

Lab7 WinForm 窗体及控件

学习目标：

- 1.了解 WinForm 程序开发步骤
- 2.掌握选择类控件的应用
- 3.掌握定时器控件的应用

一、WinForm 应用程序开发步骤

- (1) 创建窗体：创建窗体才能容纳其他各种界面对象；
- (2) 创建控件：创建组成界面的各种元素，例如按钮、文本框等；
- (3) 指定布局：根据具体需要排列它们的位置关系；
- (4) 响应事件：定义图形用户界面的事件和各界面元素对不同事件的响应，从而实现图形界面与用户的交互功能。

二、选择类控件

在使用文本框构建用户的输入界面时，虽然可以检查或验证用户输入的有效性，但仍然不能完全确保用户的输入是系统所期望的数据。为此，必须设计只需通过选择即可完成数据输入的操作界面。.NET Framework 为 Windows 窗体提供了丰富的选择类控制，选择类控件包括单选按钮控件（RadioButton）、复选框控件（CheckBox）、列表框控件（ListBox）以及下拉框控件（ComboBox）等。

1. RadioButton 控件

RadioButton，即单选按钮控件，是 System.Windows.Forms.RadioButton 类提供的，主要用于将一个或多个选项列出，让用户从中选择一项。RadioButton 控件的常用属性和常用事件如表所示。

RadioButton 控件的常用属性和事件

类别	名称	用途
属性	Name	指示代码中用来标识该对象的名称
	Text	显示的文本
	Checked	指示单选按钮是否选中
	AutoSize	指定控件是否自动调整自身的大小以适应其内容的大小
事件	CheckedChanged	每当“Checked”属性更改值时发生

2. CheckBox 控件

CheckBox，即复选框控件，是 System.Windows.Forms.CheckBox 类提供的，主要用于将一个或多个选项列出，让用户从中选择一项或多项。当某一个选项被选中后，其左边的小方框会显示打下一个勾。CheckBox 控件和 RadioButton 控件具有一个相似的功能，都允许用户从选项列表中进行选择。CheckBox 控件允许用户选择多个选项，而 RadioButton 控件用来构造相互排斥的选项组，并且只允许用户从中选择一个。CheckBox 控件的常用属性和常用事件如表所示。

CheckBox 控件的常用属性和事件

类别	名称	用途
属性	Name	指示代码中用来标识该对象的名称
	Text	显示的文本
	Checked	指示单选按钮是否选中
	CheckState	用来设置或返回复选框的状态，有 3 种可能：Checked、Unchecked、Indeterminate

事件	CheckedChanged	每当“Checked”属性更改值时发生
	CheckStateChanged	每当“CheckState”属性更改时发生

3. ListBox 控件

ListBox，即列表框控件，是 System.Windows.Forms.ListBox 类提供的，主要用于将一个集合数据以列表框的形式显示给用户，供用户从中选择一项或多项。ListBox 控件有两种工作模式：单选模式和多选模式。当工作在单选模式时，列表框与单选按钮的功能相同；当工作在多选模式时，则与复选框功能相同。所不同的是，需要多个 RadioButton 控件或 CheckBox 控件才能构造一个选项组，而使用 ListBox 控件，则只需要一个就可以产生一个选项列表。ListBox 控件的常用属性和常用事件如表所示。

ListBox 控件的常用属性和事件

类别	名称	用途
属性	Name	指示代码中用来标识该对象的名称
	Enabled	指示是否启用该控件
	Items	列表框中的项目集合
	ItemHeight	列表框中项的高度
	SelectionMode	指示列表框将是单项、多项还是不可选择
事件	SelectedIndexChanged	SelectedIndex 属性值更改时发生

【注意】当 ListBox 的 SelectionMode 的属性值为 MultiSimple 或 MultiExtended 时，SelectedIndex 返回的是选中的最小索引，SelectedItem 返回的是选中的索引值最小的选项值。Items 本身也包含很多属性，如 Count 属性指示 Items 包含项的个数等。

4. ComboBox 控件

ComboBox，即下拉框控件，是 System.Windows.Forms.ComboBox 类提供的，主要用于将一个集合数据以下拉框显示给用户，供用户从中选择一项。ComboBox 控件的默认行为是显示一个可编辑文本框，该文本框具有一个隐藏的下拉列表。ComboBox 控件只支持单选，可替代 RadioButton 选项组。ComboBox 控件的常用属性和常用事件如表所示。

ComboBox 控件的常用属性和事件

类别	名称	用途
属性	Name	指示代码中用来标识该对象的名称
	Items	列表框中的项目集合
事件	SelectedIndexChanged	SelectedIndex 属性值更改时发生

【实例】设计一个注册界面。分别定义 RadioButton 控件显示性别，ComboBox 控件显示城市，CheckBox 控件显示技术方向，ListBox 控件显示兴趣爱好。当用户点击“注册”按钮时，如果填写的信息完整，提示“注册成功！”，否则，提示“请填写全部信息”。

【操作步骤】

(1) 新建一个窗体项目 Lesson8。

(2) 双击 Form1.cs，切换到设计视图，从工具栏中拖拽 6 个 Label 控件、2 个 TextBox 控件、2 个 RadioButton 控件、1 个 ComboBox 控件、3 个 CheckBox 控件、1 个 ListBox 控件和 1 个 Button 控件到窗体设计区，并根据图 4-14 调整控件大小和布局，并完成所有 Label 控件 Text 属性设置。

(3) 在窗体设计区中右击窗体 Form3 和每一个控件，设置窗体和控件的相关属性。表中列出了除 Label 以外的窗体及控件属性。

窗体及控件属性设置

窗体和控件	属性	属性值
Form3	Text	注册
textBox1	Name	txtName
textBox2	Name	txtPassword
	UseSystemPasswordChar	True
radioButton1	Name	rdMale
	Text	男
	Checked	True
radioButton2	Name	rdFemale
	Text	女
comboBox1	Name	city
	Items	北京/上海/哈尔滨/大连/深圳/广州
checkBox1	Name	net
	Text	.NET
checkBox2	Name	c
	Text	C
checkBox3	Name	java
	Text	JAVA
listBox1	Name	interest
	Items	旅行/看书/运动
button1	Name	btnRegister
	Text	注册

(4) 定义一个名为 `IsNullOrEmpty` 的方法，用于判断用户是否已输入全部注册信息，程序代码如下。

```
public bool IsNullOrEmpty(Control.ControlCollection controls)
{
    bool flag = false;
    foreach (Control item in controls)
    {
        //判断当前控件内容是否为空
        if (string.IsNullOrEmpty(item.Text))
        {
            flag = true;
        }
    }
    return flag;
}
```

以上代码的功能是, 判断窗体中所有控件内容是否为空, 如果某一个控件内容为空, 则返回true, 否则, 返回false。

(5) 双击“注册”按钮, 为其添加单击事件处理程序, 程序代码如下。

```
private void btnRegister_Click(object sender, EventArgs e)
{
    bool flag = IsNullOrEmpty(this.Controls);
}
```

```

        if (flag == true)
        {
            MessageBox.Show("请填写全部信息!");
        }
        else
        {
            MessageBox.Show("注册成功!");
        }
    }
}

```

以上代码的功能是，调用IsNullOrEmpty方法，实参为窗体中所有控件集合，方法返回值如果为true，表示窗体中某控件Text属性值为空，即用户填写注册信息不全，则提示用户“请填写全部信息!”；否则，提示用户“注册成功!”。

(6) 在解决方案资源管理器 Controls 项目中双击“Program.cs”文件，将 Main()方法中的最后一行代码修改如下。

```
Application.Run(new Form1());
```

(7) 编译并运行，运行结果如图所示。

三、定时器控件

1. 定时器控件

Timer，即定时器控件，是 System.Windows.Forms.Timer 类提供的，它是一个功能性的控件，没有用户界面，主要用于在程序中按照时间间隔产生定时消息，然后执行消息代码。Timer 控件的常用属性和常用事件如表所示。

Timer 控件的常用属性和事件

类别	名称	用途
属性	Name	指示代码中用来标识该对象的名称
	Interval	事件的频率（以毫秒为单位）
	Enabled	是否产生定时消息
事件	Tick	每当经过指定的时间间隔时发生

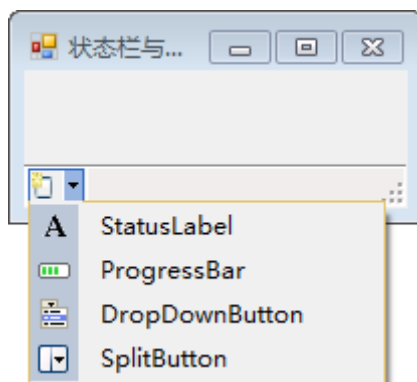
4.状态栏控件

StatusStrip，即状态栏控件，是 System.Windows.Forms.StatusStrip 类提供的，主要用于在应用程序中显示用户状态的简单信息，一般位于 Windows 窗体的底部。StatusStrip 控件的常用属性和常用事件如表所示。

StatusStrip 控件的常用属性和事件

类别	名称	用途
属性	Name	指示代码中用来标识该对象的名称
	Text	显示的文本
	Items	显示的项的集合
事件	ItemClicked	当单击项时发生

在状态栏中，可以使用文字或图标来显示应用程序的状态，也可以用一系列图标组成动画来表示正在进行某个过程。在窗体中添加 StatusStrip 控件后，通过 Items 属性或单击右边的三角形按钮，将弹出一个下拉框，可以为状态栏添加 StatusLabel、ProgressBar、DropDownButton、SplitButton 等窗格控件，如图所示。



这些控件的意义如表所示。

状态栏中可以添加的控件

名称	说明
StatusLabel	表示 StatusStrip 控件中的一个面板
Progressbar	显示进程的完成状态
DropDownButton	显示用户可以从中选择单个项的关联的选项
SplitButton	表示作为标准按钮和下拉菜单的一个组合控件

【实例】设计一个窗体，在状态栏中显示当前时间。

【操作步骤】

- （1）在 Lesson8 项目中新建一个窗体 Form2cs。
- （2）双击 Form2cs，切换到设计视图，从工具栏中添加 1 个 StatusStrip 控件和 1 个 Timer 控件到窗体设计区。
- （3）在窗体设计区中右击窗体 Form8 和每一个控件，设置窗体和控件的相关属性。表中列出了窗体及控件属性。

窗体及控件属性设置

窗体和控件	属性	属性值
Form8	Text	状态栏与定时器
statusStrip1: Items	Name	toolStripStatusLabel1
	Text	(空)
timer	Name	timer1
	Enabled	True
	Interval	1000

(4) 双击 Timer 控件，为其添加单击事件处理程序，程序代码如下。

```
private void timer1_Tick(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text = DateTime.Now.ToString();
}
```

以上代码的功能是，每1秒在状态栏中显示一次当前时间。

(5) 在解决方案资源管理器 Controls 项目中双击“Program.cs”文件，将 Main()方法中的最后一行代码修改如下。

```
Application.Run(new Form2());
```

(6) 编译并运行，运行结果如图所示。

