

# 项目

# 4

## 使用工具箱中常用控件



### 技能目标

能根据项目要求正确使用工具箱中的常用控件，并能设置属性、编写事件代码。



### 知识目标

- 了解工具箱中的控件
- 掌握工具箱中常用控件的使用



### 项目任务

本项目完成向窗体中添加工具箱常用控件及设置控件属性、编写事件代码。



### 项目解析

Visual Studio 2008 控件是用来创建 Visual Basic 程序用户界面的图形工具。控件放置在开发环境的工具箱中，用户通过一系列简单的单击和拖动操作可在窗体上创建控件对象。为了方便使用工具箱中控件，Visual Studio 2008 对控件进行了分类，以选项卡形式列出。

Visual Studio 2008 可以创建两类窗体：Windows 窗体和 Web 窗体页。其中公共控件既可以在 Windows 窗体上添加，也可以在 Web 窗体页上添加。公共控件中有些控件经常使用，我们称之为常用控件。

在窗体中要能准确快速地添加控件，首先必须知道该控件在工具箱哪一个选项卡中，对于常用的控件，用户不仅能准确地添加，而且会使用。

根据知识的学习规律，本项目被分解成两个子任务：工具箱中的控件认识和使用常用控件。

## 任务一 认识工具箱中的控件

### 一、任务描述

认识工具箱中各选项卡的用途。

### 二、任务分析

Visual Studio 2008 对控件进行了分类，以选项卡的形式呈现在工具箱中。程序开发者若想

熟练使用控件，首先应对工具箱中的各选项卡有所了解。窗体上包含许多控件，用户可以使用对齐控件将窗体上的控件快速对齐。

### 三、任务实现

#### 1. 基本知识

控件是窗体上用户可交互的各种工具，是构成应用程序界面的部件，没有控件的窗体是没有多少意义的。Visual Studio 2008 中所有可添加到窗体中的控件都位于工具箱中。默认情况下，工具箱停靠在设计环境的左边。当要向窗体中添加控件时，只需将鼠标移到工具箱上，此时会弹出如图 4-1 所示的工具箱。

将工具箱中的控件添加到窗体有以下三种方法：

- ① 在工具箱中双击控件。
- ② 从工具箱中将所选中的控件拖到窗体中。
- ③ 绘制控件。从工具箱中选中控件，将鼠标移至窗体中希望的位置，按住鼠标左键不放拖动鼠标，然后松开鼠标左键。

工具箱中的控件按使用类别进行了分类，包括【所有 Windows 窗体】、【公共控件】、【容器】、【菜单和工具栏】、【数据】、【组件】、【打印】、【对话框】、【WPF 互操作性】、【报表】、【Visual Basic PowerPacks】、【常规】选项卡，用户在使用时可以展开和折叠分类标题。

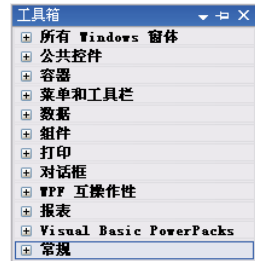


图 4-1 工具箱

【公共控件】选项卡提供了添加到窗体上的常用控件，它包括 Button 控件、文本控件、选择控件、列表选择控件、图形显示和定时器控件等；【菜单和工具栏】选项卡中的控件用于在窗体上创建菜单栏、工具栏和状态栏；【数据】选项卡提供了操作数据库的控件；【打印】选项卡中的控件是与打印机操作相关的控件；【对话框】选项卡的控件用于打开 Windows 通用对话框，如“颜色”对话框、“字体”对话框；【报表】选项卡中的控件可创建 Crystal Report 报表和 Microsoft Report 报表。

#### 2. 设置控件的属性

控件的属性既可以通过代码设置，又可以使用【属性】窗口设置。








#### 3. 使用对齐控件对齐控件

Visual Basic 包含许多格式工具，可用于设计美观的界面，它们大多数都在【布局】工具栏中。右击 Visual Basic 顶部的工具栏，从弹出的菜单中选择【布局】选项，显示【布局】工具栏。【布局】工具栏包含水平和垂直对齐控件边缘或中心的选项，如图 4-2 所示。



图 4-2 【布局】工具栏

使用【布局】工具栏，可以：

- ：使选中的控件左对齐、居中对齐或右对齐。
- ：使选中的控件顶部对齐、中间对齐或底部对齐。
- ：使选中的控件等宽、等高或高和宽相同。
- ：使选中的控件在水平方向进行调整。
- ：使选中的控件在垂直方向进行调整。
- ：使选中的控件在水平方向或垂直方向的间距相等。
- ：将选中的控件前移或后移。

## 任务二 使用工具箱中常用控件

### 一、任务描述

工具箱中【公共控件】选项卡中包含窗体上常用的控件和【组件】选项卡中常用的组件。这些控件和组件用于接受用户输入及显示程序的输入信息。不同控件具有不同的属性，只有熟悉控件的功能和属性才能正确使用。

### 二、任务分析

Windows 中经常使用以下控件：

- 文本控件：Label 控件、LinkLabel 控件、TextBox 控件、RichTextBox 控件。
- 按钮控件：Button 控件。
- 选择控件：RadioButton 控件、GroupBox 控件、CheckBox 控件。
- 列表控件：ListBox 控件。
- 数字控件：NumericUpDown 控件。
- 下拉列表控件：ComboBox 控件。
- 进度条控件：ProgressBar 控件。
- 定时控件和图片控件：Timer（定时）控件、PictureBox 控件。


其中，RichTextBox 控件将在项目 9 中介绍，其他控件在本任务中介绍。

控件除具有相同的属性（如 Name 属性、Width 属性、Height 属性）外，还具有不同的属性，这些属性决定控件的不同。此外，控件还具有方法和事件，不同的方法和不同的事件产生不同的动作结果。

### 三、任务实现

#### 1. Label 控件

Label（标签）控件通常用于显示文本信息，而文本信息不能被用户直接编辑，所以标签

控件通常用于为其他控件提供说明性文本，如文本框控件（TextBox）、列表框控件（ListBox）等。标签一般不用于触发事件。Label 控件在工具箱中的图标为  Label。

### （1）常用属性

Label 控件的常用属性如表 4-1 所示。


表 4-1 Label 控件的常用属性

属性	作用
AutoSize	设置控件的大小是否随标题内容的大小自动调整。True: 自动调整, False: 不自动调整。默认值为 False。只对不换行的文本标签有效
BorderStyle	设置标签的边框样式。None: 无边框 (默认值), FixedSingle: 单线边框, Fixed3D: 边框具有 3D 效果
Enabled	设置控件是否允许操作。True: 允许操作, 默认值; False: 禁止操作, 此时标签呈灰色
Font	用于设置标签中文本的字体
Image	设置标签上显示的图片
ImageAlign	设置标签上图片对齐的方式
Name	用于唯一标识标签的名称, 便于代码中引用该对象, 默认值是 LabelN (N=1,2,...)
Text	用于设置在标签上显示的文本信息, 默认值为 LabelN (N=1,2,...)。可以在【属性】窗口中设置, 也可以在程序中修改。在程序中修改标题属性的语句格式如下: 标签控件名.text="显示的文本"
TextAlign	指定标签上文本的位置
TabIndex	设置此控件占用的 Tab 键顺序索引
Visible	用于设置标签是否可见。True: 可见 (默认值), False: 不可见

### （2）常用事件

Label 控件一般没有事件。

## 2. LinkLabel 控件

LinkLabel 控件用于用户在 Windows 窗体应用程序中添加 Web 样式的链接。一切可以使用 Label 控件的地方都可以使用 LinkLabel 控件。LinkLabel 控件在工具箱中的图标为  LinkLabel。

### （1）常用属性

LinkLabel 控件除具有 Label 控件的属性外, 还具有以下属性:

- LinkArea 属性: 标签中要呈现为超链接的文本部分。
- LinkBehavior 属性: 确定该超链接的下划线行为。
- ActiveLinkColor 属性: 确定当用户单击超链接时该链接的颜色。
- LinkColor 属性: 确定超链接处于默认状态时的颜色。
- LinkVisited 属性: 设置为 True 时表示该链接已经被访问。
- VisitedLinkColor 属性: 当链接被访问后显示的颜色。

### （2）事件和方法

LinkClicked 方法: 单击超链接后产生的事件。

**【案例 4-1】Label 和 LinkLabel 控件的应用**

要求:新建一个名为 Library 的项目,在窗体上添加一个 Label 控件和一个 LinkLabel 控件。窗体的 Text 属性设定为“小型图书管理系统”,Label 控件的 Text 属性设定为“欢迎使用小型图书管理系统”,Font 的 Name 属性为“微软雅黑”、Size 属性为 15; LinkLabel 控件的 Text 属性设定为“新浪或网易”。单击“新浪”或“网易”可打开其网页,效果如图 4-3 所示。

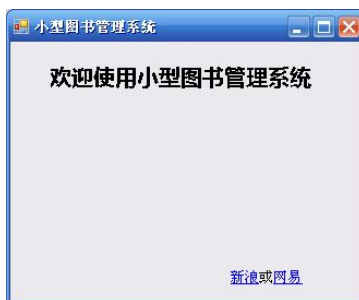


图 4-3 【案例 4-1】结果

步骤 1: 启动 Visual Studio 2008, 新建一个名为“Library”的项目。

步骤 2: 单击窗体, 按 F4 键打开【属性】窗口。将窗体的 Text 属性设定为“小型图书管理系统”。

步骤 3: 从工具箱中分别将 Label 控件和 LinkLabel 控件拖到窗体中。

步骤 4: 设置 Label 控件和 LinkLabel 控件的属性。将 Label 控件的 Text 属性设定为“欢迎使用小型图书管理系统”, Font 的 Name 属性设定为“微软雅黑”、Size 属性设定为 15; 将 LinkLabel 的 Text 属性设定为“新浪或网易”。


步骤 5: 为窗体的 Load 事件和 LinkLabel 控件的 LinkClicked 事件编写代码:

```
Public Class Form1
    Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs)
        Dim target As String = e.Link.LinkData.ToString()
        System.Diagnostics.Process.Start(target)
        e.Link.Visited = True
    End Sub
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Me.LinkLabel1.Text = "新浪或网易"
        Me.LinkLabel1.Links.Add(0, 2, "www.sina.com.cn") '设置新浪为链接
        Me.LinkLabel1.Links.Add(3, 2, "www.163.com") '设置网易为链接
    End Sub
End Class
```

步骤 6: 按 F5 键运行程序, 结果如图 4-3 所示。单击【新浪】或【网易】时打开相应的网页。

### 3. TextBox 控件

TextBox (文本框) 控件是 Windows 应用程序中最常用的控件之一, 它既能显示文本又能

接受用户的输入。TextBox 控件可以用来输入单行文本，也可以将文本框设置成多行的，用于输入整个文档。TextBox 控件还可以充当密码输入框。工具箱中 TextBox 控件的图标为  TextBox。

#### (1) 常用属性

TextBox 控件的常用属性如表 4-2 所示。

表 4-2 TextBox 控件的常用属性

属性	作用
AcceptReturn	指定在多行编辑控件中是否允许输入回车符。False: 不允许(默认值), True: 允许
AcceptsTab	指定在多行文本控件中按 Tab 键时, 是否作为制表符输入。False: 否(默认值), 按 Tab 键移至下一个对象; True: 是
CharacterCasing	指定输入的所有字符保持不变还是应转换为大写或小写。Normal: 不变(默认), Upper: 大写, Lower: 小写
HideSelection	指定当编辑控件失去焦点时, 是否应隐藏选定内容。True: 隐藏(默认值), False: 不隐藏
Lines	获取或设置文本框控件中的文本行数
MaxLength	指定文本框中可以输入的最大字符数
Multiline	设置文本框是否可输入多行文本。True: 多行文本, False: 单行文本(默认值)
ReadOnly	设置指定文本框中文本是否为只读。True: 只读, False: 可修改(默认值)
Passwordchar	指定用户在文本框中输入字符时显示的字符。例如, 将该值设置为“*”, 则用户在文本框中输入的任意字符都将显示为“*”
ScrollBars	指定对于多行文本编辑框显示哪些滚动条。None: 无滚动条, Horizontal: 垂直滚动条, Vertical: 水平滚动条, Both: 垂直和水平滚动条都有
Text	文本框的内容
WordWrap	指定多行文本编辑框是否自动换行。True: 自动换行, False: 不自动换行

#### (2) 常用事件

TextBox 的常用事件如表 4-3 所示。

表 4-3 TextBox 控件的常用事件

事件	作用
TextChanged	文本更改时触发该事件
KeyPress	当文本框具有焦点时, 按下任意键, 就会触发该事件。通常利用该事件检查输入的字符
GotFocus	用鼠标单击该对象使其获得焦点时, 触发该事件

#### (3) 常用方法

TextBox 的常用方法如表 4-4 所示。

### 4. Button 控件

Button 控件又称按钮控件。在 VB.NET 的 Windows 应用程序设计中, 用于建立项目的按


钮界面。工具箱中 Button 控件的图标为  Button。

表 4-4 TextBox 控件的常用方法

事件	作用
SelectAll	选择 TextBox 框中的全部文本
Focus	使 TextBox 获得焦点
Copy	复制选取的文本
Cut	剪切选取的文本
Paste	粘贴剪贴板上的文本

#### (1) 常用属性

Button 控件的常用属性如表 4-5 所示。

表 4-5 Button 控件的常用属性

属性	作用
Enabled	指定按钮是否有效。True: 按钮有效, False: 按钮无效, 呈灰色显示
Text	设置按钮上显示的文本
TextAlign	设置按钮上文本的对齐方式
Image	设置按钮上显示的图片
ImageAlign	设置按钮上图片的对齐方式
BackColor	设置按钮控件的背景颜色
ForeColor	设置按钮控件所包含文本或图像的前景颜色
Font	设置按钮上显示文本的字体
Visible	设置按钮控件是否可见。True: 可见 (默认值), False: 不可见
FlatStyle	设置用户将鼠标移到控件上并单击时按钮的外观, 有以下几个选项: Flat: 控件以平面显示 Popap: 控件以平面显示, 当鼠标指针移到该控件上时, 该控件外观显示为 Windows 标准按钮 Standard: 指定控件外观为标准按钮 (默认) System: 指定控件外观由用户的操作系统决定

#### (2) Button 控件的常用事件

Button 控件的常用事件只有 Click 事件, 当用户用鼠标单击对象时触发该事件。

**注意:** Button 控件不支持 DoubleClick 事件。

#### 【案例 4-2】TextBox 控件和 Button 控件的使用

**要求:** 在【案例 4-1】的基础上, 添加三个 Label 控件、两个 TextBox 控件和两个 Button 控件。当用户输入“用户名”和“口令”后, 单击【确定】按钮, 若输入的用户名和口令正确, 则显示“用户名和口令正确”, 否则显示“用户名或口令不正确”; 单击【结束】按钮会关闭窗口, 如图 4-4 所示。

步骤 1: 启动 Visual Studio 2008, 打开“Library”项目。

步骤 2: 在窗体上添加三个 Label 控件、两个 TextBox 控件和两个 Button 控件。控件名取默认值。

步骤 3: 设置 Label 控件的属性。Label2 的 Text 属性设定为“用户名:”, Label3 的 Text 属性设定为“口令:”。

步骤 4: 设定 Button 控件的属性。Button1 的 Text 属性设定为“确定”, Button2 的 Text 属性设定为“结束”。

步骤 5: 控件的其他属性取默认值。

步骤 6: 调整控件的大小和位置, 效果如图 4-5 所示。



图 4-4 【案例 4-2】结果



图 4-5 界面设计

步骤 7: 为【确定】和【结束】按钮编写事件代码。

【结束】按钮的 Click 代码如下:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Me.Close()
End Sub
```

【确定】按钮的 Click 代码如下:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If TextBox1.Text = "向隅" And TextBox2.Text = "12312" Then
        Label4.Text = "用户名和口令输入正确"
    Else
        Label4.Text = "用户名或口令输入不正确"
    End If
End Sub
```

## 5. RadioButton、GroupBox 控件和 CheckBox 控件


### (1) RadioButton 控件和 GroupBox 控件

RadioButton 控件又称为单选按钮控件。在 VB.NET 的 Windows 应用程序设计中, 它通常与 GroupBox 控件组合, 用于建立项目的成组选择界面。

#### 1) RadioButton 控件

RadioButton 控件 (单选按钮控件) 用来让用户在一组相关的选项中选择一项, 因此单选按钮控件总是成组出现的。通常将多个 RadioButton 控件放在一个 GroupBox 控件内组成一组,



当某个单选按钮控件被选中时，该控件的圆圈中间出现一个小圆点，表示该项被选中。工具箱中的 RadioButton 控件图标为  RadioButton。

RadioButton 控件的常用属性如表 4-6 所示。

表 4-6 RadioButton 控件的常用属性

属性	作用
Appearance	指定单选按钮控件的外观。Normal: 外观为 Windows 标准单选按钮, Button: 外观显示为切换式按钮
Text	设置控件的标题文字
TextAlign	设置控件中标题文字的对齐方式
CheckAlign	设置控件中文本与单选按钮的位置
Checked	设置单选按钮是否为选中状态。False: 没选中, True: 选中并触发 Click 事件
AutoCheck	设置单选按钮在单击时是否被自动选定。True: 自动选定, False: 不选定
BackColor	设置控件的背景颜色
BackgroundImage	设置控件所包含文字或图形的前景色
ForeColor	设置控件背景图像
Font	设置控件上文字的字体
Image	设置控件的前景图像
Visible	设置按钮控件是否可见
FlatStyle	设置控件的样式。默认为 Standard

RadioButton 控件的常用事件如下：

Click 事件：当用户单击该按钮时，将触发 Click 事件。

CheckedChanged 事件：当 Checked 属性的值更改时，触发该事件。

## 2) GroupBox 控件

GroupBox 控件又称为分组框，它用于为其他控件提供可识别的分组。

GroupBox 控件的常用属性如表 4-7 所示。

表 4-7 GroupBox 控件的常用属性

属性	作用
Text	设置 GroupBox 控件的标题文字
BackColor	设置控件的背景色
BackgroundImage	设置控件的背景图像
ForeColor	设置控件所包含文字或图像的前景色
Font	设置控件标题文字的字体
Image	设置控件的前景图像
Visible	设置控件是否可见
FlatStyle	设置控件的样式。默认为 Standard

### 【实例 4-3】单选按钮控件和分组框控件的应用

要求：在窗体中使用分组框控件和单选按钮控件，当选中单选按钮选项时，标签控件上的文本相应地发生变化。

步骤 1：启动 Visual Studio 2008，新建一个项目。

步骤 2：将窗口的 Text 属性设定为“单选按钮控件的应用”。

步骤 3：在窗口中添加一个 Label 控件和三个分组框控件，并在每个分组框中添加三个 Radio-Button 控件。

步骤 4：分别设定 Label 控件和三个分组框中 RadioButton 控件的 Text 属性值及其他属性值，如图 4-6 所示。

步骤 5：代码编写：

```
Public Class Form1
    Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged
        '选中“黑体”单选按钮代码
        If RadioButton1.Checked Then
            Label1.Font = New Font("黑体", Label1.Font.Size)
        End If
    End Sub
    Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged
        '选中“楷体”单选按钮代码
        If RadioButton2.Checked Then
            Label1.Font = New Font("楷体_GB2312", Label1.Font.Size)
        End If
    End Sub
    Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles RadioButton3.CheckedChanged
        '选中“幼圆”单选按钮代码
        If RadioButton3.Checked Then
            Label1.Font = New Font("幼圆", Label1.Font.Size)
        End If
    End Sub
    Private Sub RadioButton4_CheckedChanged(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles RadioButton4.CheckedChanged
        '选中“10”单选按钮代码
        If RadioButton4.Checked = True Then
            Label1.Font = New Font(Label1.Font.FontFamily, 10)
        End If
    End Sub
    Private Sub RadioButton5_CheckedChanged(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles RadioButton5.CheckedChanged
        '选中“12”单选按钮代码
```




图 4-6 界面设计

```
    If RadioButton5.Checked = True Then
        Label1.Font = New Font(Label1.Font.FontFamily, 12)
    End If
End Sub
Private Sub RadioButton6_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton6.CheckedChanged
    '选中“14”单选按钮代码
    If RadioButton6.Checked = True Then
        Label1.Font = New Font(Label1.Font.FontFamily, 14)
    End If
End Sub
Private Sub RadioButton7_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton9.CheckedChanged
    '选中“红色”单选按钮代码
    If RadioButton7.Checked = True Then
        Label1.ForeColor = Color.Red
    End If
End Sub
Private Sub RadioButton8_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton8.CheckedChanged
    '选中“蓝色”单选按钮代码
    If RadioButton8.Checked = True Then
        Label1.ForeColor = System.Drawing.ColorTranslator.FromOle( RGB(0, 0, 255))
    End If
End Sub
Private Sub RadioButton9_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton7.CheckedChanged
    '选中“绿色”单选按钮代码
    If RadioButton9.Checked = True Then
        Label1.ForeColor = System.Drawing.ColorTranslator.FromOle( RGB(0, 255, 0))
    End If
End Sub
End Class
```

步骤6: 按 F5 键运行程序, 用户在三个选项组进行选择时, 其效果如图 4-7 所示。

### (2) CheckBox 控件

CheckBox 控件又称为复选框, 它与单选按钮一样, 也提供一组选项供用户选择。但它与单选按钮有所不同, 每个复选框都是一个单独的选项, 用户既可以选择它, 也可以不选择它, 不存在互斥的问题, 可以同时选择多项。工具箱中 CheckBox 控件图标为  CheckBox。

复选框的默认名称为 CheckBox1、CheckBox2, ..., 等等。

CheckBox 控件的常用属性如表 4-8 所示。



图 4-7 【案例 4-3】运行效果

表 4-8 CheckBox 控件的常用属性

属性	作用
Text	设置复选框显示的文本
TextAlign	指定在控件上显示的文本的对齐方式
CheckAlign	设置控件中文本与复选框的位置
ThreeState	设置复选框是三种状态还是两种状态。False: 两种, True: 三种
BackColor	设置控件的背景色
BackgroundImage	设置控件的背景图像
ForeColor	设置控件所包含文字或图像的前景色
Font	设置控件标题文字的字体
Image	设置控件前景图像
Visible	设置按钮控件是否可见
FlatStyle	设置复选框的样式。默认为 Standard
Checked	设置复选框是否被选中。如果复选框被选中, 则其 Checked 属性为 True, 并且触发 Click 事件
Checkstate	设置控件的状态。Unchecked: 没选中, Checked: 选中 (出现“√”), Indeterminate: 选中 (出现“■”)

CheckBox 控件的常用事件如下:

Click 事件: 当用户单击该控件时, 将触发 Click 事件。

CheckedChanged 事件: 当 Checked 属性的值更改时, 触发该事件。

#### 【实例 4-4】复选框的应用

要求: 在窗体中使用分组框控件和复选框控件, 当选中复选框选项时, 标签控件上的文本相应地发生变化。

步骤 1: 启动 Visual Studio 2008, 新建一个项目。

步骤 2: 将窗口的 Text 属性设定为“复选框的应用示例”。

步骤 3: 在窗口中添加一个 Label 控件和一个分组框控件, 并在分组框中添加四个 CheckBox 控件。

步骤 4: 分别设定 Label 控件和分组框中 CheckBox 控件的 Text 属性值及其他属性值, 如图 4-8 所示。



图 4-8 界面设计

步骤 5: 编写 CheckBox 控件的事件代码:


```
Public Class Form1
    Public style As FontStyle
    ' “加粗” 复选框选中事件代码
    Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox1.CheckedChanged
        If CheckBox1.Checked = True Then
            style = style + FontStyle.Bold
        Else
            style = style - FontStyle.Bold
        End If
        Label1.Font = New Font(Font.Name, Font.Size, style)
    End Sub
    ' “倾斜” 复选框选中事件代码
    Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles CheckBox2.CheckedChanged
        If CheckBox2.Checked = True Then
            style = style + FontStyle.Italic
        Else
            style = style - FontStyle.Italic
        End If
        Label1.Font = New Font(Font.Name, Font.Size, style)
    End Sub
    ' “删除线” 复选框选中事件代码
    Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles CheckBox3.CheckedChanged
        If CheckBox3.Checked = True Then
            style = style + FontStyle.Strikeout
        Else
            style = style - FontStyle.Strikeout
        End If
        Label1.Font = New Font(Font.Name, Font.Size, style)
    End Sub
    ' “下划线” 复选框选中事件代码
    Private Sub CheckBox4_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles CheckBox4.CheckedChanged
        If CheckBox4.Checked = True Then
            style = style + FontStyle.Underline
        Else
            style = style - FontStyle.Underline
        End If
        Label1.Font = New Font(Font.Name, Font.Size, style)
    End Sub
End Class
```

步骤 6: 按 F5 键运行程序, 选中各复选框, 结果如图 4-9 所示。



图 4-9 【案例 4-4】运行结果

## 6. ListBox 控件

ListBox 控件称为列表框，它显示一个项目列表供用户选择。在列表框中，用户既可以一次选择一项，也可以一次选择多项。工具箱中 ListBox 控件图标为  ListBox。

### (1) 常用属性

ListBox 控件的常用属性如表 4-9 所示。

表 4-9 ListBox 控件的常用属性

属性	作用
HorizontalScrollbar	设置 ListBox 控件是否为超出 ListBox 右边缘的项显示水平滚动条
Items	设置列表框中的列表项
ItemHeight	设置列表框中项的高度
MultiColumn	设置 ListBox 控件是否应在列中水平显示
SelectionMode	指示 ListBox 控件提供单项选择、多重选择或是不可选择。默认值为 One，即单项选择
Sorted	指示对 ListBox 控件中的项是否按字母顺序排序
Text	此属性只在运行时可用，Text 属性总是对应用户在运行时选定的列表项目。通常，获取当前选定项目值的最简单方法是使用 Text 属性

### (2) 常用事件

① SelectedIndexChanged 事件。该事件在 SelectedIndex 属性更改后发生。可以为该事件创建事件处理程序，以确定 ListBox 控件中选定的索引何时更改。这在需要根据 ListBox 控件中的当前选定内容显示其他控件中的信息时非常有用。可以使用该事件的事件处理程序来加载其他控件中的信息。

② Click 事件。该事件在单击 ListBox 控件时发生。通常，在 ListBox 控件需要和其他控件协同工作时，可以使用该事件创建事件处理程序。

### (3) 常用方法

#### 1) Items.Insert 方法

使用该方法可以在 ListBox 控件内将项插入列表框的指定索引处，其语法如下：

```
ListboxName.Items.Insert(index As Integer, item As Object)
```

其中 ListboxName 是列表框的名称，item 是添加到列表中的一个对象，index 指定在列表

中插入新项目的位置。`index` 为 0 表示第一个位置。如果 `ListBox` 的 `Sorted` 属性设置为 `True`，则该项添加到排序列表内的正确位置处，而不考虑 `index` 参数中指定的值。

任何时候都可以使用 `Items.Insert` 方法添加项，包括可动态响应用户操作的添加项目。下列代码将“武汉”、“上海”、“广州”和“沈阳”添加到名为 `ListBox1` 的列表框中：

```
Private Sub Form1_Load (ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    ListBox1.Items.Insert(0, "武汉")
    ListBox1.Items.Insert(1, "上海")
    ListBox1.Items.Insert(2, "广州")
    ListBox1.Items.Insert(3, "沈阳")
End Sub
```

### 2) Items.Add 方法

该方法表示向 `ListBox` 控件的列表框中添加项，其语法如下：

```
ListboxName.Items.Add(item As Object)
```

其中 `ListboxName` 是列表框的名称，`item` 是一个对象，它表示要添加到集合中的项。如果 `ListBox` 的 `Sorted` 属性设置为 `true`，则该项将按字母顺序插入到列表中。否则，在列表的结尾处插入项。也可以使用下列代码将“武汉”、“上海”、“广州”和“沈阳”添加到名为 `ListBox1` 的列表框中：

```
Private Sub Form1_Load (ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles MyBase.Load
    ListBox1.Items.Add("武汉")
    ListBox1.Items.Add("上海")
    ListBox1.Items.Add("广州")
    ListBox1.Items.Add("沈阳")
End Sub
```

### 3) Items.Remove 方法

使用 `Items.Remove` 方法可以从列表中删除项，其语法如下：

```
ListboxName.Items.Remove(value As Object)
```

`value` 是一个对象，表示要从集合中移除的项。从列表中移除项时，列表中后面的项的索引将发生变化。有关移除的项的所有信息均会被删除。可以使用此方法通过指定要从列表中移除的实际项来从列表中移除特定项。

### 4) Items.Clear 方法

要删除列表框中的所有项目，应使用 `Clear` 方法：

```
ListboxName.Items.Clear()
```

### 5) FindString 方法

使用 `FindString` 方法可以查找 `ListBox` 中以指定字符串开始的第一个项。其语法如下：

```
value =ListBoxName.FindString(s)
```

其中，`value` 为 `Integer` 类型值，表示找到的第一个项的从零开始的索引，如果找不到匹配项，则返回 `ListBox.NoMatches`，即值为 -1。`s` 是要搜索的文本。

由此方法执行的搜索不区分大小写。


【实例 4-5】列表框控件的应用

要求：在窗体中使用 ListBox 控件，当在列表框选中一项时标签控件上的文本相应地发生变化，如图 4-10 所示。

步骤 1：启动 Visual Studio 2008，新建一个项目。

步骤 2：将窗口的 Text 属性设定为“ListBox 控件应用实例”。

步骤 3：在窗口中添加一个 Label 控件和一个 ListBox 控件。

步骤 4：设置 ListBox 控件的 Items 属性。在设计窗口选择 ListBox 控件，打开其【属性】窗口，单击 Items 右侧按钮，打开【字符串集合编辑器】对话框，输入字符串，如图 4-11 所示。

步骤 5：单击【确定】按钮，设计界面如图 4-12 所示。

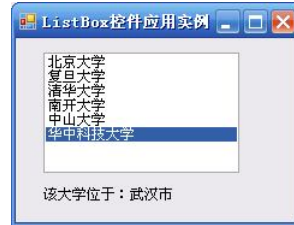


图 4-10 【案例 4-5】的运行结果

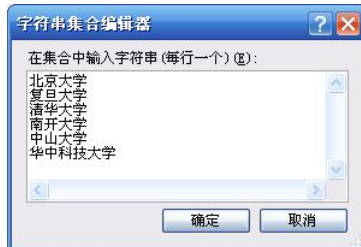


图 4-11 【字符串集合编辑器】对话框



图 4-12 设计界面


步骤 6：为 ListBox 控件的 SelectedIndexChanged 事件编写代码。

```
Private Sub ListBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ListBox1.SelectedIndexChanged
    With ListBox1
        Select Case .Text
            Case "北京大学"
                Label1.Text = "该大学位于：北京市"
            Case "复旦大学"
                Label1.Text = "该大学位于：上海市"
            Case "清华大学"
                Label1.Text = "该大学位于：北京市"
            Case "南开大学"
                Label1.Text = "该大学位于：天津市"
            Case "中山大学"
                Label1.Text = "该大学位于：广州市"
            Case "华中科技大学"
                Label1.Text = "该大学位于：武汉市"
        End Select
    End With
End Sub
```

步骤 7：按 F5 键运行程序，当选中“华中科技大学”时，标签中显示“该大学位于：武汉市”信息，如图 4-10 所示。



## 7. NumericUpDown 控件

NumericUpDown 控件用来让用户通过单击向上和向下箭头、按向上和向下键或在控件的文本框部件中键入一个数字来增大和减小数字。单击向上键时，值向最大值方向移动；单击向下键时，值向最小值方向移动。NumericUpDown 控件用于许多 Windows 控制面板应用程序，例如在为音乐播放器应用程序创建音量控件时。工具箱中 NumericUpDown 控件的图标为  NumericUpDown。

### (1) 主要属性

NumericUpDown 控件的主要属性如表 4-10 所示。


表 4-10 NumericUpDown 控件的常用属性

属性	作用
Value	设置该控件中当前选择的数字
Maximum	设置该控件的最大值。默认值为 100
Minimum	设置该控件的最小值。默认值为 0
Increment	设置用户单击向上或向下箭头时数字的调整量。默认值为 1

### (2) 主要方法

- UpButton: 增加数字显示框（也称作 up-down 控件）中的值。
- DownButton: 减小数字显示框（也称作 up-down 控件）中的值。

## 8. ComboBox 控件

ComboBox 控件称为组合框控件，它是文本框和列表框的组合，它对应于 System.Windows.Forms.ComboBox 类。用户可以在组合框中输入信息，也可以从下拉列表中选择预设的项。工具箱中 ComboBox 控件的图标为  ComboBox。

### (1) 常用属性

ComboBox 控件的常用属性如表 4-11 所示。

表 4-11 ComboBox 控件的常用属性

属性	作用
Items	表示该 ComboBox 中所包含项的集合，用于获取对当前存储在 ComboBox 中的项列表的引用。通过此引用，可以在集合中添加项、移除项和获得项的计数。其属性值设置及使用方式跟 ListBox 的 Items 属性一致
Text	表示当前 ComboBox 中显示的文本内容。在运行时无论向控件的文本框部分输入了什么文本，Text 属性都与这个文本相对应。它可以是选定的列表选项，或者是用户在文本框中输入的字符串
DropDownStyle	确定要显示的组合框的样式。可以输入一值，该值提供以下功能：将 DropDownStyle 属性设置为 Simple 时，显示样式为简单的下拉列表，显示列表中所有项，必须将列表框绘制得足够大，当选项数超过可显示的限度时将自动插入一个垂直滚动条；将 DropDownStyle 属性设置为 DropDownList 时，显示样式为下拉列表框，文本部分不可编辑，并且必须单击箭头才能查看下拉列表框；默认值为 DropDown，显示样式为下拉列表框，文本部分可编辑，并且用户必须单击箭头才能查看列表

续表

属性	作用
Sorted	指示对 ComboBox 中的项是否按字母顺序排序, 默认值为 false。若指定要按字母顺序添加到列表中的项目, 则将 Sorted 属性设置为 true
SelectedIndex	想要知道组合框中选定项目位置, 可以由 SelectedIndex 属性得到。该属性设置或返回控件中当前选定项目的索引值, 而且只在运行时有效。对组合框的 SelectedIndex 属性进行设置时也会触发控件的 Click 事件。若选定第一个(顶端)项目, 则属性值为 0; 选定的下一个项目属性值为 1, 依次类推。如果未选定项目, 或者用户在组合框中输入选项(样式 0 或 1) 而不在列表中选择现有的项目, 则 SelectedIndex 为 -1
Items.Count	为了返回组合框中的项目数, 应使用 Items.Count 属性

## (2) 事件和方法

ComboBox 的常用事件 SelectedIndexChanged 事件和 Click 事件与 ListBox 相应的事件响应基本一致, 可以参考上一小节中对这些事件的介绍。

ComboBox 的常用方法 Items.Insert 方法、Items.Add 方法、Items.Remove 方法、Items.Clear 方法以及 FindString 方法, 使用语法和方式与 ListBox 相应的方法也基本一致, 同样可以参考上一小节中对这些方法的介绍。

### 【案例 4-6】组合框控件的应用

要求: 使用组合框、数字显示框和列表框, 实现将输入学生信息在列表框中按顺序列出来。效果如图 4-13 所示。

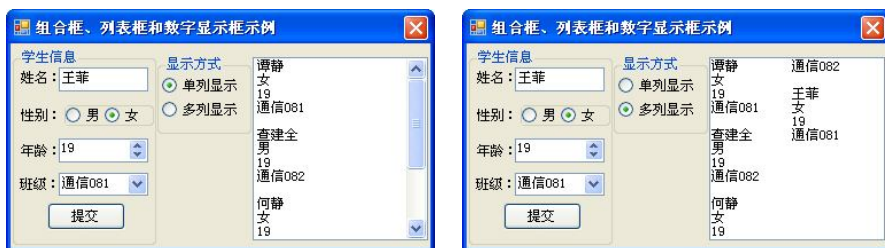


图 4-13 【案例 4-6】效果

步骤 1: 启动 Visual Studio 2008, 新建一个项目。

步骤 2: 将窗口的 Text 属性设定为“组合框、列表框和数字显示框示例”。

步骤 3: 按图 4-13 添加各控件。

步骤 4: 设置显示班级的 ComboBox 控件的 Items 属性。打开其【字符串集合编辑器】对话框, 输入班级名称, 如图 4-14 所示。

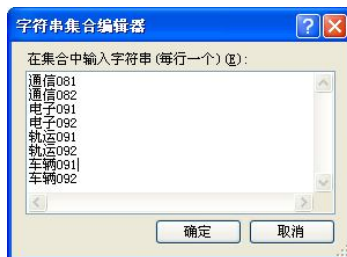



图 4-14 【字符串集合编辑器】对话框

步骤 5: 编写【提交】按钮和【显示方式】分组框事件代码:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    ListBox1.Items.Add(TextBox1.Text)
    If RadioButton1.Checked Then
        ListBox1.Items.Add("男")
    Else
        ListBox1.Items.Add("女")
    End If
    ListBox1.Items.Add(CStr(NumericUpDown1.Value))
    ListBox1.Items.Add(ComboBox1.Text)
    ListBox1.Items.Add(" ")
End Sub
Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton3.CheckedChanged
    If RadioButton3.Checked Then
        ListBox1.MultiColumn = False
    End If
End Sub
Private Sub RadioButton4_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton4.CheckedChanged
    If RadioButton4.Checked Then
        ListBox1.MultiColumn = True
    End If
End Sub
```

步骤 6: 按 F5 键运行程序, 结果如图 4-13 所示。

## 9. ProgressBar 控件

ProgressBar 控件俗称进度条控件, 它是一个水平放置的指示器, 显示相应数目的矩形来指示操作的进度。操作完成时, 进度栏被填满。进度栏通常用于帮助用户了解等待一项长时间的操作(如加载大文件)完成所需的时间。工具箱中该控件的图标为  ProgressBar。

ProgressBar 控件以三种样式中的一种指示较长操作的进度 (Style 属性值):

- Block: 从左向右分步递增的分段块。
- Continuous: 从左向右填充的连续栏。
- Marquee: 以字幕方式在 ProgressBar 中滚动的块。

(1) 常用属性

表 4-12 ProgressBar 控件常用属性

属性	作用
Style	确定显示的 ProgressBar 的样式。可选 Block、Continuous、Marquee
Maximum	设置 ProgressBar 正使用的上限范围
Minimum	设置 ProgressBar 正使用的下限范围
Value	表示应用程序在完成操作的过程中的进度
Step	当调用 PerformStep()方法时, 控件当前值的增量

## (2) 常用方法和事件

- **PerformStep:** 按照 Step 属性的数量增加进度栏的当前位置。
- **Increment:** 按指定的数量增加进度栏的当前位置。当 Style 属性设定为 Marquee 时不必调用 Increment, 否则将引发 InvalidOperationException。

## 10. Timer 控件

Timer (定时) 控件也称为定时器或计数器, 是按一定时间间隔周期性地自动触发事件的控件。在程序运行时, 定时器是不可见的。定时器控件的默认名称为 Timer1、Timer2 等。

## (1) 常用属性

Timer 控件的常用属性如表 4-13 所示。

表 4-13 Timer 控件的常用属性

属性	作用
Enabled	设置定时器是否正在运行。True: 运行 (默认值), False: 不运行
Interval	设置定时器开始计时之间的时间 (以毫秒为单位)

## (2) 常用方法

- **Start:** 启动定时器。
- **Stop:** 停止定时器。

## (3) 常用事件

- **Tick:** 当定时器处于启动状态时, 每隔一个 Interval 时间, 触发一次该事件。

## 【案例 4-7】Timer 控件的应用

要求: 利用 Timer 控件的 Tick 事件建立一个窗体标题在窗体中移动的动画。

步骤 1: 启动 Visual Studio 2008, 新建一个项目。

步骤 2: 将工具箱的组件选项中 Timer 控件  拖到窗体中, 如图 4-15 所示。

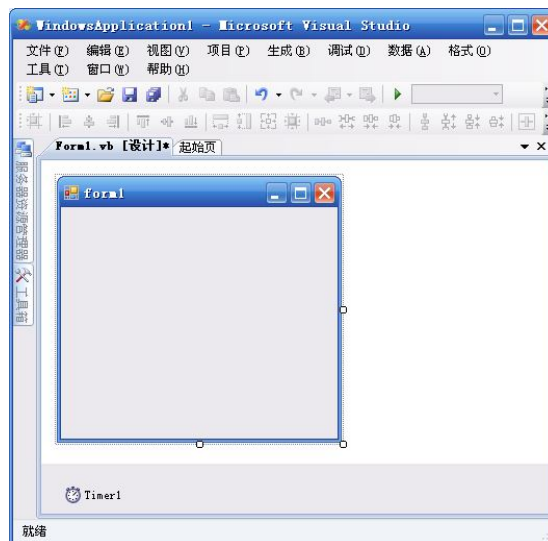


图 4-15 界面设计

步骤3: 设计 Tick 事件代码:

```
Public i As Integer = 30
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        If i <= 0 Then
            i = 30
        Else
            i = i - 1
        End If
        Me.Text = Space(i) + "timer 事件事例"
    End Sub
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Timer1.Enabled = True
    End Sub
```

## 11. PictureBox 控件

PictureBox 控件常用于图形设计和图像处理应用程序, 又称为图形框。可加载到控件上的图像文件格式有: 位图文件 (.bmp)、图标文件 (.ico)、图元文件 (.wmf)、.jpg 和 .gif 文件。

### (1) 常用属性

PictureBox 控件的常用属性如表 4-14 所示。

表 4-14 PictureBox 控件的常用属性

属性	作用
Image	设置控件显示的图形
Visible	设置控件是否显示
Height	设置控件的高度
Width	设置控件的宽度
SizeMode	设置图像的显示方式
Size	设置控件的高度和宽度
BackColor	设置控件的背景色
BackgroundImage	设置在控件中显示的背景图像
BorderStyle	设置控件边框的样式

### (2) 控件的常用方法

PictureBox 控件的常用方法如表 4-15 所示。

表 4-15 PictureBox 控件的常用方法

方法	作用
CreateGraphics	为控件创建一个 Graphics 对象
Dispose	释放由 Component 占用的资源

续表

方法	作用
Hide	隐藏控件
Refresh	强制控件的工作区无效并立即重绘
Show	向用户显示控件

## (3) 控件的常用事件

PictureBox 控件的常用事件如表 4-16 所示。

表 4-16 PictureBox 控件的常用事件

事件	作用
Click	单击 PictureBox 控件时发生
DoubleClick	双击 PictureBox 控件时发生
MouseEnter	当鼠标指针移入控件上时发生
Move	控件被移动时发生
Paint	重绘控件时发生

## 项目小结

本项目包含两个子任务：

任务一：简要介绍了工具箱的组成及添加控件的方法。控件有许多属性，其属性既可在【属性】窗口设置，也可在程序运行时通过代码设置。窗体上的控件可使用对齐控件快速对齐。

任务二：详细介绍了 Visual Basic 2008 中常用控件的属性、方法和事件，并通过案例给出其使用技巧。

## 习题四

### 一、填空题

1. 控件的名称相同但索引号不同，称为\_\_\_\_\_。
2. 设置定时器的时间间隔的属性是\_\_\_\_\_。
3. 在程序中关闭定时器控件 Timer1 的方法是\_\_\_\_\_。
4. 要使列表框中的选项功能同时选中多个，应设置列表框的\_\_\_\_\_属性。
5. 已知窗体中有一个图片框，名为 Pic1，现在要把图形文件“D:\a1.bmp”显示在该控件中，使用的语句是\_\_\_\_\_。
6. 为了能使文本框自动调整大小以显示全部文本内容，应把标签的\_\_\_\_\_属性设置为

True。

7. 给命令按钮或者菜单项设置快捷访问键应该使用的符号是\_\_\_\_\_。
8. 窗体的窗口状态 `WindowState` 属性有 3 种, 分别是\_\_\_\_\_、最小化和最大化状态。
9. 当程序运行装入窗体时, 最先触发的是\_\_\_\_\_事件。
10. 补充下面的程序代码, 使单击窗体上的命令按钮 `Command1` 时, 该窗体高度缩小一半, 宽度增加一倍。

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) _  
Handles Me.Load  
    Me.Height = Me.Height / 2  
    Me.Width = _____  
End Sub
```

## 二、选择题

1. 复选框被选中时, `Checked` 属性的值为 ( )。  
A. True            B. False            C. 0                D. 1
2. 将文本框控件的 ( ) 属性设置为 `True`, 可以正常显示文本但不可编辑。  
A. `ReadOnly`      B. `Enabled`        C. `MultiLine`      D. `Visible`
3. 在界面设计时 `Label` 控件的各属性使用默认值, 则标签控件的 `Name` 属性和 `Text` 属性的默认值分别为 ( )。  
A. `Label1`、`Label1`                      B. `Label`、`Label1`  
C. `Label1`、`Label`                        D. `Label`、`Label`
4. 文本框既可以输入单行文本, 又可以输入多行文本, 若使文本框能输入多行文本, 则应将 ( ) 属性设置为 `True`。  
A. `Text`            B. `MultiLine`      C. `ReadOnly`        D. `Visible`
5. 表示控件在窗体上位置的属性是 ( )。  
A. `Size`            B. `Location`        C. `Visible`            D. `Locked`
6. 下面关于文本框的说法, 错误的是 ( )。  
A. `Text1.Text="Hello"`, 在文本框 `Text1` 中显示 Hello  
B. `Text1.Locked=true`, 设置文本框 `Text1` 不能编辑  
C. `Text1.Visible=False`, 设置文本框 `Text1` 不可见  
D. `Text1.passwordchar="#"`, 设置文本框 `Text1` 接收字符以 “#” 显示
7. 暂时关闭定时器 (`Timer`) 控件, 应将 ( ) 属性设置为 `False`。  
A. `Timer`            B. `Enabled`            C. `Interval`            D. `Visible`
8. ( ) 属性能够改变窗体的边框线。  
A. `Font`            B. `FormBorderStyle`    C. `ForeColor`            D. `Cursor`
9. 组合框中的 `FlatStyle` 属性值确定了组合框的类型和显示方式, 以下选项不属于 `FlatStyle` 属性值的是 ( )。  
A. `Standard`        B. `System`            C. `Popup`                D. `Style`

10. 窗体上有三个文本框 Text1、Text2 和 Text3，有一个命令按钮 Button1，设文本框 Text1 中内容为 11，文本框 Text2 中内容为 12，执行 `TextBox3.Text = TextBox1.Text + TextBox2.Text` 语句后 Text3.text 中的结果为（ ）。

A. 11                      B. 12                      C. 23                      D. 1112

11. 可以唯一标识控件数组中的每一个控件属性的是（ ）。

A. Name                    B. Text                    C. Visible                    D. Enabled

12. 运行 Visual Basic 程序的快捷键为（ ）。

A. F4                      B. F5                      C. F1                      D. F2