的数据项有（ ）。  
Private Sub Form\_Click()  
 For i = 1 to 6  
 List1.AddItem i  
 Next i



```

For i = 1 to 3
    List1.RemoveItem i
Next i
End Sub
A) 1, 5, 6      B) 2, 4, 6      C) 4, 5, 6      D) 1, 3, 5

```

7. 如果列表框 List1 中没有选定的项目, 则执行语句“List1.RemoveItem List1.ListIndex”的结果是 ( )。

- A) 移去第一项                                      B) 移去最后一项  
C) 移去最后加入列表中的一项                  D) 以上都不对

8. 如果列表框 List1 中只有一个项目被用户选定, 则执行“Debug.Print List1.Selected(List1.ListIndex)”语句的结果是 ( )。

- A) 在 Debug 窗口输出被选定的项目的索引值  
B) 在 Debug 窗口输出 True  
C) 在窗体上输出被选定的项目的索引值  
D) 在窗体上输出 True

9. 假定时钟控件的 Interval 属性为 1000, Enabled 属性为 True, 并且有下面的事件过程, 程序结束后变量 x 的值为 ( )。

```

Dim x As Integer
Private Sub Timer1_Timer()
    For i = 1 to 100
        x = x + 1
        beep
    Next i
End Sub
A) 1155      B) 1000      C) 100      D) 以上都不对

```

10. 下列说法中正确的是 ( )。

- A) 通过适当的设置, 可以在程序运行期间, 让时钟控件显示在窗体上  
B) 在列表框中不能进行多项选择  
C) 在列表框中能够将项目按字母从大到小排序  
D) 框架也有 Click 和 DblClick 事件

11. 为了防止用户随意将光标置于控件之上, 应 ( )。

- A) 将控件的 TabIndex 属性设置为 0            B) 将控件的 TabStop 属性设置为 True  
C) 将控件的 TabStop 属性设置为 False       D) 将控件的 Enabled 属性设置为 False

12. 滚动条产生 Change 事件是因为 ( ) 值改变了。

- A) SmallChange    B) Value            C) Max              D) LargeChange

13. 如果要每隔 15s 产生一个 Timer 事件, 则 Interval 属性应设置为 ( )。

- A) 15                B) 900              C) 15000            D) 150

14. 列表框的 ( ) 属性是数组。

- A) List 和 ListIndex                            B) List 和 ListCount  
C) List 和 Selected                             D) List 和 Sorted

15. 使用驱动器列表框 Drive1、目录列表框 Dir1、文件列表框 File1 时, 需要设置控件的同步, 以下能够正确设置两个控件同步的命令是 ( )。

- A) Dir1.Path = Drive1.Path                    B) File1.Path = Dir1.Path  
C) File1.Path = Drive1.Path                   D) Drive1.Drive = Dir1.Path

16. 在窗体上画一个名称为 Timer 的计时器控件, 要求每隔 0.5 秒发生一次计时事件, 则以下正确的属性设置语句是 ( )。

- A) Timer.Interval=0.5                         B) Timer.Interval=5

- C) Timer.Interval=50                              D) Timer.Interval=500
17. 通过设置 Line 控件的 ( ) 属性可以绘制多种形状的图形。  
A) Shape                              B) Style                              C) FillStyle                              D) BorderStyle
18. 若在 Shape 控件内以 FillStyle 属性所指定的图案填充区域, 而填充图案的线条的颜色由 FillColor 属性指定, 非线条的区域由 BackColor 属性填充, 则应 ( )。  
A) 将 Shape 控件的 FillStyle 属性设置为 2 至 7 间的某个值, BackStyle 属性设置为 1  
B) 将 Shape 控件的 FillStyle 属性设置为 0 或 1, BackStyle 属性设置为 1  
C) 将 Shape 控件的 FillStyle 属性设置为 2 至 7 间的某个值, BackStyle 属性设置为 0  
D) 将 Shape 控件的 FillStyle 属性设置为 0 或 1, BackStyle 属性设置为 0
19. 设窗体上有一个列表框控件 List1, 且其中有若干列表项, 则以下能表示当前被选中的列表项内容的是 ( )。  
A) List1.list                              B) List1.Text                              C) List1.Index                              D) List1.listIndex
20. 下列 ( ) 程序段能删除列表 List1 中的所有项。  
A) Private Sub Command1\_Click()  
    For I=0 To List.ListCount-1  
        List1.RemoveItem 0  
    Next I  
End Sub  
B) Private Sub Command1\_Click()  
    For I=0 To List.ListCount-1  
        List1.RemoveItem 1  
    Next I  
End Sub  
C) Private Sub Command1\_Click()  
    For I=0 To List1.ListCount  
        List1.RemoveItem 0  
    Next I  
End Sub  
D) Private Sub Command1\_Click()  
    For I=0 To List1.ListCount-1  
        List1.RemoveItem 1  
    Next I  
End Sub
21. 要使列表框中的列表显示成复选框形式, 则应将其 Style 属性设置为 ( )。  
A) 0                              B) 1                              C) True                              D) False
22. 在窗体上画一个列表框和一个命令按钮, 其名称分别为 List1 和 Command1, 然后编写如下事件过程代码:
- ```
Private Sub Form_Load()
    List1.AddItem "Item1"
    List1.AddItem "Item2"
    List1.AddItem "Item3"
End Sub
Private Sub Command1_Click()
    List1.List(List1.Listcount)="AAAA"
End Sub
```
- 程序运行后, 单击命令按钮, 其结果为 ( )。  
A) 把字符串"AAAA"添加到列表框中, 但位置不能确定  
B) 把字符串"AAAA"添加到列表框的最后 (即 Item3 的后面)  
C) 把列表框中原有的最后一项改为"AAAA"  
D) 把字符串"AAAA"插入到列表框的最前面 (即 Item1 的前面)
23. 在窗体上画一个名称为 List1 的列表框, 一个名称为 Label1 的标签, 列表框中显示若干城市的名称。单击列表框中的某个城市名时, 该城市名从列表框中消失, 并在标签中显示出来。下列能正确实现上述操作的程序是 ( )。  
A) Private Sub List\_Click()  
    Label1.Caption=List1.ListIndex  
    List1.RemoveItem List1.Text  
End Sub  
B) Private Sub List\_Click()  
    Label1.Name=List1.ListIndex  
    List1.RemoveItem List1.Text  
End Sub

- C) Private Sub List\_Click()  
 Label1.Caption=List1.Text  
 List1.RemoveItem List1.ListIndex  
 End Sub
- D) Private Sub List\_Click()  
 Label1.Name=List1.Text  
 List1.RemoveItem List1.ListIndex  
 End Sub

24. 在窗体上画一个文本框和一个计时器控件, 名称分别为 Text1 和 Timer1, 在属性窗口中把计时器的 Interval 属性设置为 1000, Enabled 属性设置为 False, 程序运行后, 如果单击命令按钮, 则每隔一秒钟在文本框中显示一次当前的时间。以下是实现上述操作的程序:

```
Private Sub Command1_Click()
    Timer1. _____
End Sub
Private Sub Timer1_Timer()
    Text1.Text = Time
End Sub
```

在\_\_\_\_\_处应填入的内容是 ( )。

- A) Enabled=True    B) Enabled=False    C) Visible=True    D) Visible=False
25. 假定在图片框 Picture1 中装入了一个图形, 为了清除该图形 (不删除图片框), 应采用的正确方法是 ( )。

- A) 选择图片框, 然后按 Del 键  
 B) 执行语句 Picture1.Picture=LoadPicture("")  
 C) 执行语句 Picture1.Picture=""  
 D) 选择图片框, 在属性窗口中选择 Picture 属性, 然后按回车键

26. 在窗体上画一个名称为 Text1 的文本框, 然后画一个名称为 HScroll1 的滚动条, 其 Min 和 Max 属性分别为 0 和 100, 程序运行后, 如果移动滚动框, 则在文本框中显示滚动条的当前值。以下能实现上述操作的程序段是 ( )。

- A) Private Sub HScroll1\_Change()  
 Text1.Text = HScroll1.Value  
 End Sub
- B) Private Sub HScroll1\_Click()  
 Text1.Text = HScroll1.Value  
 End Sub
- C) Private Sub HScroll1\_Change()  
 Text1.Text = HScroll1.Caption  
 End Sub
- D) Private Sub HScroll1\_Click()  
 Text1.Text = HScroll1.Value  
 End Sub

27. 在窗体上画两个滚动条, 名称分别为 HScroll1、HScroll2; 六个标签, 名称分别为 Label1、Label2、Label3、Label4、Label5、Label6, 其中标签 Label4~Label6 分别显示“A”、“B”、“A\*B”等文字信息, 标签 Label1、Label2 分别显示其右侧的滚动条数值, Label3 显示“A\*B”的计算结果。当移动滚动框时, 在相应的标签中显示滚动条的值。当单击命令按钮“计算”时, 对标签 Label1、Label2 中显示的两个值求积, 并将结果显示在 Label3 中, 以下不能实现上述功能的事件过程是 ( )。

- A) Private Sub Command1\_Chick()  
 Label3.Caption = Str(Val(Label1.Caption) \* Val(Label2.Caption))  
 End Sub
- B) Private Sub Command1\_Chick()  
 Label3.Caption = HScroll1.Value \* HScroll2.Value  
 End Sub
- C) Private Sub Command1\_Chick()  
 Label3.Caption = HScroll1 \* HScroll2  
 End Sub
- D) Private Sub Command1\_Chick()  
 Label3.Caption = HScroll1.Text \* HScroll2.Text  
 End Sub

28. 在窗体上画两个单选按钮，名称分别为 Option1、Option2，标题分别为“宋体”和“黑体”；一个复选框，名称为 Check1，标题为“粗体”；一个文本框，名称为 Text1，Text 属性为“改变文字字体”。要求程序运行时，“宋体”单选按钮和“粗体”复选框被选中，则能够实现上述要求的语句序列是（ ）。

- A) Option1.Value=True  
Check1.Value=False
- B) Option1.Value=True  
Check1.Value=True
- C) Option2.Value=False  
Check1.Value=True
- D) Option1.Value=True  
Check1.Value=1

29. 如果只允许在列表框中每次只能选择一个列表项时，则应将其 MultiSelect 属性设置为（ ）。

- A) 0
- B) 1
- C) 2
- D) 3

30. 要将一个组合框设置为简单组合框 (Simple Combo)，则应该将其 Style 属性设置为（ ）。

- A) 0
- B) 1
- C) 2
- D) 3

31. 为了能在列表框中使用 Ctrl 和 Shift 键进行多个列表项的选择，则应将列表框的 MultiSelect 属性设置为（ ）。

- A) 0
- B) 1
- C) 2
- D) 3

32. 设在窗体 Form1 上有一个列表框 List1，其中有若干个项目。要求单击列表框中某一项时，把该项显示在窗体上，正确的事件过程是（ ）。

- A) Private Sub List1\_Click()  
Print List1.Text  
End Sub
- B) Private Sub Form1\_Click()  
Print List1.Text  
End Sub
- C) Private Sub List1\_Click()  
Print Form1.Text  
End Sub
- D) Private Sub Form1\_Click()  
List1.Print List1.Text  
End Sub

33. 在窗体上有如下图所示的控件，各控件的名称与其标题相同，并有如下程序：



```
Private Sub Form_Load()
    Command2.Enabled=False
    Check1.Value=1
End Sub
```

刚运行程序时，看到的窗体外观是（ ）。

- A)
- B)
- C)
- D)

34. 设在窗体上有一个名称为 List1 的列表框, 其中有若干个项目, 如下图所示。要求选中某一项后单击 Command1 按钮, 就删除选中的项, 则正确的事件过程是 ( )。



- A) Private Sub Command1\_Click()  
List1.Clear  
End Sub
- B) Private Sub Command1\_Click()  
List1.Clear List1.ListIndex  
End Sub
- C) Private Sub Command1\_Click()  
List1.RemoveItem List1.ListIndex  
End Sub
- D) Private Sub Command1\_Click()  
List1.RemoveItem  
End Sub

35. 某人在窗体上画了一个名称为 Timer1 的计时器和一个名称为 Label1 的标签。计时器的属性设置为 Enabled=True, Interval=0。并编程如下, 希望每 2 秒在标签上显示一次系统当前时间。

```
Private Sub Timer1_Timer()  
Label1.Caption=Time$  
End Sub
```

在程序执行时发现未能实现上述目的, 那么, 他应做的修改是 ( )。

- A) 通过属性窗口把计时器的 Interval 属性设置为 2000
- B) 通过属性窗口把计时器的 Enabled 属性设置为 False
- C) 把事件过程中的 Label1.Caption=Time\$ 语句改为 Timer1.Interval=Time\$
- D) 把事件过程中的 Label1.Caption=Time\$ 语句改为 Label1.Caption=Timer1.Time
36. Shape 控件的属性有 6 种取值, 分别代表 6 种几何图形。下列不属于这 6 种几何图形的是 ( )。

- A)  B)  C)  D) 

37. 在窗体上有一个名称为 Check1 的复选框数组 (含 4 个复选框), 还有一个名称为 Text1 的文本框, 初始内容为空。程序运行时, 单击任何复选框, 则把所有选中的复选框后面的文字罗列在文本框中, 如下图所示。下面能实现此功能的事件过程是 ( )。



- A) Private Sub Check1\_Click(Index As Integer)  
Text1.Text=""  
For k = 0 To 3  
If Check1(k).value = 1 Then  
Text1.Text = Text1.Text & Check1(k).Caption & " "  
'双引号中是空格  
End If  
Next k  
End Sub

- B) Private Sub Check1\_Click(Index As Integer)  
 For k = 0 To 3  
 If Check1(k).Value = 1 Then  
 Text1.Text = Text1.Text & Check1(k).Caption & " "  
 '双引号中是空格  
 End If  
 Next k  
 End Sub
- C) Private Sub Check1\_Click(Index As Integer)  
 Text1.Text = ""  
 For k = 0 To 3  
 If Check1(k).Value = 1 Then  
 Text1.Text = Text1.Text & Check1(Index).Caption & " "  
 '双引号中是空格  
 End If  
 Next k  
 End Sub
- D) Private Sub Check1\_Click(Index As Integer)  
 Text1.Text = ""  
 For k = 0 To 3  
 If Check1(k).Value = 1 Then  
 Text1.Text = Text1.Text & Check1(k).Caption & " "  
 '双引号中是空格  
 Exit For  
 End If  
 Next k  
 End Sub

38. 窗体上有一个名为 Combo1 的组合框, 含有 5 个项目, 要删除最后一项, 正确的语句是 ( )。
- A) Combo1.RemoveItem Combo1.Text      B) Combo1.RemoveItem 4  
 C) Combo1.RemoveItem Combo1.ListCount      D) Combo1.RemoveItem 5
39. 要使两个单选按钮属于同一个框架, 正确的操作是 ( )。
- A) 先画一个框架, 再在框架中画两个单选按钮  
 B) 先画一个框架, 再在框架外画两个单选按钮, 然后把单选按钮拖到框架中  
 C) 先画两个单选按钮, 再用框架将单选按钮框起来  
 D) 以上三种方法都正确
40. 窗体上有一个名称为 Frame1 的框架, 若要把框架上显示的“Frame1”改为汉字“框架”, 下面正确的语句是 ( )。
- A) Frame1.Name="框架"      B) Frame1.Caption="框架"  
 C) Frame1.Text="框架"      D) Frame1.Value="框架"

## 二、填空题

1. 复选框 **【1】** 属性设置为 2-Grayed 时, 变成灰色, 禁止用户使用。
2. 组合框是组合了文本框和列表框的特性而形成的一种控件。 **【2】** 风格的组合不允许用户输入列表框中没有的选项。
3. 滚动条响应的重要事件有 **【3】** 和 Change。
4. 当用户单击滚动条的空白处时, 滑块移动的增量由 **【4】** 属性决定。

5. 在程序运行时, 如果将框架 【5】 属性设置为 False, 则框架的标题呈灰色, 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。

6. 下面的程序段是将列表框 List1 中重复的项目删除, 只保留一项。

```
For i = 0 To List1.ListCount - 1
    For j = List1.ListCount - 1 To 【6】 Step - 1
        If List1.List(i) = List1.List(j) Then
            【7】
        End If
    Next j
Next i
```

7. 列表框中列表项的数目可通过 【8】 属性获得。

8. 为了使计时器控件 Timer1 每隔 1 秒触发一次 Timer 事件, 应将 Timer1 控件的 【9】 属性设置为 【10】。

9. 在窗体上画一个名称为 Command1 标题为“计算”的命令按钮; 画两个文本框, 名称分别为 Text1 和 Text2; 然后画四个标签, 名称分别为 Label1、Label2、Label3 和 Label4, 标题分别为“操作数 1”、“操作数 2”、“运算结果”和空白; 再建立一个含有四个单选按钮的控件数组, 名称为 Option1, 标题分别为“+”、“-”、“\*”和“/”。程序运行后, 在 Text1 和 Text2 中输入两个数值, 选中一个单选按钮后单击命令按钮, 相应的计算结果显示在 Label4 中。请填入适当的内容, 将程序补充完整。

```
Private Sub Command1_Click()
    For i = 0 To 3
        If 【11】 = True Then
            opt = Option1(i).Caption
        End If
    Next
    Select Case 【12】
        Case "+"
            Result = Val(Text1.Text) + Val(Text2.Text)
        Case "-"
            Result = Val(Text1.Text) - Val(Text2.Text)
        Case "*"
            Result = Val(Text1.Text) * Val(Text2.Text)
        Case "/"
            Result = Val(Text1.Text) / Val(Text2.Text)
    End Select
    【13】 = Result
End Sub
```

10. 窗体上有一个组合框, 其中已输入了若干个项目。程序运行时, 单击其中一项, 即可把该项与最上面的一项交换。例如: 单击图 1 中的“重庆”, 则与“北京”交换, 得到图 2 的结果。下面是可实现此功能的程序, 请填空。

```
Private Sub Combo1_Click()
    Dim temp
    temp = Combo1.Text
    【14】 = Combo1.List(0)
    Combo1.List(0) = temp
End Sub
```



图 1



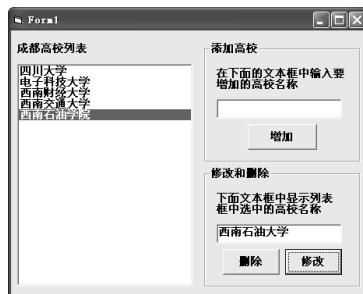
图 2

11. 如下图所示, 在列表框 List1 中已经有若干人的简单信息, 运行时在 Text1 文本框 (即“查找对象”右边的文本框) 输入一个姓或姓名, 单击“查找”按钮, 则在列表框中进行查找, 若找到, 则把该人的信息显示在 Text2 文本框中。若有多个匹配的列表项, 则只显示第一个匹配项; 若未找到, 则在 Text2 中显示“查无此人”。请填空。



```
Private Sub Command1_Click()
    Dim k As Integer, n As Integer, found As Boolean
    found = False
    n = Len(____【15】____)
    k = 0
    While k < List1.ListCount And Not found
        If Text1 = Left$(List1.List(k), n) Then
            Text2 = ____【16】____
            found = True
        End If
        k = k + 1
    Wend
    If Not found Then Text2 = "查无此人"
End Sub
```

12. 有一应用程序, 其用户界面如下图所示, 该程序用于增加、修改、删除列表框 (List1 对象) 中的项目。当用户在列表框中单击选中某一项时, 该项将出现在下边的文本框 (Text2 对象) 中, 用户可在该文本框中编辑该项, 然后单击“修改”按钮 (Command3 对象) 以修改列表框中对应内容; 若单击“删除”按钮 (Command2 对象) 则可删除用户所选中的列表框中的项目, 同时清空文本框; 当需要增加项目时, 可将该项目输入到上面的文本框 (Text1 对象) 中, 然后单击“增加”按钮 (Command1 对象) 即可。



下面给出了控件的有关事件代码, 请填空。

```
选中列表框中的项目, 并使其显示在 Text2 中
Private Sub List1_Click()
```



【17】    

End Sub  
'该按钮用于将文本框中的内容增加到列表框中

Private Sub Command1\_Click()  
    If (Text1.Text <> "") Then

    【18】    

        Text1.Text = ""

    End If

End Sub

'该按钮用于删除你所选中的列表框中的项目，同时清空文本框

Private Sub Command2\_Click()  
    If (List1.ListIndex <> -1) Then

        List1.RemoveItem     【19】    

        Text2.Text = ""

    End If

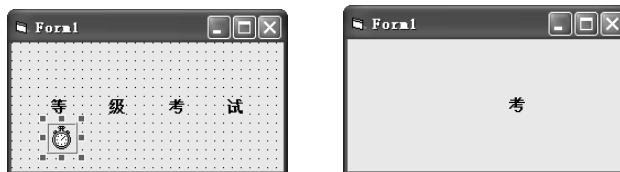
End Sub

'在文本框中编辑选中的项目后单击该按钮，用于修改所选中的列表框中的项目

Private Sub Command3\_Click()  
        【20】     = Text2.Text

End Sub

13. 在窗体上建立一个有四个元素、名称为 Label1 的标签数组，下标从 0 开始，各元素的 Caption 属性值依次为“等”、“级”、“考”、“试”；再画一个名称为 Timer1 的计时器控件，Interval 属性设为 1000，窗体如下图所示。



程序运行时，从左到右反复显示这 4 个字，但每次只显示 1 个。请填空，使程序实现上述功能。

```
Dim i As Integer
Private Sub Form_Load()
    For i = 0 To 3
        Label1(i).Visible = False
    Next
End Sub
Private Sub Timer1_Timer()
    If i     【21】     Then
        Label1(i).Visible = True
        If i = 0 Then
            Label1(3).Visible =     【22】    
        Else
            Label1     【23】    .Visible = False
        End If
        i = i + 1
    Else
        i = 0
    End If
End Sub
```

14. 窗体上有一个名称为 Combo1 的 combo 框, 其中已经添加了若干项目。程序运行时, 单击 Command1 命令按钮, 会把选中的项目从 combo 框中删除。请填写。



```
Private Sub Command1_Click()
    Dim str As String
    Dim i As Integer
    str = RTrim(____【24】____.Text)
    If str > "" Then
        For i = 0 To ____【25】____
            If Combo1.List(i) = str Then Combo1.RemoveItem ____【26】____
        Next i
    End If
End Sub
```

### 三、判断题

- ( ) 1. 如果要时钟每分钟发生一个 Timer 事件, 则 Interval 属性应设置为 1, Interval 属性值为 0 时, 表示屏蔽计时器。
- ( ) 2. 要在同一个窗体中建立几组相互独立的单选按钮时, 就要用框架将每一组单选按钮框起来。
- ( ) 3. 窗体上有一个 List1 控件和一个 Combo1 控件, 则清除 List1 列表框对象的内容的语句是 List1.Cls, 清除 Combo1 组合框的内容的语句是 Combo1.Clear。
- ( ) 4. 若在列表框中第 5 项之后插入一个项目"ABCD", 则所用语句为 List1.AddItem "ABCD",6。
- ( ) 5. 组合框的 Change 事件在用户改变组合框的选中项时被触发。
- ( ) 6. 组合框兼有文本框和列表框两者的功能, 用户可以通过键入文本或选择列表中的项目来进行选择。
- ( ) 7. 移动框架时, 框架内的控件也跟随移动, 并且框架内各控件的 Top 和 Left 属性值也将分别随之改变。
- ( ) 8. 在用户拖动滚动滑块时, 滚动条的 Change 事件连续发生。
- ( ) 9. 单选框控件和复选框控件都具有 Value 属性, 它们的作用完全一样。
- ( ) 10. 复选框不支持鼠标的双击事件, 如果双击则系统会解释为两次单击事件。
- ( ) 11. 当列表框中表项太多、超出了设计时的长度时, VB 会自动给列表框加上垂直滚动条。
- ( ) 12. 列表框和文本框一样均没有 Caption 属性, 但都有 Text 属性。
- ( ) 13. 将组合框的 Style 属性设置为 0 时, 组合框称为下拉式组合框, 选项可以从下拉列表中选择, 也可以由用户输入。
- ( ) 14. 组合框有 List 属性, 但没有 Text 属性。
- ( ) 15. 滚动条控件可作为用户输入数据的一种方法。
- ( ) 16. 由于定时器控件在运行时是不可见的, 因此在设置时可任意地将其放在任何位置。
- ( ) 17. 通用对话框 CommonDialog 的 FileName 属性返回的是一个输入或选取的文件名字符串。
- ( ) 18. 图片框 PictureBox 可以通过 Print 方法来显示文本。
- ( ) 19. 图像框和图片框都可以用 AutoSize 属性来控制控件大小调整的行为, 当 AutoSize 属性值为 True 时, 两者控件大小根据图片来调整, 设置为 False 时, 只有一部分图片可见。
- ( ) 20. 驱动器列表框、目录列表框和文件列表框三者之间能够自动实现关联。

