第5章 常用控件及键盘、鼠标事件

在 Visual Basic 中，面向对象的设计方法主要涉及的是常用的基本控件，而事件驱动则主要涉及键盘、鼠标触发的相应事件，因此本章主要针对 Visual Basic 中常用控件和事件进行学习。

5.1 列表框、组合框和滚动条

5.1.1 列表框和组合框控件

列表框(ListBox)和组合框(ComboBox)都是列表类控件，向用户提供可选择项目的列表。它们有许多相似的功能、属性、方法和事件。列表框控件提供一个项目列表，用户可以从中选择一个或多个项目。在应用程序中，可以显示多列列表项目，也可以显示单列列表项目。如果列表中的项目超过列表框可显示的数目时，控件上将自动出现滚动条，供用户浏览项目，以便选择。组合框将文本框和列表框的功能结合在一起，用户既可以在组合框中像文本框一样直接输入文本来选定项目，也可以直接从列表中选定项目。组合框控件不支持多列显示。如图 5-1 所示为列表框控件。如图 5-2 所示为组合框控件。

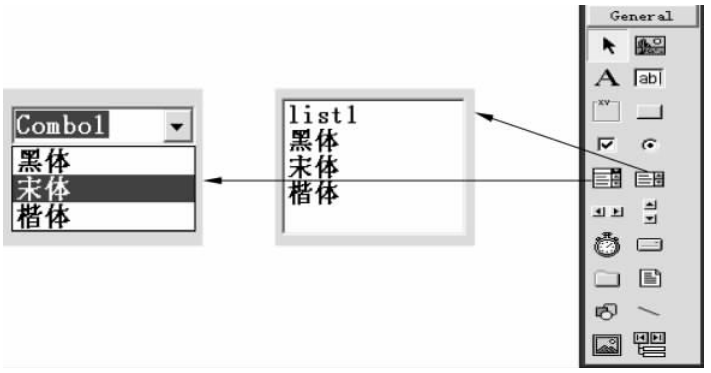

图 5-1 列表框控件　　图 5-2 组合框控件

1. 常用属性

列表框及组合框控件常用属性主要包括：Height、Index、Left、List、ListCount、ListIndex、Name、Style(组合框)、Text、Top、Width、Enabled、FontBold、FontItalic、FontName、

FontSize、Sorted、ForeColor、MultiSelect（列表框）、NewIndex、Selected（列表框）、Visible 等。

1）Columns 属性

用于指定列表框中列的数目(栏数)。列表框中的项目可以单列垂直显示，也可以水平单列或水平多列列表显示，其取值如表 5-1 所示。

表 5-1 Columns 属性值及其含义

值	描述
0	垂直单列列表
1	水平单列列表
大于 1	水平多列列表

说明：Columns 属性不适用于组合框，因为组合框不支持项目的多列显示。

2）Text 属性

用于直接返回当前选中的项目文本。该属性是一个只读属性，不能在设计时通过“属性”窗口设置，也不允许在程序运行时通过代码设置，它只用于获取当前选定的项目值。

3）List 属性

用来访问列表或组合框中的所有列表项，它是以字符串数组的方式存在的。在列表中，每一项都是 List 属性的一个元素。通过该属性，可以实现对列表框中每一列表项进行单独操作。列表框中第一个列表项的数组下标索引值为 0，最后一个列表项的数组下标索引值为 ListCount－1。

4）ListIndex 属性

用于设置或返回列表框或组合框中当前选定项目的下标索引。对于列表框，其索引的默认值为当前选中的项，对组合框而言，其索引默认值为－1。当 ListIndex 属性值为－1 时，表示当前没有列表项被选中，或者用户在组合框中输入了新的文本。

ListIndex 属性可以与 List 属性结合起来使用，共同确定选定项目的文本。如当前列表框控件名称为 List1，则 List1.List(ListIndex)的值为列表框 List1 当前选定的项目文本，它与 List1.Text 的值是完全相同的。

5）ListCount 属性

用于返回列表框或组合框中当前列表项的数目。ListCount 属性的值总是等于列表中最后一个列表项的 ListIndex 的属性值加 1。该属性是一个只读属性，不能在“属性”窗口中设置，只能在程序运行时访问它。

6）NewIndex 属性

返回最新加到列表框或组合框中列表项的下标索引值。该属性设计时不可用，运行时为只读属性。该属性主要用于已排序的列表框和组合框。当向已排序的列表框或组合框插入一项时，NewIndex 属性将会告诉你，该项插在列表中的什么位置。如果在列表中没有任何列表项，则 NewIndex 属性返回值为－1。

7）Sorted 属性

指定列表框或组合框中的项是否按字母顺序进行排列。Sorted 属性为运行时只读属性，它有两个值：True 或 False。值为 True 时，表示按字母顺序对列表中的项进行排序，排序时区分列表项中字母的大小写，同时，更改列表项的下标索引值；值为 False 时表示不对列表项进行排序。

8) MultiSelect 属性(只适用于列表框控件)

该属性可以实现在列表中同时选择多个项目。MultiSelect 属性的取值如表 5-2 所示。

表 5-2　MultiSelect 属性值及其含义

属　性　值	描　　述
0(None)	默认值,每次只能选择一个项目
1(Simple)	简单多项选择
2(Extended)	扩充多项选择

多项选择的方法:既可以同时按下 Shift 键和方向键选择彼此相邻的项目,也可以按下 Ctrl 键,用鼠标逐个选择彼此不相邻的项目。

9) Selected 属性(只适用于列表框控件)

当 MultiSelect 属性不为 0 时,它用于确定列表框中某一项的选定状态。当某一项被选中时,对应数组元素的值为 True,否则,对应的值为 False。

10) Style 属性

列表框和组合框都具有 Style 属性,该属性只能在设计时设定。

其中:

(1) 列表框的 Style 属性:用于确定列表框中列表项的表现形式,其取值有两种,为 0(Standard)表示标准列表框;为 1(Checkbox)表示在列表项的前面加上一个复选框,如图 5-3 所示。

(2) 组合框的 Style 属性:用于确定组合框的样式。其取值有以下三种。

① Style 值为 0 时,组合框为标准下拉式样式,如图 5-4 所示。

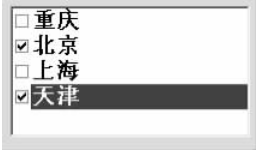

图 5-3　下拉列表样式组合框

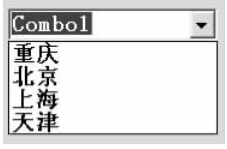

图 5-4　标准下拉式组合框

在这种情况下,用户可以直接输入文本,也可以单击组合框右侧的箭头,打开组合框所有选项列表,当用户选定了某一列表项后,该选项就插入到组合框顶部的文本框,同时关闭下拉列表。

② Style 值为 1 时,组合框为简单组合框样式。在这种情况下,用户可以直接输入文本,也可以从列表中选择项目。简单组合框的右侧没有下拉箭头,在任何时候,其列表都是显示的。当列表选项数目超过可显示的限度时,将自动添加一个垂直滚动条。

③ 当 Style 的值为 2 时,组合框为下拉列表样式,用户只能从列表中选择。

2. 常用事件

列表框和组合框的常用事件:Click、Change(组合框)、DblClick、KeyDown、KeyPress、KeyUp 等。

1）Click 事件

当单击某一列表项目时，将触发列表框与组合框控件的 Click 事件。该事件发生时系统会自动改变列表框与组合框控件的 ListIndex、Selected、Text 等属性，无须另行编写代码。

2）DblClick 事件

当双击某一列表项目时，将触发列表框与简单组合框控件的 DblClick 事件。所有类型的组合框都能响应 Click 事件，但只有简单组合框（Style 属性为 1）才能接受 DblClick 事件。

3）Change 事件（只适用于组合框控件）

对于下拉式组合框或简单组合框控件，当用户通过键盘输入改变了文本框部分的内容时，或者通过代码改变了 Text 属性的设置时，将触发 Change 事件。虽然通过单击列表选项可以改变组合框的 Text 属性值，但这并不会触发组合框的 Change 事件。

通常情况下，列表框和组合框的主要作用是通过它们的 Text 属性为应用程序的其他部分提供被选择的信息，程序员一般不需要为列表框和组合框编写事件过程代码。

3. 常用方法

主要方法有：AddItem、Clear、RemoveItem 等。

1）AddItem 方法

向列表框或组合框添加新的列表项。

调用格式：

```
控件名.AddItem Item [Index]
```

其中，

控件名：列表框或组合框控件的名称。

Item：添加到列表中的字符串表达式。

Index：指定在列表中插入新项目的位置。

例如，Index 为 0，表示将新项目添加到控件的第一个位置，如果省略该参数，对于 Sorted 属性为 True 的控件，新项目按字母顺序添加到合适的位置上；对于 Sorted 属性为 False 的控件，新项目插入到列表的末尾。

对列表项目的添加是比较灵活的，在程序运行的任何时候都可以使用该方法动态地添加项目，通常在窗体的 Load 事件中添加列表项目。

2）RemoveItem 方法

从列表框或组合框中删除指定位置的列表项。

调用格式：

```
控件名.RemoveItem Index
```

其中，Index 参数是要删除项目在列表中所在的位置。

3）Clear 方法

用于删除列表框或组合框中的所有项目。Clear 方法经常在列表刷新时使用。

4. 实例

例 5.1 使用列表框的属性、方法和事件的应用，要求从一个列表框向另外的列表框中

添加选项。其界面组成如图 5-5 所示。

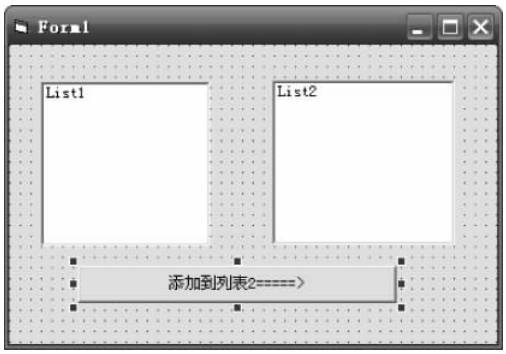

图 5-5　添加控件后的窗体

1）界面设计

在屏幕上添加一个空白的窗体，向窗体上添加两个列表框（ListBox）控件和一个命令（CommandButton）控件，如图 5-5 所示。其中，控件的属性设置如表 5-3 所示。

表 5-3　控件的属性设置

命令（CommandButton）控件	（Name）	Command1
	Caption	添加到列表 2＝＝＝＝＝＞
列表框（ListBox）控件	（Name）	List1
	MultiSelect	2-Extended
	（Name）	List2
	TabIndex	1

2）初始化代码设计

在程序代码窗口中确定窗体的 Form_Load()事件，并且在其中添加程序的初始化代码为：

```
Private Sub Form_Load()
    List1.AddItem "北京"
    List1.AddItem "上海"
    List1.AddItem "重庆"
    List1.AddItem "哈尔滨"
    List1.AddItem "深圳"
    List1.AddItem "广东"
    List1.AddItem "珠海"
    List1.AddItem "汕头"
    List1.AddItem "海南"              '以上信息是增加在第 1 个列表框控件(List1)
    List2.Clear                       '对第 2 个列表框控件(List2)清空
 End Sub
```

在以上程序初始化代码中，首先向 List1 中添加 9 个选项信息，然后通过一条语句 List2.Clear 把控件 List2 清空。

3）事件响应动作

命令按钮事件的操作，在 Command1_Click()事件中编写下列代码：

```
Private Sub Command1_Click()
   For i = 0 To List1.ListCount - 1
    If List1.Selected(i) Then            '判断是否选中列表框的内容
       List2.AddItem List1.List(i)       '把列表框 1 的内容添加到列表框 2 中
    End If
   Next i
  End Sub
```

4）程序运行结果

程序运行的界面如图 5-6 所示。在程序运行的操作中，由于执行了 List2.Clear 语句，所以开始控件 List2 被清空，在列表框 List1 中可用 Shift 键、Ctrl 键和鼠标键选择选项（由于 List1 属性中 MultiSelect 为 2-Extended，表示能进行多重选择的功能），单击“添加到列表 2=======>”命令按钮，把列表框 1 的内容复制到列表框 2 中。

图 5-6　程序运行的界面

5.1.2　滚动条

滚动条（ScrollBar）在 Windows 的工作环境中经常可以见到，当一个页面上的内容不能在当前窗口中完全显示时，可以单击滚动条两端的滚动箭头，或者拖动滚动条上的滚动块，移动窗口，浏览页面内容的不同部分。

在应用程序中，有时还可以把滚动条作为一种特殊的数据输入工具，用来对程序的运行进行某些控制。Visual Basic 在工具箱中提供了水平滚动条（HScrollBar）和垂直滚动条（VScrollBar），二者只是表现形式不同，其实功能是完全一样的，用户可以根据界面设计的需要选择适当样式的滚动条。在 Visual Basic 中，滚动条控件常常与需要浏览信息，但又不支持滚动功能的控件（如图片框控件）配合使用，为它们提供滚动浏览信息的功能；也可以作为用户信息输入的控件，如在多媒体应用程序中，使用滚动条来作为控制音量的设备。具体来说，当项目列表很长或者信息量很大时，可以通过滚动条实现简单的定位功能。此外，滚动条还可以按比例指示当前位置，以控制程序输入，作为速度、数量的指示器来使用。滚动条在工具箱中的图标为 。

滚动条是一个独立的控件，它有自己的事件、属性和方法集。其中，文本框、列表框和组合框内部在特定情况下都会出现滚动条，但它们属于这些控件的一部分，不是一个独立的控件。

1. 常用属性

滚动条的常用属性有 Height、Left、Name、Max、Min、Top、Value、Width、Enabled、

FontBold、FontItalic、FontName、FontSize、FontStrikethru、FontUnderline、ForeColor、LargeChange、SmallChange、Visible 等。

1）Value 属性

对应于滚动框在滚动条中的相对位置，其值是一个整数。

（1）对于水平滚动条：当滚动框处于最左边时，该属性取最小值。

（2）对于垂直滚动条：当滚动框处于最顶端时，该属性也取最小值。

当滚动框处于中间的各个位置时，Value 值介于最大值和最小值，并严格按照比例设定滚动框在滚动条中的位置。改变滚动条 Value 属性的方法有以下 4 种。

（1）直接在“属性”窗口中设定 Value 值；

（2）鼠标单击两端箭头键改变滚动条数值；

（3）将滚动框沿滚动条拖动到任意位置；

（4）鼠标单击滚动条中滚动框与滚动箭头之间的部分，使滚动框以翻页的速度移动。

2）Max 和 Min 属性

用于设定滚动条 Value 属性的取值范围。通常情况下，Max 代表 Value 的最大值，Min 代表 Value 属性的最小值。默认情况下，若未对 Max 和 Min 属性进行设置，Value 属性的取值在 0～32 767 变化。

3）LargeChange 和 SmallChange 属性

LargeChange 属性确定当在滚动框和滚动箭头之间单击鼠标时，Value 属性值的变化量；SmallChange 属性确定当用鼠标单击滚动条两端箭头时，Value 属性值的变化量。这两个属性的默认值都为 1，变化量应该在 Min 和 Max 属性之间进行选择。

2. 常用事件

1）Change 事件

在移动滚动框或通过代码改变其 Value 属性值时发生。可通过编写 Change 事件过程来协调各控件间显示的数据或使它们同步。

2）Scroll 事件

当滚动框被重新定位或按水平方向或垂直方向滚动时，Scroll 事件发生。在拖动滚动框时触发。

Scroll 事件与 Change 事件的区别在于：当滚动条控件滚动时，Scroll 事件一直发生，而 Change 事件只是在滚动结束之后才发生一次。

3. 实例

例 5.2　创建一个应用程序，使用滚动条来设置字体大小的程序，界面如图 5-7 所示。

图 5-7　用滚动条设置字号界面

要求如下。

（1）在文本框中输入 1～100 内的数值后，滚动条的滚动框会滚动到相应位置，同时标签的字号也会相应改变。

（2）当滚动条的滚动框的位置改变后，文本框中也

会显示出相应的数值，标签的字号也会相应改变。

各个控件及属性按如表 5-4 所示设置。

表 5-4 各对象的主要属性设置

对　象	属性(属性值)		说　明
窗体	Name(Form1)	Caption(“字号设置”)	
标签	Name(ztDisp)	Caption(“学生”)	用来显示字体
水平滚动条	Name(hsbFontSize)		用来调整字体的大小
文本框	Name(txtFontSize)		用来显示字体的大小的数字

程序代码如下。

```
Private Sub Forrm_Load()
  ztDisp.FontSize = 10                    '初始化字体大小为 10
  hsbFontSize.Min = 1
  hsbFontSize.Max = 100
  hsbFontSize.SmallChange = 1
  hsbFontSize.LargeChange = 5
  hsbFontSize.Value = 10
  txtFontSize.Text = "10"
End Sub
Private Sub hsbFontSize_Change()          '滚动条的 Change 事件
  ztDisp.FontSize = hsbFontSize.Value
  txtFontSize.Text = Str(hsbFontSize.Value)
End Sub
Private Sub txtFontSize_Change()          '文本框的 Change 事件
'下面的代码判断数据有效性
  If IsNumeric(txtFontSize.Text) And Val(txtFontSize.Text) >= _
      hsbFontSize.Min And Val(txtFontSize.text) <= hsbFontSize.Max Then
        hsbFontSize.Value = Val(txtFontSize.Text)
  Else
        txtFontSize.Text = "无效数据"
  End If
End Sub
```

5.2 单选按钮和复选框

单选按钮(OptionButton)和复选框(CheckBox)是应用程序的用户界面上常用的两类控件。这两类控件单个使用通常是没有意义的，实际应用中总是成组出现。

单选按钮和复选框都是选择类型的控件，它们有许多相同的地方，也有着明显的区别。单选按钮常以数组形式出现，有且只有一个选项被选中，多个单选按钮同处在一个容器中，只能选择其中的一个；多个复选框，也就是检查框，可以同时选中多个。

5.2.1 单选按钮

在任何时刻用户只能从单选按钮中选择一个选项，实现一种“单项选择”的功能，被选中项目左侧圆圈中会出现一个黑点。单选按钮在工具箱中的图标为  。另外，同一“容器”中

的单选按钮提供的选项是相互排斥的，即只要选中某个选项，其余选项就自动取消选中状态，如图 5-8 所示。

图 5-8　单选按钮

1. 属性

Caption 属性：设置标题内容。

Value 属性：是单选按钮控件最重要的属性，为逻辑型值，当为 True 时，表示已选择了该按钮，为 False(默认值)则表示没有选择该按钮，如表 5-5 所示。用户可在设计阶段通过“属性”窗口或在运行阶段通过程序代码设置该属性值，也可在运行阶段通过鼠标单击某单选按钮控件将其 Value 属性设置为 True。

表 5-5　Value 属性值

设置值	值	状态
False	0	未选定
True	1	选定

2. 方法

SetFocus 方法是单选按钮控件最常用的方法，可以在代码中通过该方法将焦点定位于某单选按钮控件，从而使其 Value 属性设置为 True。与命令按钮控件相同，使用该方法之前，必须要保证单选按钮控件当前处于可见和可用状态，必须把 Visible 与 Enabled 属性值均设为 True。

3. 事件

单选按钮控件最基本的事件是 Click 事件，用户无须为单选按钮控件编写 Click 事件过程，因为当用户单击单选按钮控件时，它会自动改变 Value 属性值。

例 5.3　设计一个字体设置程序，界面如图 5-9 所示。要求：程序运行后，单击“宋体”或“黑体”单选按钮，可将所选字体应用于标签，单击“结束”按钮则结束程序。在“属性”窗口中按如表 5-6 所示设置各对象的属性。

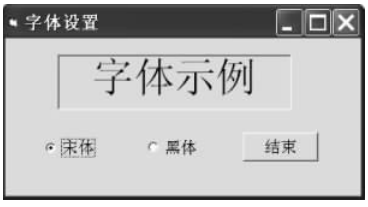

图 5-9　字体设置

表 5-6　各对象的主要属性设置

对　象	属性(属性值)			
窗体	Name(Form1)	Caption(“字体设置”)		
标签	Name(lblDisp)	Caption(“字体示例”)	Alignment(2)	BorderStyle(1)
单选按钮 1	Name(optSong)	Caption(“宋体”)		
单选按钮 2	Name(optHei)	Caption(“黑体”)		
命令按钮	Name(cmdEnd)	Caption(“结束”)		

程序代码如下。

```
Private Sub optSong_Click()              '设置宋体
    lblDisp.FontName = "宋体"
End Sub
Private Sub OptHei_Click()               '设置黑体
    lblDisp.FontName = "黑体"
End Sub
Private Sub cmdEnd_Click()               '结束
   End
End Sub
```

说明：程序在运行后，“宋体”单选按钮自动处于选中状态。

5.2.2 复选框

复选框(CheckBox)也称作检查框。一组复选框控件可以提供多个选项，它们彼此独立工作，用户可以选择其中的一个或多个，也可以一个不选。所以用户可以同时选择任意多个选项，实现一种“不定项选择”的功能。选择某一选项后，该控件将显示“√”，而清除此选项后，“√”消失。复选框在工具箱中的图标为 ☑ 。

有三种选择复选框的方法：鼠标，键盘，程序代码。

1. 属性

Caption 属性：设置标题内容。

Value 属性：是复选框控件最重要的属性，但与单选按钮不同，该控件的 Value 属性为数值型数据，可取三种值：0 为未选中(默认值)，1 为选中，2 为变灰，如表 5-7 所示。同样，用户可在设计阶段通过“属性”窗口或通过程序代码设置该属性值，也可在运行阶段通过鼠标单击来改变该属性值。

表 5-7 Value 属性值

设 置 值	值	状 态
Unchecked	0	未选定
Checked	1	选定
Grayed	2	禁止使用

说明：

(1) 复选框的 Value 属性值为 2 并不意味着用户无法选择该控件，用户依然可以通过鼠标单击或设置焦点的方法，即用 SetFocus 方法将焦点定位在复选框上。

(2) 若要禁止用户选择，必须将其 Enabled 属性设置为 False。

2. 事件

复选框控件最基本的事件也是 Click 事件。同样，用户无须为复选框编写 Click 事件过程，但其对 Value 属性值的改变遵循以下规则。

单击未选中的复选框时，复选框变为选中状态，Value 属性值变为 1；

单击已选中的复选框时，复选框变为未选中状态，Value 属性值变为 0；

单击变灰的复选框时，复选框变为未选中状态，Value 属性值变为 0。

运行时反复单击同一复选框，其只在选中与未选中状态之间进行切换，即 Value 属性值只能在 0 和 1 之间交替变换。

3. 常用事件

Click：二者均支持本事件。

Dbclick：单选按钮支持本事件，可双击 OptionButton。

例 5.4　创建一个应用程序，通过对文本控制的选择，改变标签中文本“VB 程序设计教程”的表现形式。界面如图 5-10 所示。要求：程序运行后，单击各复选框，可将所选字型应用于标签，单击“结束”按钮则结束程序。属性设置如表 5-8 所示。

图 5-10　字形设置

分析：字体样式选择是在 4 种可选字体类型中任选一种，可从选项中任意选择一个、两个或不选，符合复选框控件的使用条件。

表 5-8　各对象的主要属性设置

对　象	属性(属性值)			
窗体	Name(Form1)	Caption(“字形设置”)		
标签	Name(IblDisp)	Caption(“字体示例”)	Alignment(2)	BorderStyle(1)
复选框 1	Name(chkBold)	Caption(“加粗”)		
复选框 2	Name(chkItalic)	Caption(“倾斜”)		
复选框 3	Name(chkUline)	Caption(“下画线”)		
复选框 4	Name(chkSth)	Caption(“删除线”)		
命令按钮	Name(cmdEnd)	Caption(“结束”)		

程序代码如下。

```
Private Sub chkBold_Click()              '设置加粗
    If chkBold.Value = 1 Then
      IblDisp.FontBold = True
    Else
      IblDisp.FontBold = False
    End If
End Sub
Private Sub chkItalic_Click()            '设置倾斜
    If chkItalic.Value = 1 Then
      IblDisp.FontItalic = True
    E1Se
      IblDisp.FontItalic = False
    End If
End Sub
Private Sub chkUline_Click()             '设置下画线
    If chkUline.Value = 1 Then
```

```
        lblDisp.FontUnderline = True
    E1Se
        lblDisp.FontUnderline = False
    End If
End Sub
Private Sub chkSth_Click()                    '设置删除线
    If chkSth.Value = 1 Then
        lblDisp.FontStrikethru = True
    E1Se
        lblDisp.FontStrikethru = False
    End If
End Sub
```

5.3 框架控件

同窗体一样，框架(Frame)控件也是一种“容器”，主要用于为其他控件分组，并将它们分成可标识的控件组。框架中的对象将随着框架移动，而其中对象的位置也是相对于框架的。框架在工具箱中的图标为 。

Frame 控件的常用属性有：Name，Caption，Enabled(指定 Frame 控件是否可用)。

框架除 Caption 属性外，一般情况下很少使用其他的属性。

在框架上创建控件的步骤如下。

(1) 先建框架，后建其中的控件。

(2) 在框架控件上创建其他控件。首先单击工具箱中的工具图标，光标在框架上变成小十字，然后在框架上拉出一个矩形，即可在框架上创建一个控件。

利用现有的控件将它们分组：选中要分组的控件，将它们剪切到剪贴板上选定框架控件，将剪贴板上的控件粘贴到 Frame 控件上。

图 5-11 框架、复选框和单选按钮应用示例

例 5.5 使用框架、复选框和单选按钮显示字体应用示例。程序运行后，分别单击字体、字型，就会使标签中的文字按规定的效果显示出来，显示结果如图 5-11 所示。

1) 界面组成

在窗体上，首先创建“字体”和“字型”两个框架。框架建好后，在“字体”框架上放置单选按钮分别表示“宋体”“黑体”“楷体”，在“字型”框架上放复选框控件，分别表示粗体、斜体、下画线功能。各个控件名称(Name)属性使用默认值。

2) 程序代码

```
Private Sub Option1_Click()                   '单选按钮的操作
  Label1.FontName = "宋体"
End Sub
Private Sub Option2_Click()                   '单选按钮的操作
  Label1.FontName = "黑体"
```

```
End Sub
Private Sub Option3_Click()              '单选按钮的操作
  Label1.FontName = "楷体_GB2312"
End Sub
Private Sub Check1_Click()               '复选框按钮的操作
  If Check1.Value = 1 Then
    Label1.FontBold = True
  Else
    Label1.FontBold = False
  End If
End Sub
Private Sub Check2_Click()               '复选框按钮的操作
  If Check2.Value = 1 Then
    Label1.FontItalic = True
  Else
Label1.FontItalic = False
  End If
End Sub
Private Sub Check3_Click()               '复选框按钮的操作
  If Check3.Value = 1 Then
    Label1.FontUnderline = True
  Else
    Label1.FontUnderline = False
  End If
End Sub
```

注意：

(1) 在窗体上创建框架及其内部部件时，应先添加框架，然后再添加其他的控件。

(2) 在复选框代码操作中，设置及取消操作的方式为：

```
[Object.]FontFold = True/False          '为 True,设置为粗体; 为 False,取消粗体设置
[Object.]FontItalic = True/False        '为 True,设置为斜体; 为 False,取消斜体设置
[Object.]FontUnderline = True/False     '为 True,设置为下画线; 为 False,取消下画线设置
```

(3) 在代码中设置字体的格式为：

```
[Object.]FontName = "字体名"
```

5.4　时钟控件

时钟控件(Timer)又称计时器、定时器控件，用于有规律地定时执行指定的工作，适合编写不需要与用户进行交互就可直接执行的代码，如倒计时、动画等。时钟控件在工具箱中的图标为  ，在程序运行阶段，时钟控件不可见。

定时器控件是一种按一定时间间隔触发事件的控件，其作用是使应用程序按照一定的时间间隔执行某些操作。

1. 常用属性

Interval 属性：时间间隔属性。

Enabled 属性：有效性属性。

1）Interval 属性

取值范围为 0～64 767(包括这两个数值)，单位为 ms(0.001s)，表示计时间隔。若将 Interval 属性设置为 0 或负数，则时钟控件停止工作。

说明：时钟控件的时间间隔并不精确，当 Interval 属性值设置过小时，将可能影响系统的性能。其中，Interval 属性值设置为 1000，表示 1s。

2）Enabled 属性

无论何时，只要时钟控件的 Enabled 属性被设置为 True 而且 Interval 属性值大于 0，则时钟控件开始工作，以 Interval 属性值为间隔，触发 Timer 事件。

通过把 Enabled 属性设置为 False 可使时钟控件无效，即时钟控件停止工作。

2. 方法

Visual Basic 没有为时钟控件提供操作的方法，即没有方法可使用。

3. 事件

定时器只有一个 Timer 事件，用来完成应用程序需要定时完成的任务。时钟控件只能响应 Timer 事件，当 Enabled 属性值为 True 且 Interval 属性值大于 0 时，该事件以 Interval 属性指定的时间间隔发生，需要定时执行的操作即放在该事件过程中完成。

例 5.6 设计一个倒计时程序，界面设计如图 5-12 所示，各个控件及属性按如表 5-9 所示设置。程序运行结果如图 5-13 所示。

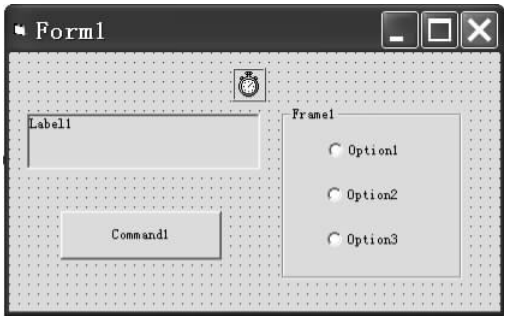

图 5-12 倒计时应用界面

图 5-13 程序运行结果

程序设计要求如下。

（1）程序运行后，通过单选按钮选择计时时间（默认为 1min），单击“计时开始”按钮进行倒计时。

（2）在标签中显示计时情况，计时结束后在标签中显示“时间到”。

（3）单选按钮和“计时开始”按钮在计时开始后被禁用，直到计时结束后才可以使用。

表 5-9　各对象的主要属性设置

对　象	属性（属性值）			
窗体	Name（Form1）	Caption（“倒计时”）	BorderStyle（1）	
框架	Name（Frame1）	Caption（“选择时间”）		
单选按钮 1	Name（optOne）	Caption（“1 分钟”）	Value（True）	
单选按钮 2	Name（optFive）	Caption（“5 分钟”）		
单选按钮 3	Name（optTen）	Caption（“10 分钟”）		
标签	Name（lblTime）	Caption（“1 分 0 秒”）	BorderStyle（1）	Alignment（2）
命令按钮	Name（cmdStart）	Caption（“计时开始”）		
时钟	Name（Timer1）	Interval（1000）	Enabled（False）	

程序代码如下。

```
'声明窗体级变量 pretime,mm,ss 用于存放余下时间的总秒数,分钟数及除去整分钟后的秒数
Dim pretime As Integer, mm As Integer, ss As Integer
Private Sub cmdStart_Click()              '命令按钮,开始倒计时
    cmdStart.Enabled = False
    Framel.Enabled = False                '禁用框架中的所有单选按钮
    mm = pretime \ 60
    ss = pretime Mod 60
    lb1Time.Caption = Str(mm) & "分" & Str(ss) & "秒"
    Timer1.Enabled = True
End Sub
Private Sub optOne_Click()                '1 分钟单选框
    Pretime = 60                          '60 秒为 1 分钟
End Sub
Private Sub optFive_Click()
    Pretime = 300                         '300 秒为 5 分钟
End Sub
Private Sub optTen_Click()
    Pretime = 600                         '600 秒为 10 分钟
End Sub
Private Sub Timer1_Timer()                '计时控件事件的启动,每 1 秒启动一次
    Pretime = Pretime - 1                 '减少 1 秒
    mm = pretime \ 60                     '计算剩余的分钟
    ss = pretime Mod 60                   '除去整分后的秒数
   lb1Time.Caption = Str(mm) & "分" & Str(ss) & "秒"
   If mm = 0 And ss = 0 Then
     lblTime.Caption = "时间到!"
```

```
        Timer1.Enabled = False
        Frame1.Enabled = True
        cmdStart.Enabled = True
    End If
End Sub
```

说明：由于 Interval 设置为 1000，计时器事件启动为 1s，如果设为 2000，那么，计时器事件启动就为 2s。

5.5 图片框与图像框控件

窗体、图形框和图像框可以显示来自图形文件的图形。主要有：

(1) 图像文件，以 *.jpg、*.gif、*.bmp 或 *.dib 为文件扩展名。

(2) 图标(icon)文件：一般在 Windows 中用来表示最小化的应用程序的图标。图标是位图，最大为 32×32 像素，以 *.ico 为文件扩展名。

(3) 元文件(metafile)：将图像作为线、圆或多边形这样的图形对象来存储，而不是存储其像素。元文件的类型有两种，分别是标准型 *.wmf 和增强型 *.emf。在图像的大小改变时，元文件保存图像会比像素更精确。

Visual Basic 工具箱中提供了 4 种与图形有关的控件：图片框(PictureBox)、图像框(Image)、形状控件(Shape)和直线控件(Line)。Line 控件和 Shape 控件是画图工具，十分快捷、有效；而 PictureBox 控件和 Image 控件是加载图形的控件。利用线与形状控件，用户可以迅速地显示简单的线与形状或打印输出，与其他大部分控件不同的是，这两种控件只用来显示或打印，不会响应任何事件。

使用控件绘图适合于窗体内需要较少的直线与圆等情况，其优点是占用系统资源较少，运行速度快，设计阶段可预览图形效果，代码较短。

5.5.1 图片框控件

图片框控件 用于将图片加载到图片框中，可以在“属性”窗口中设置，用来把图形装入这些对象中。在图片框中显示的图形以文件的方式存放在磁盘上，Visual Basic 支持下述格式的图片文件：位图(.bmp)、图标(.ico)、图元文件(.wmf)、增强型图元文件(.emf)、JPEG 文件(.jpg)或 GIF 文件(.gif)。在程序运行时，可用 Picture 属性或用 LoadPicture 函数来加载图片。另外，图片框控件也可作为容器功能来使用，可以在该控件中进行数据的输出，即用 Print 方法输出文本或进行绘图方法的输出，如输出点、线、圆等图形。

图 5-14 窗体及图片框中显示数据界面

例 5.7 在窗体及图片框中显示数据，如图 5-14 所示。程序代码如下。

```
Private Sub Command1_Click()
```

```
    Print "窗体输出 5 * 5"                    '在窗体中显示信息
    Print 5 * 5
                                              '用图片框的方式输出信息
    Picture1.Print "在图片框输出 5 + 5"       '在图片框中显示信息
    Picture1.Print 5 + 5
End Sub
```

1. 向图片框中加载图片

在图片框中显示的图片是由 Picture 属性决定的，有以下两种方法向图片框中加载图形。

1）在设计时加载

从控件的“属性”窗口中选择 Picture 属性。单击右边的“…”按钮，就会出现打开文件对话框，找到需要显示的图像文件即可。

在设计时设置的 Picture 属性，当保存窗体时，系统将自动生成一个与窗体文件同名且后缀为.frx 的二进制文件，图片数据就保存在该文件中。如果将应用程序编译成一个可执行文件，图像将保存在 EXE 文件中，因此可以在没有原始图片文件的任何计算机上运行。

2）在运行时显示或替换图片

可使用 LoadPicture 函数加载图片，以设置 Picture 属性，使用的语句格式为：

```
[object].Picture = LoadPicture([FileName])
```

其中，参数 FileName 为包含全路径名或有效路径名的图片文件名。若省略 FileName 参数，该语句则用来清除图片框中的图像。

（1）在运行时向窗体 Form1 中加载一幅图片：

```
Private Sub form_load()
    Form1.Picture = LoadPicture("C:\windows\flower.jpg")
End Sub
```

（2）在运行时在窗体 Form1 中清除图片：

```
Form1.Picture = LoadPicture()
```

在未指定文件名时，LoadPicture 函数将清除控件对象中的所有图片。

说明：在运行时加载图片，应用程序必须能够访问该图形文件才能显示图像。如果将应用程序编译成一个可执行文件，图像将不会保存在 EXE 文件中，要成功显示图像，运行时必须在宿主计算机上有该图形文件可用。

2. 保存图片

SavePicture 方法主要用于从窗体对象或图片框控件的 Picture 属性或图像控件中将图形保存到图形文件中。无论在设计时还是运行时加载到窗体或图片框中的文件，只要它们是位图、图标、元文件或增强元文件，则保存图形时将使用与原始文件同样的格式。如果是 GIF 或 JPEG 文件，则将保存为位图文件。而图像控件（ImageBox）中的图形总是以位图的格式保存而不管其原始格式。

使用 SavePicture 语句，可将对象或控件的 Picture 或 Image 属性保存为图形文件，其使

用格式为：

```
SavePicture[Object.]Picture | Image,FileName
```

其中，Object 为对象表达式，可以是窗体、图片框、影像框及有 Picture 或 Image 属性的对象。Picture 与 Image 是指对象的 Picture 或 Image 属性。

注意：

（1）对于使用绘图方法和 Print 方法输出在窗体或图片框中的图形和文字，则只能使用 Image 属性保存，而在设计时或在运行时通过 LoadPicture 函数给 Picture 属性加载的图片，则既可使用 Image 属性，也可使用 Picture 属性来保存。若用 Image 属性保存，既包括使用绘图方法和 Print 方法输出在窗体或图片框中的图形和文字，也包含在设计时或在运行时通过 LoadPicture 函数给 Picture 属性加载的图片。

（2）FileName 为必选参数，指定将图形保存的文件名，一般包含盘符、路径及文件名。

（3）无论在设计时还是运行时从文件加载到对象 Picture 属性的位图、图标、元文件或增强元文件，图形都将以原始文件的格式保存。

（4）Image 属性中的图形总是以位图的格式保存，而不管其原始格式。

利用 SavePicture 方法保存图形的代码如下。

```
Private Sub Command4_Click()
  SavePicture Picture2.Image, "c:\Image\small.bmp"
  SavePicture Picture3.Image, "c:\Image\smallhreserse.bmp"
  SavePicture Picture4.Image, "c:\Image\smallvreserse.bmp"
End Sub
```

该事件过程是将处理过的图片框中的图形存储到文件中。

3. 图片框的两个特有属性

1）AutoSize 属性

如果想让图片框能自动扩展到可容纳新图片的大小，可将该图片框的 AutoSize 属性设置为 True。在运行时当向图片框加载或复制图片时，Visual Basic 会自动扩展该控件到恰好能够显示整个图片。由于窗体不会改变大小，如果加载的图像大于窗体的边距，图像从右边和底部被裁剪后才被显示出来。也可以使用 AutoSize 属性使图片框自动收缩，以便对新图片的尺寸做出反应。

说明：窗体没有 AutoSize 属性，并且也不能自动扩大以显示整个图片。

2）Align 属性

该属性值用来决定图片框出现在窗体上的位置，即决定它的 Height、Width、Left 和 Top 属性的取值；Align 属性的取值及含义如表 5-10 所示。

表 5-10　Align 属性的取值及含义

内 部 常 数	数值	含　义
VbAlignNone	0	（非 MDI 窗体的默认值），可以在设计时或在程序中确定大小和位置。如果对象在 MDI 窗体上，则忽略该设置值

续表

内部常数	数值	含　义
VbAlignTop	1	(MDI窗体的默认值),显示在窗体的顶部,其宽度自动等于窗体的ScaleWidth属性设置值
VbAlignBottom	2	显示在窗体的底部,其宽度自动等于窗体的ScaleWidth属性设置值
VbAlignLeft	3	显示在窗体的左面,其高度自动等于窗体的ScaleHeight属性设置值
VbAlignRight	4	显示在窗体的右面,其高度自动为窗体的ScaleHeight属性设置值

说明：当Align属性的值为1或2时,设置图片框和窗体的顶部或底部对齐,该图片框的宽度等于窗体内部的宽度,会自动地改变大小以适合窗体的宽度。当Align属性的值为3或4时,设置图片框和窗体的左边或右边对齐,该图片框的高度等于窗体内部的高度。

通常利用这个特点来建立一个图片框,用于创建窗体顶端的工具条和位于底部的状态栏,这就是手工创建工具栏或状态栏的方法。

5.5.2　图像框控件

图像框控件也是用来显示图片的,与PictureBox控件相似,但它只用于显示图片,而不能作为其他控件的容器,即不支持绘图方法和Print方法。所以,图像框的功能少一些,但是,该图像框在显示图片时比图片框占用更少内存,同时对图片有拉伸的功能。

1. 向图像框中加载图片

Image控件加载图片的方法和PictureBox中的方法一样。

1) 在设计时加载

从控件的"属性"窗口中选择Picture属性。单击右边的"…"按钮,就会出现打开文件对话框,找到需要显示的图像文件即可。

2) 在运行时显示或替换图片

可使用LoadPicture函数设置Picture属性,使用的语句格式为:

```
[object].Picture = LoadPicture([FileName])
```

所以在设计时,将Picture属性设置为文件名,运行时,可利用LoadPicture函数为图像框加载图片文件,格式与图片框相同。

2. 图像框的Stretch属性

Image控件调整大小的行为与PictureBox不同,它具有Stretch(拉伸)属性,该值用来指定一个图形是否要调整大小,以适应Image控件的大小。

(1) 当Stretch属性设为False(默认值)时,Image控件可根据图片来调整自己的大小。

(2) 当Stretch属性设为True时,Image控件的大小不变,图片根据Image控件来调整图片的大小,这可能使图片变形。

说明：使用Image控件可创建自己的按钮。因为Image控件也可以识别Click事件,因此,可在需要用CommandButton的任何地方使用该控件。

例5.8　在窗体上放置两个Image控件Image1和Image2,其中,Image1图像框中的图

片是自动拉伸，即图片的大小自动调整到图像框的大小，Image2 图像框中的图片是按原尺寸的大小在图像框中显示，在窗体的 Load 事件中编写如下代码。

```
Private Sub Form_Load()
   Image1.Stretch = True          'Stretch 属性为 True,使 Image1 图像框中的图片具备自动拉伸
   '加载图片 Wallpaper_1.bmp,在 E 盘图例子目录下
   Image1.Picture = LoadPicture("E:\图例\ Wallpaper_1.bmp")
   Image2.Stretch = False         '将 Stretch 属性设为 False,Image2 图像框的图片按原尺寸显示
   Image2.Picture = LoadPicture("E:\图例\ Wallpaper_1.bmp ")
End Sub
```

程序运行结果如图 5-15 所示。

图 5-15　Stretch 属性值不同的图片显示结果

例 5.9　模拟升降旗示例。

1）界面组成及实现方式

界面组成如图 5-16 所示，在窗体上放置一个图像框(Image1)、一个直线控件(Line1)，DrawWidth 属性设为 4，两个命令按钮（Command1、Command2），一个计时器(Timer1)，Timer1 的 Interval 属性值设为 10，如果设为 5，速度加快一倍。

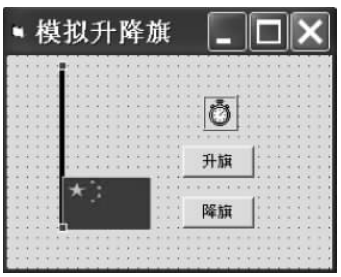

图 5-16　模拟升降旗示例

2）图片装入

在图像框(Image1)中的 Picture 属性中加入一个国旗的图像文件。

3）程序代码

```
Dim f As Integer
Private Sub Command1_Click()
   Timer1.Enabled = True                    '启动定时器,执行升旗操作
   Command1.Enabled = False                 '禁止"升旗"按钮响应功能
End Sub
Private Sub Command2_Click()
   Timer1.Enabled = True                    '启动定时器,执行降旗操作
   Command2.Enabled = False                 '禁止"降旗"按钮响应功能
End Sub
```

```
Private Sub Form_Activate()
   f = 1                                    '升旗或降旗标识,1 为升旗,0 为降旗
   Command1.Enabled = True                  '允许升旗
   Command2.Enabled = False                 '禁止降旗
End Sub

Private Sub Timer1_Timer()
   If f = 1 Then
      If Image1.Top > Line1.Y1 Then         '判断是否升到顶,若否,上升旗帜
         Image1.Top = Image1.Top - 10
      Else
         Timer1.Enabled = False             '若是,即到顶后,关闭定时器
         Command2.Enabled = True            '开启"降旗"按钮的响应功能
         f = 0                              '允许降旗
      End If
   Else
      If Image1.Top + Image1.Height < Line1.Y2 Then          '判断是否降到底,若否,下降旗帜
         Image1.Top = Image1.Top + 10
      Else
         Timer1.Enabled = False             '若是,即到底,关闭定时器
         Command1.Enabled = True            '开启"升旗"按钮的响应功能
         f = 1                              '允许升旗
      End If
   End If
End Sub
```

例 5.10　用图像框、滚动条、计时器组成一个让图形闪烁的示例。单击“图片闪烁”按钮,图片开始闪动,单击“停止闪烁”按钮图片停止闪烁,用滚动条控制闪烁的速度。

1) 界面组成及实现方式

界面组成如图 5-17 所示,在窗体上放置一个图像框(Image1)、一个滚动条(HScroll1)、两个命令按钮(Command1、Command2),一个计时器(Timer1)。

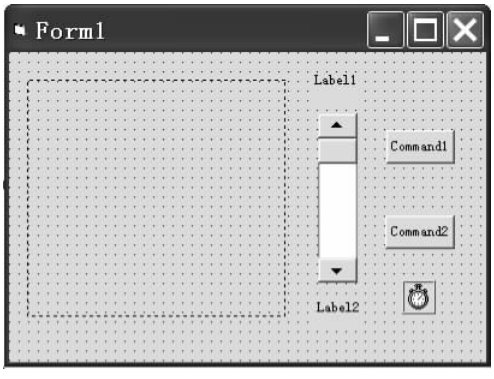

图 5-17　图形闪烁示例

2) 图像操作方式

图像闪烁是使图像控件从可见到不可见,再从不可见到可见,反复进行。图片操作结果如图 5-18 所示,操作方式如下。

图 5-18　图形闪烁结果

（1）闪烁的实现用以下语句来进行：

```
<Object.>Visible = Not <Object.>Visible
```

（2）速度的控制用滚动条 HScroll1 的 Value 属性值给计时器 Timer1 的 Interval 属性赋值来实现。

3）程序代码

```
Private Sub Form_Load()
    Image1.Stretch = True
    Image1.Picture = LoadPicture("E:\图例\3 - 12.bmp")
    VScroll1.Max = 2000
    VScroll1.Min = 1
    VScroll1.LargeChange = 100
    VScroll1.SmallChange = 15
End Sub
Private Sub Command1_Click()
  Timer1.Enabled = True                    '使计时器 Timer1 有效,开始计时
End Sub
Private Sub Command2_Click()
  Timer1.Enabled = False                   '使计时器 Timer1 无效,停止计时
End Sub
  Private Sub Timer1_Timer()
  Image1.Visible = Not Image1.Visible      '实现闪烁
End Sub
Private Sub VScroll1_Change()
  Timer1.Interval = VScroll1.Value         '控制速度
End Sub
```

5.6　坐标系统及图形颜色

5.6.1　坐标系统

在 Visual Basic 中，每个容器对象（屏幕、窗体或图片框等）都有一个坐标系。对象的坐

标系统是绘制各种图形的基础，坐标系统选择的恰当与否直接影响着绘图的质量。同样的绘图命令，可能仅仅由于用户定义或选择的坐标系统不同，而不能正确地在屏幕上显示或在打印机上打印出结果来，或者即使能显示或打印出来，也可能会比例不协调，达不到预期效果。因此，在绘制图形前，必须首先确定坐标系。构成一个坐标系需要三个要素：坐标原点，坐标度量单位，坐标轴的长度与方向。

Visual Basic 在窗体对象中，有一套默认的坐标系统，其坐标原点(0,0)总是在其左上角，X 轴的正向水平向右，Y 轴的正向垂直向下，默认坐标的刻度单位是缇(twip)。

Visual Basic 中的坐标系统，在各种容器对象中都可以应用。所谓容器对象，就是可以放置其他对象的对象。Visual Basic 中能作容器的对象除窗体外，还有图片框、框架控件。系统对象即屏幕(Screen)，也是一个容器。窗体就是放置在屏幕中的。此外，系统容器还有打印机(Printer)。在每个容器对象中，移动控件或调整控件的大小时，使用控件容器的坐标系统，所有的绘图方法和 Print 方法，也使用容器的坐标系统。

在使用坐标系统中，那些在窗体上绘制的控件，使用的是窗体的坐标系统，而在图片框里绘制控件，使用的是图片框的坐标系统。窗体是放在屏幕中的对象，因此在编写用来调整窗体大小或移动窗体位置的代码时，则要使用到屏幕坐标系统，应先检查屏幕对象 Screen 的 Height 属性和 Width 属性，以确保窗体在屏幕上大小合适。

5.6.2　标准坐标系

显示器是以像素(分辨率)为度量单位的。常见显示器的分辨率为 640×480、800×600、1024×768 等。同样的一幅图形，由于使用的显示器分辨率不同，所显示的效果就不同。因此在传统的图形设计中，常根据显示器分辨率来确定绘制图形的大小。

默认状态下，标准坐标系统和常用的笛卡儿坐标系不同，容器对象左上角的点是原点，坐标是(0,0)，X 轴正向水平向右，Y 轴的正向垂直向下，如图 5-19 所示。

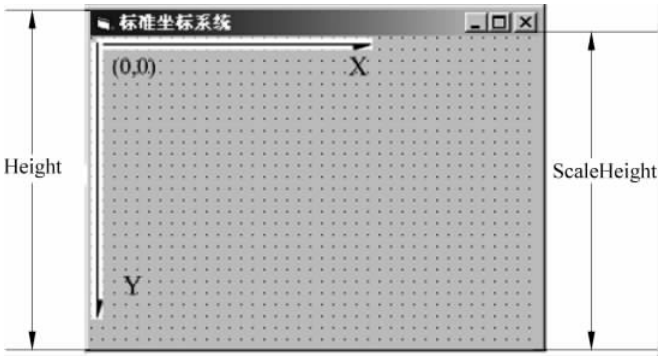

图 5-19　标准坐标系统

容器对象标准坐标系的默认标度是 twip(缇)，可由容器的 ScaleMode 属性指定标度。

对象的 ScaleTop 属性和 ScaleLeft 属性设置容器内部左边和顶端的坐标，这两个属性的默认值都为 0。根据对象的 ScaleTop 和 ScaleLeft 属性可确定坐标原点。对象的 ScaleWidth 和 ScaleHeight 属性定义对象内部水平和垂直方向的单元数。

用户可通过改变坐标系标度或改变坐标系原点来自定义坐标系。只要采用了自定义坐标系，ScaleMode 属性由 Visual Basic 设为 vbUser。要返回标准坐标系，可使用无参数的 Scale 方法：[容器对象].Scale。

5.6.3 自定义坐标系

开发图形程序时，标准坐标系可能使用起来并不一定方便。为达到最大的编程灵活性，坐标系可以根据需求重新定义。如数学中常用的笛卡儿坐标系，可以重新定义坐标系标度、原点和坐标轴的方向来改变坐标系。

1. 坐标系的标度

坐标系的标度是指坐标系的度量单位。Visual Basic 提供了 8 种标度，如表 5-11 所示。坐标系统可以使用以下三种不同的标度：默认标度，标准标度或者自定义标度。用户可用 ScaleMode 属性设置坐标系统的标度单位。ScaleMode 属性的取值及含义如表 5-11 所示。

表 5-11 Visual Basic 坐标系标度

定标单位	常量	值	说明
用户定义	vbUser	0	由程序员定义坐标单位
twip(缇)	vbTwips	1	默认单位，1 英寸＝1440 缇，1 厘米＝567 缇
point(磅)	vbPoints	2	通常用于字体。1 英寸＝72 磅
pixel(像素)	vbPixels	3	通常用于图像。表示屏幕分辨率的最小单位
character(字符)	vbCharacters	4	水平 120 缇，垂直 240 缇
inch(英寸)	vbInches	5	英寸
millimeter(毫米)	vbMillimeters	6	毫米
centimeter(厘米)	vbCentimeters	7	厘米

说明：用 ScaleMode 属性只能改变标度单位，不改变坐标原点及坐标轴的方向。当设置容器对象的 ScaleMode 属性值大于 0 时，将使容器对象的 ScaleLeft 属性和 ScaleTop 属性自动设置为 0，ScaleHeight 属性和 ScaleWidth 属性的度量单位也将发生改变。

标度(ScaleMode 属性)可以在设计阶段设置，也可以在运行阶段改变。标度单位转换可使用 ScaleX 和 ScaleY 方法，其语法为：

```
[对象名.]ScaleX(转换值,原坐标单位,目标坐标单位)
[对象名.]ScaleY(转换值,原坐标单位,目标坐标单位)
```

一般情况下，坐标系都是使用系统的默认设置，即坐标系统以 twip(缇)为单位。若要返回默认标度，可使用：[容器对象].ScaleMode。

2. 使用 Scale 属性建立自己的坐标系

容器对象的 Scale 属性共有 4 个：ScaleLeft、ScaleTop、ScaleWidth 和 ScaleHeight。可使用这 4 个 Scale 属性来创建用户自定义坐标系统及刻度单位，其含义如表 5-12 所示。

表 5-12　Scale 属性

属　性	含　义
ScaleLeft	确定对象左边的水平坐标
ScaleTop	确定对象顶端的垂直坐标
ScaleWidth	确定对象内部水平的宽度，不包括边框
ScaleHeight	确定对象内部垂直的高度，不包括边框标题(对窗体)和边框

其中：

(1) ScaleTop 属性和 ScaleLeft 属性的值用于控制对象左上角坐标，所有对象的 ScaleTop 属性、ScaleLeft 属性的默认值为 0，坐标原点在对象的左上角。ScaleTop＝N，表示将 X 轴沿 Y 轴的负方向平移 N 个单位；ScaleTop＝－N，表示 X 轴沿 Y 轴的正方向平移 N 个单位；同样，设置 ScaleLeft 属性的值可向左或向右平移坐标系的 Y 轴。

(2) ScaleWidth 属性和 ScaleHeight 属性的值可确定对象坐标系 X 轴与 Y 轴的正向及最大坐标值。如果 ScaleWidth 属性的值小于 0，则 X 轴的正向向左；如果 ScaleHeight 属性的值小于 0，则 Y 轴的正向向上。默认时其值均大于 0。

容器对象右下角坐标值为(ScaleLeft＋ScaleWidth，ScaleTop＋ScaleHeight)。

3. 使用 Scale 方法设置坐标系

除直接设置相关属性外，也可采用更简单的 Scale 方法自定义坐标系统。

Scale 方法是建立用户坐标系最方便的方法，其使用格式如下：

```
[Object.]Scale [(x1,y1) - (x2,y2)]
```

其中：

(1) Object：是可选的一个对象表达式，如果省略 Object，则指带有焦点的 Form 对象。

(2) x1、y1：是可选的，均为单精度数值，指示定义对象左上角的水平(x 轴)和垂直(y 轴)坐标。这些数值必须用括号括起来。x1、y1 就是 ScaleLeft、ScaleTop。

(3) x2、y2：是可选的，均为单精度数值，指示定义对象右下角的水平和垂直坐标。这些数值必须用括号括起来。x2-x1、y2-y1 就是 ScaleWidth、ScaleHeight。

Scale 方法能够将坐标系统重置到所选择的任意刻度。Scale 对运行时的图形语句以及控件位置的坐标系统都有影响。如果使用不带参数的 Scale(两组坐标都省略)，对象的坐标系统将重置为默认坐标系统。

比如语句 Scale(－500，250)－(500，－250)将窗体左上角坐标设为(－500，250)，右下角坐标设为(500，－250)，将窗体的坐标系统的原点定义在其中心，X 轴的正向向右，Y 轴的正向向上，窗体高与宽分别为 1000 和 500 单位长度。改变后的坐标系如图 5-20 所示。

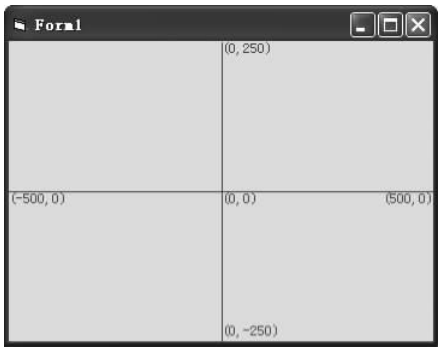

图 5-20　用 Scale 方法自定义坐标系统

5.6.4 图形颜色

在 Visual Basic 系统中，对于字体、背景、控件都有可能使用颜色的信息，所有的颜色属性在 Visual Basic 系统中都由一个 Long 整数表示。颜色的设置有 4 种表示方式。

1. 使用 RGB 函数

(1) 使用 QBColor 函数，选择 16 种 QBASIC 颜色中的一种。
(2) 使用系统提供的颜色常数。
(3) 直接使用 Long 型颜色值。

RGB 函数是 Visual Basic 系统中的一个内部函数，也是目前大多数软件表示颜色的函数，它是用红(red)、绿(green)、蓝(blue)三个颜色值来表示任何颜色，每一个颜色值的表示为 0～255，即红为 0～255、绿为 0～255、蓝为 0～255，总共表示的颜色值为：256×256×256＝16 777 216。RGB 函数使用格式为：

```
RGB(red,green,blue)
```

说明：可以用 RGB 函数来指定任何颜色，因为每一种可视的颜色，都可由红、绿、蓝三种主要颜色组合产生。为了用 RGB 函数指定颜色，要对三种主要颜色中的每种颜色，赋予 0～255 的一个亮度值(0 表示亮度最低，255 表示亮度最高)，将结果赋给颜色属性或颜色参数。

用 RGB 函数表示颜色的方法如下。
(1) 表示红色：RGB(255,0,0)。
(2) 表示绿色：RGB(0,255,0)。
(3) 表示蓝色：RGB(0,0,255)。
(4) 表示任意颜色：RGB(25,123,6)。
在窗体控件中设置背景色为红色的表示：

```
Form1.BackColor = RGB(255,0,0)
```

2. 使用 QBColor 函数

QBColor 函数是 Visual Basic 系统中，用来表示所对应颜色值的 RGB 颜色码，即固定值，其使用格式为：

```
QBColor(color)
```

其中，color 参数是一个界于 0～15 的整型数，分别代表 16 种颜色，如表 5-13 所示。
在窗体控件中设置背景色为红色的表示：

```
Form1.BackColor = QBColor(12)
```

3. 使用系统定义的颜色常数

在 Visual Basic 系统中，系统已经预先定义了常用颜色的颜色常数，如常数 vbRed 就代表红色，vbGreen 代表绿色等。表 5-14 是系统预定义的最常用的颜色常数。

表 5-13　QBColor 函数的 color 参数

参　数　值	颜　　色	参　数　值	颜　　色
0	黑色	8	灰色
1	蓝色	9	亮蓝色
2	绿色	10	亮绿色
3	青色	11	亮青色
4	红色	12	亮红色
5	洋红色	13	亮洋红色
6	黄色	14	亮黄色
7	白色	15	亮白色

表 5-14　常用颜色常数

内 部 常 数	值	颜　　色
VbBlack	&H0	黑色
VbRed	&HFF	红色
VbGreen	&HFF00	绿色
VbYellow	&HFFFF	黄色
VbBlue	&HFF0000	蓝色
VbMagenta	&HFF00FF	洋红
VbCyan	&HFFFF00	青色
VbWhite	&HFFFFFF	白色

把窗体的背景色设为红色的代码为：

```
Form1.BackColor = vbRed
```

4. 直接使用颜色设置值

Visual Basic 可以直接使用数值来指定颜色，给颜色参数和属性指定一个值，通常使用十六进制数。用十六进制数指定颜色的格式为：

```
&BBGGRR
```

其中，BB 指定蓝颜色的值，GG 指定绿颜色的值，RR 指定红颜色的值。每个数段都是两位十六进制数，即 BB 为 00～FF，GG 为 00～FF，RR 为 00～FF。

例如，将窗体背景指定为红色的语句为：

```
Form1.BackColor = &H0000FF
```

它相当于：

```
Form1.BackColor = RGB(255,0,0)
```

例 5.11　简单图片浏览器设计。本例使用驱动器列表框(DriveListBox)控件，目录列表框(DirListBox)控件和文件列表框(FileListBox)控件来制作一个文件浏览器。当单击文件列表框中的图片文件时，在右面的图片框中显示出图片。界面如图 5-21 所示。

图 5-21　简单图片浏览器

```
'图片的放大与缩小
Private Sub Command1_Click(Index As Integer)
  If Index = 0 Then
    With Image1
      If .Top > 10 And .Left > 10 Then
        .Top = .Top - 10
        .Left = .Left - 10
        .Height = .Height + 20
        .Width = .Width + 20
      End If
    End With
  Else
    With Image1
      If .Width > 10 And .Height > 10 Then
        .Top = .Top + 10
        .Left = .Left + 10
        .Height = .Height - 20
        .Width = .Width - 20
      End If
    End With
  End If
End Sub
Private Sub Dir1_Change()
  File1.Path = Dir1.Path
End Sub
  Private Sub Drive1_Change()
Dir1.Path = Drive1.Drive
End Sub
Private Sub File1_Click()
  ChDrive Drive1.Drive
  ChDir Dir1.Path
  Image1.Picture = LoadPicture(File1.FileName)
End Sub
Private Sub Form_Load()
  File1.Pattern = "*.bmp;*.gif;*.jpg"
End Sub
```

5.7　直线、形状控件

5.7.1　直线控件

直线(Line)控件用于在窗体、框架或图片框中画简单的线段，通过属性的变化可以改变直线的粗细、颜色及线型。直线显示时的长度和位置由位置属性(x1，y1)和(x2，y2)确定。(x1，y1)指定起点坐标，(x2，y2)指定终点坐标。直线控件可用来在窗体上显示各种类型和宽度的线条，在工具箱中显示为 ＼ 。对于直线控件来说，程序运行时最重要的属性是线条的长短及线条的类型。

1. 线条的坐标位置

“X1”“Y1”“X2”“Y2”属性用来决定直线显示的位置坐标，“X1”设置(或返回)直线的最左端水平位置坐标，“Y1”设置(或返回)最左端垂直坐标，“X2”“Y2”则表示右端的坐标。

2. 线宽与线型属性

1) DrawWidth 属性

DrawWidth 属性可设置线条(Line)输出的线宽。

使用格式如下：

```
Object.DrawWidth = Size
```

其中，Object 为线条对象表达式；Size 为线条宽度，其值为 1～32 767。该值以像素为单位表示线宽，默认值为 1，即一个像素宽。

2) DrawStyle 属性

DrawStyle 属性可决定线型的输出样式。

使用格式如下：

```
Object. DrawStyle = [number]
```

其中，Object 为线条表达式；Number 为整型表达式，值为 0～6，用来指定线条输出的线型，具体含义如表 5-15 所示。

表 5-15　DrawStyle 属性设置及含义

内 部 常 数	描　　述	内 部 常 数	描　　述
VbSolid	实线	VbDashDotDot	双点画线
VbDash	虚线	VbInvisible	透明线(不可见)
VbDot	点线	VbInsideSolid	内收实线
VbDashDot	点画线		

例 5.12 用不同的 DrawStyle 属性值，在窗体中画直线。

将程序代码写在窗体的单击事件中：

```
Private Sub Form_Click()
  Dim I As Integer
  DrawWidth = 1
  ScaleHeight = 8                              '将窗高设置为 8 个单位
  For i = 0 To 6
     BorderStyle = i                           '改变线型
     Line(0,i + 1) - (ScaleWidth * 2/3,i + 1)  '画新线
     CurrentY = CurrentY - 0.25                '在当前点 Y 坐标向上移 0.25 个刻度单位
     Print "DrawStyle = " ; DrawStyle
  Next I
End Sub
```

5.7.2 形状控件

形状(Shape)控件在工具箱中显示为 。该控件可在窗体、框架或图片框中创建矩形、正方形、椭圆形、圆形、圆角矩形或圆角正方形等图形。形状控件预定义形状是由 Shape 属性的取值决定的。在表 5-16 中列出了所有预定义形状、形状值和相应的 Visual Basic 常数。

表 5-16 Shape 控件的 Shape 属性设置

常　数	Shape 属性值	显 示 效 果
VbShapeRectangle	0	矩形(默认值)
VbShapeSquare	1	正方形
VbShapeOval	2	椭圆形
VbShapeCircle	3	圆形
VbShapeRoundedRectangle	4	圆角矩形
VbShapeRotmdedSquare	5	圆角正方形

其中，在使用形状控件时，用 BackStyle 属性来决定形状的背景是否为透明，默认值为 1，显示一个不透明形状。

利用线与形状控件，用户可以迅速地显示简单的线与形状或将之打印输出，与其他大部分控件不同的是，这两种控件不会响应任何事件，它们只用来显示或打印。

在图形的填充中，封闭图形的填充方式由 FillStyle 决定，填充颜色和线条颜色由 FillColor 属性决定。

1. FillStyle 属性

FillStyle 属性用来设置填充形状控件以及由 Circle 和 Line 图形方法生成的圆和方框的图形，具体取值及含义如表 5-17 所示。

说明：

(1) FillStyle 为 0 是实填充，1 为透明方式。填充图案的颜色由 FillColor 属性来决定。

(2) 对于窗体和图片框对象，FillStyle 属性设置后，并不能看到其填充效果，而只能在使用 Circle 和 Line 图形方法生成的圆和方框时，在圆和方框中显示其填充效果。

表 5-17　FillStyle 属性设置及含义

FillStyle 属性值	含　义	FillStyle 属性值	含　义
0	绘制实心图形	4	左上到右下斜线
1	透明(默认方式)	5	右上到左下斜线
2	水平线	6	网状格线
3	垂直线	7	网状斜线

2. FillColor 属性

用于设置填充形状的颜色,默认情况下,FillColor 设置为 0(黑色)。

例 5.13　模拟时钟。在窗体中加入一个 Shape 控件,三条 Line 控件,一个时钟控件,两个标签。程序运行界面如图 5-22 所示。

```
Dim r As Single, x0 As Integer, y0 As Integer '半径及圆心
Dim dx As Single                              '每 1°对应的弧度
Private Sub Form_Load()
   Form1.ScaleHeight = 2600
   Form1.ScaleWidth = 3000
   dx = 3.1416 / 180
   Timer1.Enabled = True
   Form1.AutoRedraw = True
End Sub

Private Sub Form_Resize()
  Dim i As Integer
  Form1.Cls
  Timer1.Interval = 1000
  Shape1.Height = Form1.ScaleHeight * 3 / 2
  x0 = Form1.ScaleWidth / 2
  y0 = Form1.ScaleHeight / 2
  Shape1.Move x0 - Shape1.Width / 2, y0 - Shape1.Height / 2      '移圆至窗体中心
  Line1.X1 = x0                                                  '移指针一端至窗体中心
  Line2.X1 = x0
  Line3.X1 = x0
  Line1.Y1 = y0
  Line2.Y1 = y0
  Line3.Y1 = y0
  r = Shape1.Width / 2 - 40
  For i = 0 To 330 Step 30                                       '画电子钟的刻度线
     If i Mod 90 = 0 Then
       Form1.DrawWidth = 5
       Form1.ForeColor = vbRed
     Else
       Form1.ForeColor = vbBlue
       Form1.DrawWidth = 3
     End If
      Form1.Line (x0 + r * Sin(i * dx), y0 - r * Cos(i * dx)) - (x0 + r * 0.85 * Sin(i *
dx), y0 - r * 0.85 * Cos(i * dx))
```

图 5-22　模拟时钟

```
    Next i
    Form1.ForeColor = vbBlack
    For i = 0 To 330 Step 30 '写电子钟的时间刻度值
        Form1.CurrentX = x0 + r * 0.7 * Sin(i * dx) - 100
        Form1.CurrentY = y0 - r * 0.7 * Cos(i * dx) - 100
        Form1.Print i \ 30
    Next i
End Sub

Private Sub Timer1_Timer()
    Dim h As Integer, m As Integer, s As Integer          '时、分、秒
    Dim hh As Single, mm As Single, ss As Single          '时、分、秒所对应的角度数
    Label2 = Time
    h = Hour(Time)
    m = Minute(Time)
    s = Second(Time)
    If s = 0 Then Beep
    ss = s * 6                                            '1 秒转过 6°
    mm = (m + s / 60) * 6                                 '1 分钟转过 6°
    hh = (h + m / 60 + s / 3600) * 30                     '1 小时转过 30°
    '确定指针的另一端位置
    Line1.X2 = x0 + r * 0.4 * Sin(hh * dx)
    Line1.Y2 = y0 - r * 0.4 * Cos(hh * dx)
    Line2.X2 = x0 + r * 0.6 * Sin(mm * dx)
    Line2.Y2 = y0 - r * 0.6 * Cos(mm * dx)
    Line3.X2 = x0 + r * 0.8 * Sin(ss * dx)
    Line3.Y2 = y0 - r * 0.8 * Cos(ss * dx)
End Sub
```

5.8 绘图方法

前面介绍的是使用直线、形状等控件直接绘图，由于控件绘图无法重叠，对较为复杂的图形输出，特别是动态图形输出，就需要使用绘图方法绘图。各直线或圆可重叠交叉使用，用控件绘图无法重叠。

Visual Basic 中提供的基本的绘图方法主要包括：Cls 方法、Line 方法、Circle 方法、Point 方法和 PSet 方法。此外，Print 方法也可认为是一种绘制图形方法。

5.8.1 Cls 方法

Cls 方法将以背景色清除绘制的图形以及 Print 方法在运行时所产生的文本或图形。

语法格式：

```
[对象名.]Cls
```

设计时在 Form 中使用 Picture 属性设置的背景图和放置的控件不受 Cls 方法影响。使用 Cls 方法之后，当前坐标复位到原点。

5.8.2　Line 方法

Line 方法用于在窗体或图片框对象上画直线(斜线也是直线)和矩形。

语法格式：

```
[object.]Line[[Step](x1,y1)]-[step](x2,y2)[,color][,B[F]]
```

说明：

(1) Step 表示其后的坐标值使用的是相对偏移。(x1,y1)是直线的起点坐标,若前面有 Step,则表示(x1,y1)是相对于当前位置的偏移量；否则(x1,y1)是相对于原点(0,0)的偏移量。若省略(x1,y1),则起点为当前坐标位置(CurrentX,CurrentY)。

(2) B 表示画矩形；F 表示用画矩形的颜色来填充矩形,F 必须与关键字 B 一起使用。如果只用 B 不用 F,则填充由 FillColor 和 FillStyle 属性决定。color 指定要画直线的颜色,可以使用颜色代码或颜色函数,省略时用对象的 ForeColor 属性指定的颜色绘制直线。

注意：用 Line 方法在窗体上绘制图形时,如将绘制过程放在 Form_Load 事件内,必须将窗体的 AutoRedraw 属性设置为 True,否则所绘制的图形无法在窗体上显示。各参数可根据实际要求进行取舍,如果舍去的是中间参数,则不能舍去参数的位置分隔符。

例如：

(1) Line(250,300)－(400,500)

画一条从(250,300)到(400,500)点的直线。

(2) Line－(400,500)

从当前位置(CurrentX,CurrentY)画直线到(400,500)。

(3) Line(150,250)－Step(150,50)

出发点是(150,250),终点是向 X 轴正向走 150,向 Y 轴正向走 50 的点,与 Line(150,250)－(300,300)等效。

(4) Line(20,40)－(150,200) , , B

画一个左上角在(20,40),右下角在(150,200)的矩形。参数省略时,逗号并不省略。

(5) Line(20,40)－Step (50,70), RGB(255,0,0), BF

用红色从(20,40)到(70,110)画一个实心矩形。

例 5.14　使用 Line 方法在窗体上画直线和矩形。执行结果如图 5-23 所示。

```
Private Sub Form_Resize()
    Const x0 = 10
    Const y0 = 15
    Cls
    Scale (0, 100)-(100, 0)
    Line (x0, y0)-(x0, 90)
    Line (x0, y0)-(90, y0)
    For i = 10 To 70 Step 10
    Form1.FillStyle = i / 10
    Form1.FillColor = QBColor(i / 10 - 1)
    Line (x0 + i, y0 + i)-(x0 + i + 6, y0), , B
    CurrentX = x0 + i
```

```
    CurrentY = y0 + i + 6
    Print i
    Next
End Sub
```

例 5.15 画金刚石图案。

本例用多条直线来画金刚石图案。首先重新定义坐标系，利用圆上的角度来获得正多边形的角点并存储在数组 px 和 py 中，然后用双重循环来实现任意两点之间的对角线互连。程序执行界面如图 5-24 所示。

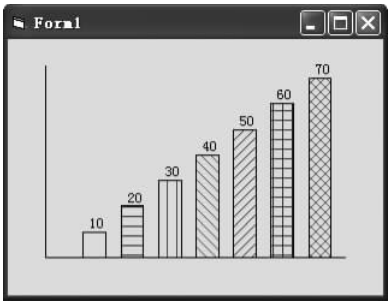

图 5-23 使用 Line 方法在窗体上画直线和矩形

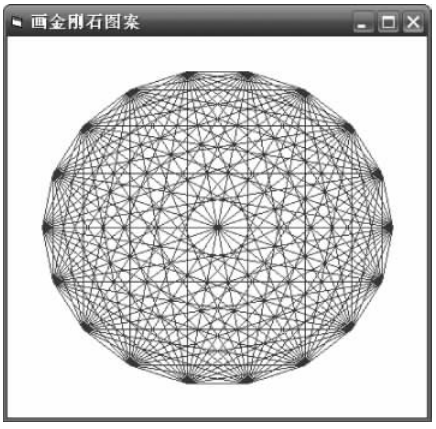

图 5-24 画金刚石图案

```
Option Explicit
Const Pi As Double = 3.1415926
Private Sub Form_Load()
  Randomize
  Form1.BackColor = vbWhite
  Form1.ForeColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255) '采用随机颜色
  Form1.Scale (-60, 60)-(60, -60)                        '重新定义坐标系
End Sub
Private Sub Form_Click()
    Cls
    Dim n, x0, y0, r As Integer
    n = 18                                               '角点个数
    r = 50                                               '取角点的圆的半径
    Dim px(), py() As Double
    ReDim px(n), py(n)
    Dim i, j As Integer
    For i = 1 To n
        px(i) = x0 + r * Cos(i * 2 * Pi / n)
        py(i) = y0 + r * Sin(i * 2 * Pi / n)
    Next                                                 '计算直线的端点坐标并存入数组
    For i = 1 To n
        For j = 1 To i - 1
            Line (px(i), py(i))-(px(j), py(j))
        Next j
```

```
    Next i                                          '连接任意两个端点
End Sub
```

例 5.16　简单鼠标绘图程序。

本程序主要是实现用鼠标在窗体上绘图，如图 5-25 所示，在绘图过程中可选择颜色及线宽。用鼠标在窗体上绘图利用窗体对象的 MouseDown 和 MouseMove 事件实现；利用通用对话框控件 CommonDialog 的 ShowColor 方法可实现前景色和背景色的选取；利用单选框来选择线宽。

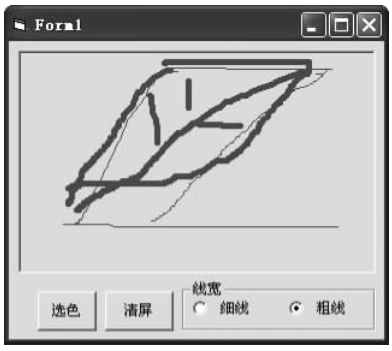

图 5-25　鼠标绘图示例

```
Private Sub Command1_Click()      '选择绘笔颜色
    CommonDialog1.Action = 3
    Picture1.ForeColor = CommonDialog1.Color
End Sub
Private Sub Command2_Click()      '清除
    Picture1.Cls
End Sub
                  '当鼠标左键按下记录下当前坐标
Private Sub Picture1_MouseDown(Button%, Shift%, X As Single, Y As Single)
   Picture1.CurrentX = X
   Picture1.CurrentY = Y
 End Sub
Private Sub Picture1_MouseMove(Button%, Shift%, X As Single, Y As Single)
                  '当鼠标左键按下并移动时画线
    If Option1.Value = True Then
        Picture1.DrawWidth = 1
    End If
    If Option2.Value = True Then
        Picture1.DrawWidth = 5
    End If
    If Button = 1 Then
        Picture1.Line -(X, Y)
    End If
End Sub
```

5.8.3　Circle 方法

Circle 方法用于在指定对象上画圆、椭圆、圆弧和扇形。圆的半径是圆心到圆周的距离；椭圆与圆的不同在于它的纵横比（如宽和高之比）不是 1；扇形是圆的一部分；圆弧是扇形的弯曲部分。前面介绍的有关属性 DrawWidth、DrawStyle、FillColor、FillStyle 等在 Circle 方法中也同样适用。

语法格式：

```
[Object.] Circle [ [Step] (x,y) ,radius [,color] [start,end], aspect]
```

说明：

(1) (x,y)指定圆心的位置，Step 关键字表示相对坐标，radius 参数用于指定圆的半径。

（2）color 参数用于指定绘制圆的颜色。

（3）start 指定弧的起始角，end 指定弧终止角。它们的单位均是弧度，范围为 0～2π。画弧时，start 和 end 都用正值。从 start 开始，逆时针画到 end 处结束。如果画扇形，则 start 和 end 都取负值，也是从 start 开始，逆时针绘制，到 end 结束。

（4）aspect 参数决定所画椭圆纵轴与横轴的比值。比值大于 1 时，绘制扁形椭圆（垂直方向大于水平方向）；小于 1 时绘制椭圆；等于 1 时绘制圆。

（5）在 VB 坐标系中，采用逆时针方向绘圆。Circle 方法中参数前出现的负号，并不能改变坐标系中旋转的方向。使用 Circle 方法时，可省掉参数但不能省掉逗号。

例如：

```
Circle (150,150),100                              '以(150,150)为圆心,100 为半径画一个圆
Circle (100,100),100,vbRed, - 3.14/2, - 3.14      '画扫过 π/2 的扇形
Circle(100,100),100,vbRed,3.0 * 3.14/2,0          '画一圆弧
Circle(100,100),100,vbRed,,,1.5                   '画一高宽比为 1.5 的椭圆
```

以上画扇形和圆弧的区别在于负号“－”，负号不代表负角度。

5.8.4 PSet 方法

PSet 方法可以在窗体、图形框或打印机等对象指定的位置用 color 参数给定的颜色画一个点。点的大小由对象的 DrawWidth 属性指定。PSet 方法用来在指定位置（x，y）处画一个点。当一些特殊的图形无法用直线和弧线组成时，可以采用 PSet 方法来逐点绘制。

语法格式：

```
[Object.PSet[Step](x,y)[,color]
```

说明：（x，y）是画点的坐标，x 和 y 参数是单精度参数，可接受整数或分数的输入。color 用来指定绘制点的颜色，数据类型为 Long。默认时，系统用对象的 ForeColor 属性值作为绘制点的颜色。color 参数还可用 QBColor()、RGB()函数指定。Step 关键字是下一个画点位置相对于当前位置的偏移量的标记，即步长（水平、垂直两个方向，可正可负）。（x，y）坐标值是相对于当前位置的偏移量。执行 PSet 方法后，CurrentX 和 CurrentY 属性被设置为参数指定的点。

例如：

```
PSet(100,200)                                     '将(100,200)点的颜色改成黑色(默认)
PSet(120,220),vbRed                               '把(120,220)点的颜色改成红色
```

例 5.17 使用 PSet 方法绘制正弦函数图像。程序执行结果如图 5-26 所示。

```
Const PI = 3.1415926
Private Sub Form_Click()
    Form1.Scale (0, 0) - (5000, 2000)
    Line (0, 1000) - (5000, 1000)
    Line (200, 0) - (200, 2200)
    For i = 0 To 5000
        x = 200 + i
        y = 1000 + 1000 * Sin(PI * i / 1800)
```

```
            PSet (x, y)
        Next i
End Sub
```

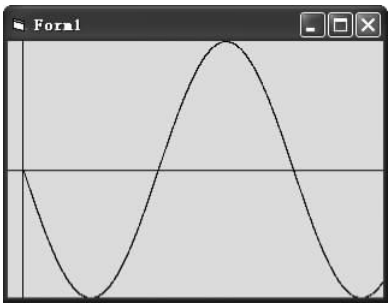

图 5-26　用 PSet 方法绘制正弦函数图像

5.8.5　Point 方法

Point 方法用于获取窗体或图片框上指定点的 RGB 颜色值。

语法：

```
[Object.] Point (x,y)
```

如果(x,y)坐标所引用的点位于对象之外，Point 方法将返回－1(True)。

5.8.6　PaintPicture 方法

PaintPicture 方法是窗体或图片框的一个很实用的方法，它能够将窗体或图片框中的一个矩形区域的像素复制到另一个对象上。使用 PaintPicture 方法，可以在窗体、图片框和 Printer 对象上的任何地方绘制图形，对图形进行复制、翻转、改变大小、重新定位及水平或垂直翻转等操作。但是，该方法只能对用 Picture 属性、LoadPicture 函数设置的图形进行操作，用绘图方法绘制的图形在未存储成图形文件前不能用它操作。

语法格式：

```
[object.]PaintPicture pic,dx,dy,[dw,dh,]sx,sy,[sw,sh],opcode
```

说明：

(1) object：是可选的窗体或图片框对象名字。如果省略 object，默认对象为带有焦点的窗体。

(2) pic：要绘制到 object 上的图形源。它是由窗体或图片框的 Picture 属性决定的。

(3) dx,dy：是传送目标矩形区域左上角坐标，可以是目标控件的任一位置。dw,dh 是目标矩形区域的宽和高。

(4) sx、sy：是要传送图形矩形区域左上角坐标，sw、sh 是要传送图形区域的大小。

(5) opcode：指定传送的像素与目标中现有的像素组合模式，其取值如表 5-18 所示。

除 opcode 外，PaintPicture 方法中的参数度量单位要受 ScaleMode 属性的影响，结果要受 AutoRedraw 属性的影响，在使用该方法前，最好将 ScaleMode 属性设置为像素，

AutoRedraw 属性设置为 True。

表 5-18 像素组合模式

常　量	数　值	说　明
vbDstInvert	&H00055009	逆转目标位图
vbNotSrcCopy	&H00330008	复制源位图的逆到目标位图
vbSrcCopy	&H00cc0020	复制源位图到目标位图
vbSrcInvert	&H00660046	用 XOR 组合源位图与目标位图

在使用 PaintPicture 方法复制时翻转只要改变坐标系即可。如果设置图形宽为负数，则水平翻转图形；如果设置图形高度为负数，则上下翻转图形；如果宽度和高度都为负数，则两个方向翻转图形。例如，目标宽度为负数，PaintPicture 方法将像素复制到原点的左边，如果控件的坐标原点在左上角，目标图形就在控件以外，为使目标图形复制到控件中，必须将原点设置到另一角，实现时可以任意选定源或目标的坐标系。

5.9 键盘、鼠标事件

在 Windows 应用软件中，大量用到鼠标、键盘事件，其中，Visual Basic 软件编程中鼠标的 Click 事件、DblClick 事件和键盘的 KeyPress 事件就是最常用的事件。Visual Basic 应用程序中大多数控件都能够响应多种鼠标事件和键盘事件。

5.9.1 鼠标事件

在事件驱动应用程序中，由对象来识别事件。事件可以由一个用户动作产生，如单击鼠标或按下一个键；也可以由程序代码或系统产生，如计时器。目前，在应用程序中，每个对象，如窗体、控件、菜单等都可以编写事件代码。触发对象事件的最常见的方式是通过鼠标或键盘的操作。一般将通过鼠标触发的事件称为鼠标事件，将通过键盘触发的事件称为键盘事件。

目前，在 Windows 应用软件中，多数应用程序是通过鼠标来操作的，如用鼠标单击按钮、选择菜单等。鼠标的操作主要有单击、双击、移动等几种，它们分别能触发一个事件。主要事件内容如下。

（1）MouseMove 事件：每当鼠标指针移动到屏幕新位置时发生。

（2）MouseDown 事件：按下任意鼠标键时发生。

（3）MouseUp 事件：释放任意鼠标键时发生。

通过这些鼠标事件，应用程序能对鼠标位置及状态的变化做出相应操作。MouseMove、MouseDown、MouseUp 三个事件处理过程的语法格式为：

（1）

```
Sub Object MouseMove(Button As Integer,Shift As Integer,X As Single,Y As Single)
```

（2）

```
Sub Object_MouseDown(Button As Integer,Shift As Integer,X As Single,Y As Single)
```

(3)

```
Sub Object_MouseUp(Button As Integer,Shift As Integer,X As Single,Y As Single)
```

说明：

(1) Object：是可选的一个对象表达式，可以是窗体对象和大多数可视控件。

(2) Button：表示鼠标按下或松开哪个按键，鼠标的不同按键，得到的值是不同的。

Button 参数是一个低三位二进制数 b_2 b_1 b_0，分别表示中间按键、右按键、左按键的状态(即 M、R、L)，相应二进制位如下。

b_2＝0 或 1：当为 1 时，表示鼠标中间键按下或释放。

b_1＝0 或 1：当为 1 时，表示鼠标右键按下或释放。

b_0＝0 或 1：当为 1 时，表示鼠标左键按下或释放。

其中，0 时表示未按下对应按钮，为 1 时表示按下了对应按键或释放按键。

(3) Shift：表示在 Button 参数指定的按键被按下或者被松开的情况下，键盘的 Shift、Ctrl 和 Alt 键的状态，通过该参数可以处理鼠标与键盘的组合操作。

其中，Shift 参数是一个低三位二进制数 b_2 b_1 b_0，相应二进制位如下。

b_2＝1：当为 1 时，表示按下 Alt 键。

b_1＝1：当为 1 时，表示按下 Ctrl 键。

b_0＝1：当为 1 时，表示按下 Shift 键。

表 5-19 列出了各种可能的按键组合中 Shift 参数的值。

表 5-19　Shift 参数的值

二进制值	十进制值	系 统 常 数	意　义
000	0		未按下任何键
011	3	vbShiftMask＋vbCtrlMask	同时按下 Shift 和 Ctrl 键
101	5	vbShiftMask＋vbAltMask	同时按下 Shift 和 Alt 键
110	6	vbCtrlMask＋vbAltMask	同时按下 Ctrl 和 Alt 键
111	7	vbCtrlMask＋vbAltMask＋vbShiftMask	同时按下 Ctrl、Alt 和 Shift 键

(4) X 和 Y：为鼠标指针的位置，通过 X 和 Y 参数返回一个指定鼠标指针当前位置的数，鼠标指针的位置使用该对象的坐标系统表示。

5.9.2　键盘事件

在 Windows 应用程序中，最常用的是鼠标操作，但有时也需要使用键盘操作，尤其是对于接收文本输入的控件，如文本框 TextBox。若需要控制文本框中输入的内容，处理 ASCII 字符，就需要对键盘事件编程。

键盘的按键操作实际上也会触发三个事件，分别是 KeyPress、KeyDown、KeyUp。当按下某键后，触发 KeyDown 事件，键被弹起后，触发 KeyUp 事件，同时又触发了 KeyPress 事件。

1. KeyPress 事件

在按下与 ASCII 字符对应的键时将触发 KeyPress 事件。ASCII 字符集不仅代表标准

键盘的字母、数字和标点符号，而且也代表大多数控制键。但是 KeyPress 事件只能识别 Enter（回车键）、Tab（制表位键）、BackSpace（退格键）三个功能键，不能够检测其他功能键。

KeyPress 事件过程的语法格式是：

```
Sub Object_KeyPress(KeyAscii As Integer)
```

Object：窗体或控件对象名。

KeyAscii：返回对应于 ASCII 字符代码的整型数值。

如果在应用程序中，通过编程来处理标准 ASCII 字符，应使用 KeyPress 事件。例如，可通过下面的代码将文本框中输入的所有字符都强制转换为大写字符。

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
   KeyAscii = Asc(Ucase(Chr(KeyAscii)))
End Sub
```

其中，Chr 函数将 ASCII 字符代码转换成对应的字符，然后用 Ucase 函数将字符转换为大写，并用 Asc 函数将结果转换回字符代码。

另外，也可通过判断 ASCII 字符代码来检测是否按下某个键。例如，下面的代码用于检测用户是否正在按 BackSpace 键。

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 8 Then MsgBox "You pressed the BACKSPACE key."
End Silb
```

其中，Visual Basic 中 BackSpace 键的 ASCII 值为 8，其常数值为 vbKeyBack。

例 5.18 通过编程序，在一个文本框（Text1）中限定只能输入数字、小数点，只能响应 BackSpace 键及回车键。

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
       Case 48 to 57,46,8,13
    Case Else
       KeyAscii = 0
    End Select
End Sub
```

说明：

（1）0～9 的数字字符的 ASCII 码值是 48～57，小数点的 ASCII 码值是 46；

（2）BackSpace 键的 ASCII 码值是 8，回车键的 ASCII 码值是 13；

（3）按其他键，让 KeyAscii ＝ 0 时，即不接受其他键的操作。

2. KeyDown 和 KeyUp 事件

当一个对象具有焦点时按下一个键则触发 KeyDown 事件，松开一个键则触发 KeyUp 事件。与 KeyPress 事件相比，KeyDown 和 KeyUp 事件能够报告键盘本身准确的物理状态：按下键（KeyDown）及松开键（KeyUp）。而 KeyPress 事件只能提供键所代表的字符 ASCII 码而不识别键的按下或松开状态。此外，KeyDown 和 KeyUp 事件能够检测各种功能键、编辑键和定位键，而 KeyPress 事件只能识别 Enter、Tab 和 BackSpace 键。

KeyUp 和 KeyDown 事件过程的语法格式为：

```
Sub Object_KeyDown(KeyCode As Integer,Shift As Integer)
Sub Object_KeyUp(KeyCode As Integer,Shift As Integer)
```

其中：

1）KeyCode 参数

KeyCode 表示按下的物理键，通过 ASCII 值或键代码常数来识别键。其中，大小字母使用同一键，它们的 KeyCode 相同，为大写字母的 ASCII 码，如“A”和“a”的 KeyCode 都是由 Asc("A")返回的数值，其数值为 65。上档键字符和下档键字符也是使用同一键，它们的 KeyCode 值也是相同的，为下档字符的 ASCII 码，如“：”与“；”使用同一键，它们的 KeyCode 相同。此外，键盘上的“1”和数字小键盘的“1”被作为不同的键返回，尽管它们生成相同的字符，但它们的 KeyCode 值是不相同的。

表 5-20 列出了部分字符的 KeyDown 或 KeyUp 事件的 KeyCode 和 KeyPress 事件的 KeyAscii 值，注意它们的值有所不同，区别它们的异同。

表 5-20　KeyCode 和 KeyAscii 值

键（字符）	KeyCode 值（十六进制/十进制）	KeyAscii 值（十六进制/十进制）
"A"	&H41(65)	&H41(65)
"a"	&H41(65)	&H61(97)
"#"	&H33(51)	&23(35)
"2"	&H33(51)	&33(51)
"1"（大键盘）	&H31(49)	&H31(49)
"1"（数字小键盘）	&H61(97)	&H31(49)
Home 键	&H24(36)	&H24(36)
F10 键	&H79(121)	无

用 KeyDown 事件判断是否按下了 A 键，语句的表述为：

```
Private Sub Text1_KeyDown(KeyCode As Integer,Shift As Integer)
    If KeyCode = vbKeyA Then
        MsgBox "You pressed the A key"
    End IF
End Sub
```

KeyDown 和 KeyUp 事件可识别标准键盘上的大多数控制键，其中包括功能键（F1～F16）、编辑键（Home、Page Up、Delete 等）、定位键和数字小键盘上的键。

可以通过键代码常数或相应的 ASCII 值检测这些键。例如：

```
Private Sub Text1_KeyDown(KeyCode As Integer,Shift As Integer)
    If KeyCode = vbKeyHome Then
        MsgBox "You pressed the HOME key"
    End IF
End Sub
```

对于字符代码的表示，可通过 MSDN 的“Visual Basic 文档”进行查找，得到它们的完整列表。也可通过“对象浏览器”搜索 KeyCodeConstants 获得此列表。

2）Shift 参数

Shift 参数主要用来表示 Shift、Ctrl 和 Alt 键的状态，其含义与 MouseMove、MouseDown、MouseUp 事件中的 Shift 参数完全相同。

由于英文有大小写，为区分大小写，KeyDown 和 KeyUp 事件常需要使用 Shift 参数，而 KeyPress 事件将字母的大小写作为两个不同的 ASCII 字符来处理，需要特别注意。

利用 Shift 参数来判断是否按下字母的大小写情况，如为 A 字符，语句的表示为：

```
Private Sub Text1_KeyDown(KeyCode As Integer,Shift As Integer)
    If KeyCode = vbKeyA And Shi ft = 1 Then
       MsgBox "You pressed the uppercase A key"
    End i f
End Sub
```

数字与标点符号键的键代码与键上数字的 ASCII 代码相同，因此，“3”和“＃”的 KeyCode 由 Asc("3")返回的数值为 51。同样，为检测“＃”，如需使用 Shift 参数，语句表示为：

```
Private Sub Text1_KeyDown(KeyCode As Integer,Shift As Integer)
  If KeyCode = vbKey3 And shift = l Then
      MsgBox "You pressed the ＃ key"
  End i f
End Sub
```

习题

一、判断题

1. 如果要时钟控件每分钟发生一个 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. 计时器控件在 Visual Basic 应用程序启动后自动计时，无法暂停或关闭。（　　）

3. 要在同一窗体中建立几组相互独立的单选按钮，就要用框架将每一组单选按钮框起来。（　　）

4. 如果将框架控件的 Enabled 属性设为 False，则框架内的控件都不可用。（　　）

5. 清除 List1 列表框对象的内容的语句是 List1.Cls，清除 Combo1 组合框中的内容的语句是 Combo1.Clear。（　　）

6. 若在列表框中第 5 项之后插入一项目“ABCD”，则所用语句为 List1.AddItem "ABCD",6。（　　）

7. 组合框的 Change 事件在用户改变组合框的选中项时被触发。（　　）

8. 不同控件具有不完全相同的属性集合。一些属性是所有控件共有的，一些属性则是部分控件所特有的。（　　）

9. 一些属性既可以在“属性”窗口中设置，又可以在代码中进行修改。另一些属性则是只读的，只能在设计阶段进行设置。（　　）

10. 除了标准控件以外，Visual Basic 可以使用其他控件、用户自定义控件和第三方厂

商研制的控件。（　　）

11. 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。（　　）

12. 在用户拖动滚动滑块时，滚动条的 Change 事件连续发生。（　　）

13. 触发 KeyPress 事件必定触发 KeyDown 事件。（　　）

14. 当按下键并放开，将依次触发获得焦点对象的 KeyDown、KeyUp、KeyPress 事件。（　　）

15. 如果在 KeyDown 事件中将 KeyCode 设置为 0，KeyPress 的 KeyAscii 参数不会受影响。（　　）

16. 当单击鼠标时，将依次触发所指向对象的 MouseDown、MouseUp 和 Click 事件。（　　）

二、填空题

1. 将文本框的 ScrollBars 属性设置为 2(有垂直滚动条)，但没有出现垂直滚动条，这是因为没有将_____属性设置为 True。

2. 检查框的_____属性设置为 2-grayed 时，将变成灰色。

3. 列表框中的_____和_____属性是数组。列表框中项目的序号是从_____开始的。列表框 List1 中最后一项的序号用_____表示。

4. 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。_____风格的组合框不允许用户输入列表中没有的项。

5. 当用户单击滚动条的空白处时，滑块移动的增量值由_____属性决定。滚动条产生 Change 事件是因为_____值改变了。

6. 在鼠标事件中(如 MouseMove 事件)，Shift 参数为 1 表示在操作鼠标的同时也按下键盘上的 Shift 键，为 2 表示同时按下了键盘上的 Ctrl 键，那么 Shift 参数为 3 时，表示同时按下键盘上的_____。

7. 要使一个文本框不接收任何键盘输入字符，在 KeyPress 事件中使用_____可实现。如果要使一个文本框只接收数字字符输入，在 KeyPress 事件中使用_____可实现。

8. PictureBox 控件的 AutoSize 属性设置为 True 时，_____能自动调整大小。

三、程序填空

1. 完成一个字体设置程序的设计，要求分别单击三个组合列表框的列表项时，都能实现对标签控件 Label1“VB 程序设计：”字体的设置。程序启动后，组合列表框 Combo3 的文本框显示为 12，对 Combo3 的相关属性做合理设置。

```
Private Sub Form Load()                    '给组合框 Combo3 中添加字号
    Dim i As Integer
    For i = 4 To 72 Step 4
       Combo3.AddItem Str(i)
    Next i
End Sub
Private Sub Combo1_Click()                 '选择并设置字体
```

```
    ____ 1 ____
End Sub
Private Sub Combo2_Click()              '选择并设置字型
    Select Case ____ 2 ____
        Case"常规"
            Label1.FontBold = False
            Label1.FontItalic = False
        Case"斜体"
            Label1.FontBold = False
            Label1.FontItalic = True
        Case"粗体"
            Label1.FontBold = True
            Label1.FontItalic = False
        Case"粗体斜体"
            ____ 3 ____
            ____ 4 ____
    End Select
End Sub
Private Sub Combo3_Click()              '选择并设置字号
    ____5____
End Sub
```

2. 下列程序段的功能是交换如图 5-27 所示的两个列表框中的项目。当双击某个项目时，该项目从本列表框中消失，并出现在另一个列表框中。列表 1 的名称为 List1，列表 2 的名称为 List2。

```
Private Sub Form_Load()
  Listl.AddItem "IBM"
  Listl.AddItem "Compaq"
  Listl.AddItem "AST"
  …
End Sub
Private Sub List1_DblClick()
  LiSt2.AddItem ____ 1 ____
  ____ 2 ____________________
End Sub
Private Sub List2_DblClick()
  _________ 3 _____________
  ____ 4 _________ List2.ListIndex
End Sub
```

图 5-27　程序界面

四、选择题

1. 将数据项“CHINA”添加到列表框 List1 中成为第一项，使用(　　)语句。

A. List1.AddItem,"CHINA",0　　B. List1.AddItem,"CHINA",1

C. List1.AddItem,0,"CHINA"　　D. List1.AddItem,1,"CHINA"

2. 执行下面的程序后，列表框中的数据项为(　　)。

```
Private Sub Form_Click()
    Dim I As Integer
    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

3. 如果列表框 List1 中没有被选定的项目,则执行 List1.RemoveItem List1.ListIndex 语句的结果是(　　)。

A. 移去第一项　　B. 移去最后一项

C. 移去最后加入列表的一项　　D. 以上都不对

4. 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 窗口中输出被选定项目的索引值

B. 在 Debug 窗口中输出 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

5. 假定时钟控件 Timer1 的 Interval 属性为 1000,Enabled 属性为 True,并且有下面的事件过程,计算机将发出(　　)次 Beep 声。

```
Private Sub Timer1_Timer()
    Dim I AS Integer
    For I = 1 to 10
    Beep
    Next I
End Sub
```

A. 1000 次　　B. 10 000 次　　C. 10 次　　D. 以上都不对

6. 在下列说法中,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顺序从大到小排列

D. 框架也有 Click 和 DblClick 事件

7. 当程序运行时,在窗体上单击鼠标,以下(　　)事件是窗体不会接收到的。

A. MouseDown　　B. MouseUp　　C. Load　　D. Click

8. 有以下事件过程,程序运行后,为了在窗体上输出“Hello”,应在窗体上执行(　　)。

A. 同时按下 Shift 键和鼠标左键　　B. 同时按下 Shift 键和鼠标右键

C. 同时按下 Ctrl、Alt 键和鼠标左键　　D. 同时按下 Ctrl、Alt 键和鼠标右键

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Shift = 6 And Button = 2 Then
       Print "Hello"
    End If
End Sub
```

9. 以下的属性和方法中(　　)可重定义坐标系。

A. DrawStyle 属性　　B. DrawWidth 属性

C. Scale 方法　　D. ScaleMode 属性

10. 当使用 Line 方法画直线后，当前坐标在(　　)。

A. (0,0)　　B. 直线起点　　C. 直线终点　　D. 容器的中心

11. 语句 Circle(1000,1000),500,8,－6,－3 将绘制(　　)。

A. 圆　　B. 椭圆　　C. 圆弧　　D. 扇形

12. 执行指令“Line(1200,1200)--Step(1000,500),B”后，CurrentX＝(　　)。

A. 2200　　B. 1200　　C. 1000　　D. 1700

13. 下列(　　)途径在程序运行时不能将图片添加到窗体、图片框或图像框的 Picture 属性。

A. 使用 LoadPicture 方法　　B. 对象间图片的复制

C. 通过剪贴板复制图片　　D. 使用拖放操作

14. 设计时添加到图片框或图像框的图片数据保存在(　　)。

A. 窗体的 Frm 文件　　B. 窗体的 Frx 文件

C. 图片的原始文件内　　D. 编译后创建的 Exe 文件

15. 当窗体的 AutoRedraw 属性采用默认值时，若在窗体装入时使用绘图方法绘制图形，则应将程序放在(　　)。

A. Paint 事件　　B. Load 事件

C. Initialize 事件　　D. Click 事件

16. 当对 DrawWidth 进行设置后，将影响(　　)。

A. Line、Circle、PSet 方法

B. Line、Shape 控件

C. Line、Circle、Point 方法

D. Line、Circle、PSet 方法和 Line、Shape 控件

17. 窗体 Form1 在左上角坐标为(－200,250)，窗体 Form1 在右下角坐标为(300,－150)，X 轴和 Y 轴的正向分别为(　　)。

A. 向右、向下　　B. 向左、向上　　C. 向右、向上　　D. 向左、向下

五、编程题

1. 设计如图 5-28 所示的添加和删除程序，根据要求编写相应的事件代码。

(1) 在组合框中输入内容后，单击“添加”按钮，如果组合框中没有该内容，则将输入内容加入到列表中，否则将不添加。另外，要求组合框中内容能自动按字母排序。

(2) 在列表中选择某一选项后，单击“删除”按钮，则删除该项。在组合框中输入内容后，单击“删除”按钮，若列表中有与之相同的选项，则删除该项。

(3) 单击“清除”按钮，将清除列表中的所有内容。

图 5-28　添加和删除程序

2. 设计如图 5-29 所示的“偶数迁移”程序。根据要求编写相应的事件代码。

图 5-29　“偶数迁移”程序运行效果

（1）窗体的左边有一个标签 Label1，标题为“两位正整数：”，标签的下面是一个列表框 List1。

（2）窗体的右边有一个标签 Label2，标题为“偶数：”，标签的下面是一个列表框 List2。

（3）单击“产生”按钮（Command1），计算机产生 10 个两位正整数放入列表框 List1 中。同时清空列表框 List2 中的内容。

（4）单击“-->”按钮（Command2），将列表框 List1 中所有偶数迁移到列表框 List2 中。

3. 设计一个如图 5-30 所示的点歌程序。窗体包含两个列表框，当双击歌谱列表框中的某首歌时，此歌便添加到已点歌曲列表框中，在已点的歌列表框中双击某歌时，此歌便被删除。

图 5-30　点歌程序