第 8 章　常见窗体控件的使用

主要内容

- C# 常用控件的使用
- 定制控件

8.1 Windows 控件

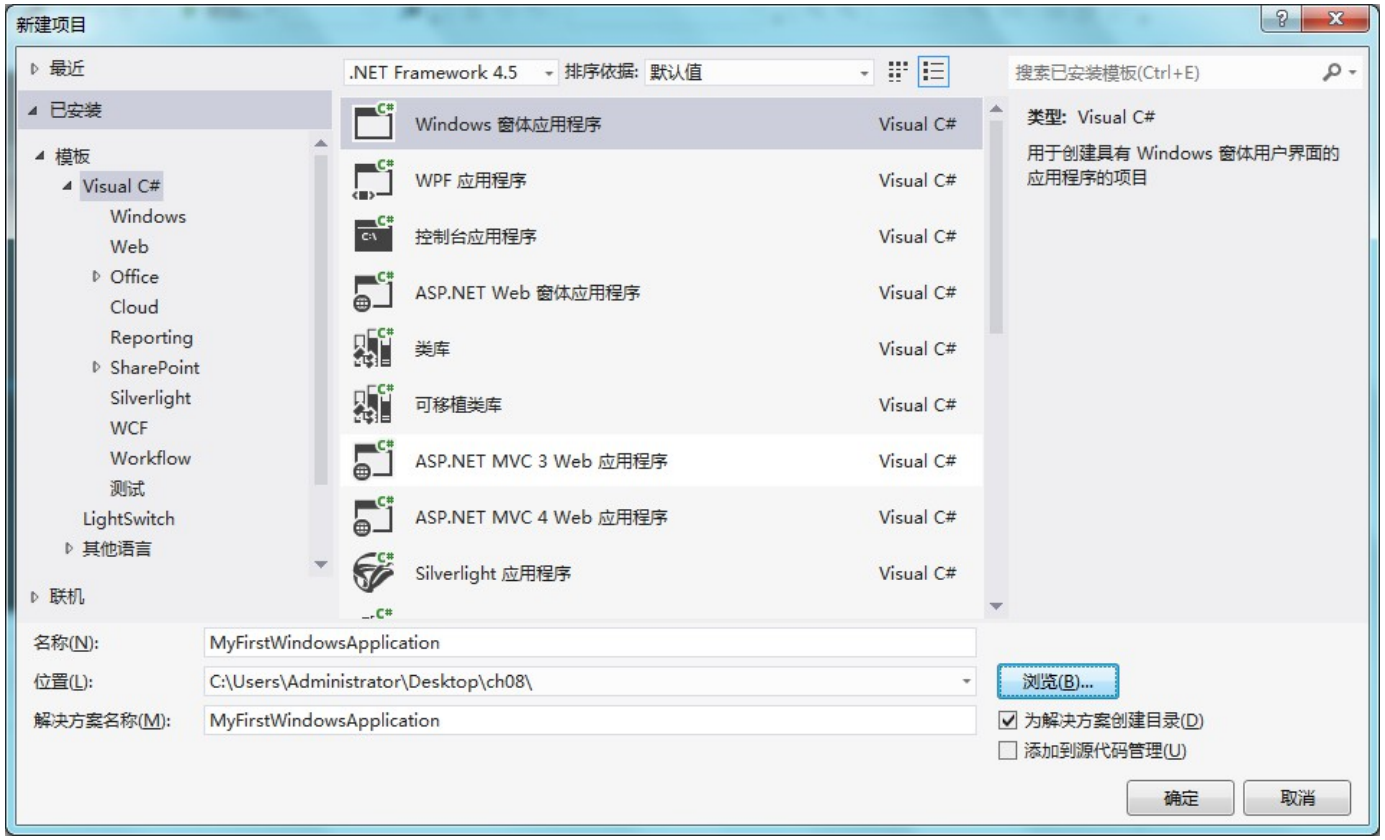
- 本章中我们将要介绍以下控件：
  - windows 窗体
  - Button 控件
  - TextBox 控件
  - RadioButton 与 CheckBox 控件
  - ListBox 控件
  - ListBox 控件
  - ListView 控件
  - ToolStrip 控件
  - StatusStrip 控件
  - StatusStrip 控件

windows 窗体

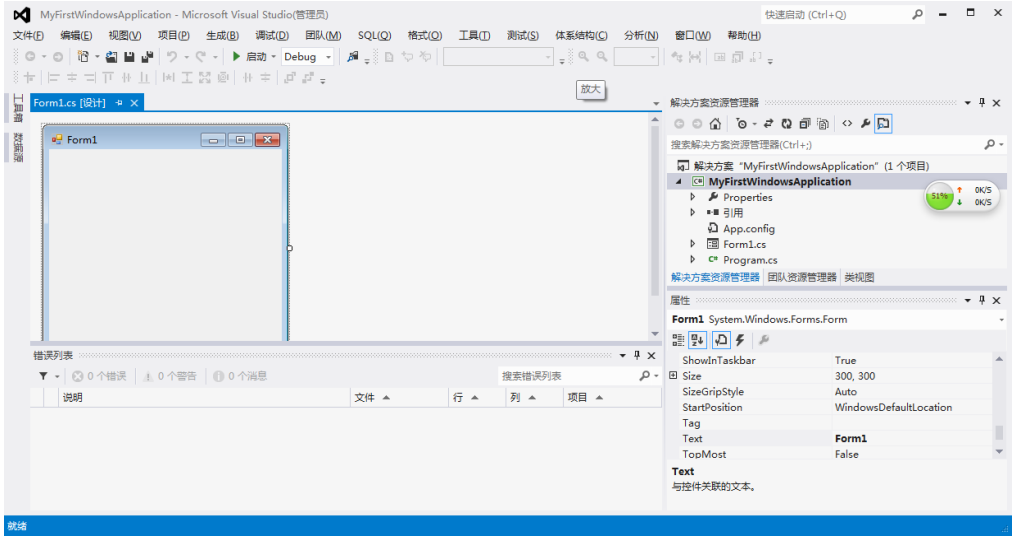
- 创建空白窗体

  创建空白窗体的操作如下：

  - （1）在 Visual Studio 2012 开发环境中，选择“文件”→“新建”→“项目”命令，弹出【新建项目】对话框。
  - （2）在左边选中【Visual C#】，右边选中【Windows 窗体应用程序】选项，然后在该对话框下方的【名称】文本框中，输入该项目的名称，如“MyFirstWindowsApplication”，在【位置】文本框中，输入保存该项目的位置，也可单击【浏览】按钮来选定保存位置，如下页图所示。


新建项目
最近
已安装
模板
Visual C#
Windows
Web
Office
Cloud
Reporting
SharePoint
Silverlight
WCF
Workflow
测试
LightSwitch
其他语言
联机
.NET Framework 4.5
排序依据: 默认值
搜索已安装模板(Ctrl+E)
Windows 窗体应用程序 Visual C#
WPF 应用程序 Visual C#
控制台应用程序 Visual C#
ASP.NET Web 窗体应用程序 Visual C#
类库 Visual C#
可移植类库 Visual C#
ASP.NET MVC 3 Web 应用程序 Visual C#
ASP.NET MVC 4 Web 应用程序 Visual C#
Silverlight 应用程序 Visual C#
类型: Visual C#
用于创建具有 Windows 窗体用户界面的应用程序的项目
名称(N): MyFirstWindowsApplication
位置(L): C:\Users\Administrator\Desktop\ch08\
解决方案名称(M): MyFirstWindowsApplication
浏览(B)...
为解决方案创建目录(D)
添加到源代码管理(U)
确定
取消

- 单击【确定】按钮，在 Visual Studio 2012 的编辑窗口中将显示一个空白窗体



设置窗体属性

- 在窗体上任意位置单击，选中要设置属性的窗体。
- 选择“视图”→“属性窗口”命令，在 Visual Studio 2012 开发窗口右侧就会出现一个属性窗口 。
- 在属性窗口中，列出了该窗体当前的各项属性，可以进行相应设置。不同的属性在属性窗口中的设置方式也有所不同，用户可以直接输入属性值 (例如 Size 、 Location 和 Text 等属性)、从下拉列表中选择一个值 (FormBorderStyle 和 StartPosition) 或者执行更加复杂的操作。

控件的公有属性、事件和方法

- .NET 中的大多数控件都派生于 System.Windows.Forms.Control 类。因此，在介绍其他各个控件之前，先来介绍一下 Control 这个类， Control 类实现了所有窗体交互控件的基本功能：处理用户键盘输入、处理消息驱动、限制控件大小等。
- Control 类的属性、方法和事件是所有窗体控件所共有的，在程序设计过程中经常会遇到，所以充分了解 Control 类的成员可以为以后的窗体编程打下坚实的基础。

1.Control 类的属性

名称	说明
AllowDrop	获取或设置一个值，该值指示控件是否可以接受用户拖放到它上面的数据。
Anchor	获取或设置控件绑定到的容器的边缘并确定控件如何随其父级一起调整大小。
BackColor	获取或设置控件的背景色。
BackgroundImage	获取或设置在控件中显示的背景图像。
BindingContext	获取或设置控件的 BindingContext。
Bottom	获取控件下边缘与其容器的工作区上边缘之间的距离（以像素为单位）。
Bounds	获取或设置控件(包括其非工作区元素)相对于其父控件的大小和位置(以像素为单位）。
CanFocus	获取一个值，该值指示控件是否可以接收焦点。
CanSelect	获取一个值，该值指示是否可以选中控件。
Capture	获取或设置一个值，该值指示控件是否已捕获鼠标。
CausesValidation	获取或设置一个值，该值指示控件是否会引起在任何需要在接收焦点时执行验证的控件上执行验证。
DataBindings	为该控件获取数据绑定。
DefaultBackColor	获取控件的默认背景色。
DefaultFont	获取控件的默认字体。
DefaultForeColor	获取控件的默认前景色。
Dock	获取或设置哪些控件边框停靠到其父控件并确定控件如何随其父级一起调整大小。

1.Control 类的属性

Enabled	获取或设置一个值，该值指示控件是否可以对用户交互作出响应。
Focused	获取一个值，该值指示控件是否有输入焦点。
Font	获取或设置控件显示的文字的字体。
ForeColor	获取或设置控件的前景色。
Handle	获取控件绑定到的窗口句柄。
HasChildren	获取一个值，该值指示控件是否包含一个或多个子控件。
Height	获取或设置控件的高度。
Left	获取或设置控件左边缘与其容器的工作区左边缘之间的距离(以像素为单位）。
Location	获取或设置该控件的左上角相对于其容器的左上角的坐标。
Margin	获取或设置控件之间的空间。
MaximumSize	获取或设置大小，该大小是 GetPreferredSize 可以指定的上限。
MinimumSize	获取或设置大小，该大小是 GetPreferredSize 可以指定的下限。
MouseButtons	获取一个值，该值指示哪一个鼠标按钮处于按下的状态。
MousePosition	获取鼠标光标的位置（以屏幕坐标表示）。

1.Control 类的属性

Name	获取或设置控件的名称。
Padding	获取或设置控件内的边距。
Parent	获取或设置控件的父容器。
Right	获取控件右边缘与其容器的工作区左边缘之间的距离（以像素为单位）。
Size	获取或设置控件的高度和宽度。
TabIndex	获取或设置在控件的容器的控件的 Tab 键顺序。
Text	获取或设置与此控件关联的文本。
Top	获取或设置控件上边缘与其容器的工作区上边缘之间的距离(以像素为单位）。
Visible	获取或设置一个值，该值指示是否显示该控件。
Width	获取或设置控件的宽度。

控件常见的属性及其用法

- （1） Text 属性
  - 每一个控件都有 Text 属性，是给用户查看或者输入的。 Name 属性虽然也是每个控件对象都有的，不过它却是给程序员看的，常在编程中使用，作为每个控件的名字用于被程序员控制与操作。
  - Text 属性在很多控件中都是经常使用的。例如：在标签框中显示的文字、在编辑框中用户输入的文字。
  - 同样，在程序中也可以直接访问 Text 属性，用来获取和设置 Text 的值，这样就可以实现在程序运行过程中修改标题的名称，获取用户输入的数据等功能。
- （2） Capture 属性
  - Capture 属性如何设为真，则不管鼠标是否在此控件的范围内，鼠标都被限定为只由此控件响应。
- （3） Anchor 和 Dock 属性
  - 在设计窗体时，这两个属性非常实用， .NET 中通过对这两个属性的设置，实现了用户改变窗口大小，却同时确保窗口看起来不显得很乱。
  - Anchor 属性用于指定在用户重新设置窗口的大小时控件该如何响应。可以指定如果控件重新设置了自己的大小，就根据控件自己的边界锁定它，或者其大小不变，但应根据窗口的边界来锚定它的位置。
  - Dock 属性与 Anchor 属性是相关的。可以使用该属性指定控件应停放在容器的边框上。如果用户重新设置了窗口的大小，该控件将继续停放在窗口的边框上。

2.Control 类的方法

名称	说明
Contains	检索一个值，该值指示指定控件是否为一个控件的子控件。
CreateControl	强制创建控件，包括创建句柄和任何子控件。
CreateGraphics	为控件创建 Graphics。
Dispose	已重载。 释放由 Control 使用的所有资源。
DoDragDrop	开始拖放操作。
DrawToBitmap	支持呈现到指定的位图。
Equals	已重载。 确定两个 Object 实例是否相等。 （从 Object 继承。）
FindForm	检索控件所在的窗体。
Focus	为控件设置输入焦点。
FromChildHandle	检索包含指定句柄的控件。
FromHandle	返回当前与指定句柄关联的控件。
GetChildAtPoint	已重载。 检索指定位置的子控件。
GetNextControl	按照子控件的 Tab 键顺序向前或向后检索下一个控件。
GetPreferredSize	检索可以容纳控件的矩形区域的大小。
GetType	获取当前实例的 Type。 （从 Object 继承。）
Hide	对用户隐藏控件。

2.Control 类的方法

Invalidate	已重载。 使控件的特定区域无效并向控件发送绘制消息。
Invoke	已重载。 在拥有此控件的基础窗口句柄的线程上执行委托。
IsKeyLocked	确定 Caps Lock、Num Lock 或 Scroll Lock 键是否有效。
PointToClient	将指定屏幕点的位置计算成工作区坐标。
PointToScreen	将指定工作区点的位置计算成屏幕坐标。
RectangleToClient	计算指定屏幕矩形的大小和位置（以工作区坐标表示）。
RectangleToScreen	计算指定工作区矩形的大小和位置（以屏幕坐标表示）。
Refresh	强制控件使其工作区无效并立即重绘自己和任何子控件。
ResetText	将 Text 属性重置为其默认值。
ResumeLayout	已重载。 恢复正常的布局逻辑。
Scale	已重载。 缩放控件和任何子控件。
Select	已重载。 激活控件。
SelectNextControl	激活下一个控件。
SendToBack	将控件发送到 Z 顺序的后面。
SetBounds	已重载。 设置控件的边界。
Show	向用户显示控件。
SuspendLayout	临时挂起控件的布局逻辑。
ToString	返回包含 Component 的名称的 String（如果有）。不应重写此方法。
Update	使控件重绘其工作区内的无效区域。

3.Control 类的事件

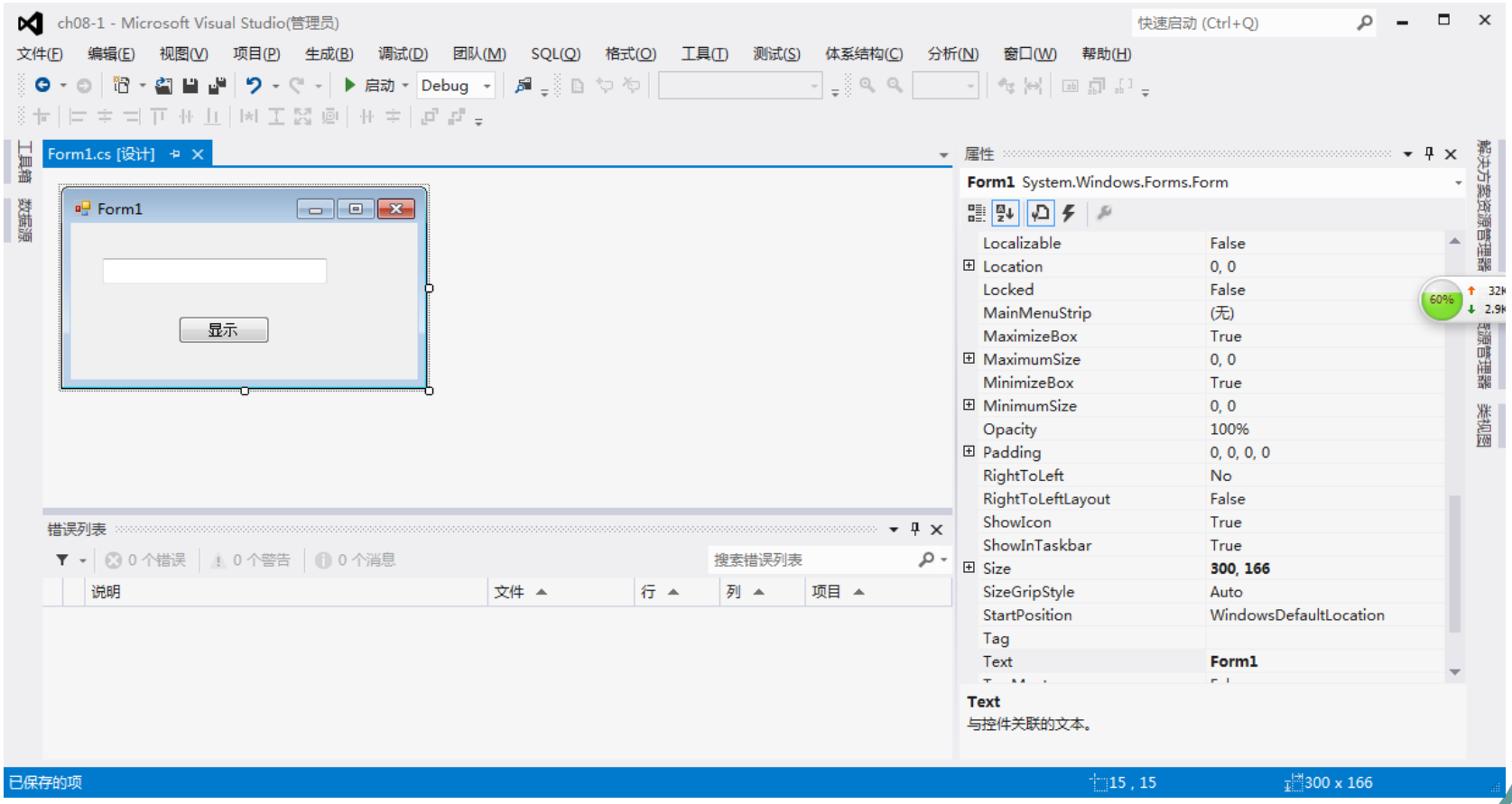
名称	说明
Click	在单击控件时发生。
ClientSizeChanged	当 ClientSize 属性的值更改时发生。
ContextMenuChanged	当 ContextMenu 属性的值更改时发生。
ControlAdded	在将新控件添加到 Control.ControlCollection 时发生。
DoubleClick	在双击控件时发生。
DragOver	在将对象拖到控件的边界上发生。
EnabledChanged	在 Enabled 属性值更改后发生。
Enter	进入控件时发生。
GotFocus	在控件接收焦点时发生。
LostFocus	当控件失去焦点时发生。
MouseCaptureChanged	当控件失去鼠标捕获时发生。
MouseClick	在鼠标单击该控件时发生。
MouseDoubleClick	当用鼠标双击控件时发生。
MouseDown	当鼠标指针位于控件上并按下鼠标键时发生。
MouseEnter	在鼠标指针进入控件时发生。
MouseHover	在鼠标指针停放在控件上时发生。
MouseLeave	在鼠标指针离开控件时发生。
MouseMove	在鼠标指针移到控件上时发生。
MouseUp	在鼠标指针在控件上并释放鼠标键时发生。
MouseWheel	在移动鼠标轮并且控件有焦点时发生。
Move	在移动控件时发生。
SizeChanged	在 Size 属性值更改时发生。
TextChanged	在 Text 属性值更改时发生。
Validated	在控件完成验证时发生。

Button 控件

- 几乎所有的 Windows 对话框中都存在按钮控件，对于按钮的处理比较简单，通常是在窗体上添加控件，再双击它，给 Click 事件添加代码。

使用 Button 控件的一个例子

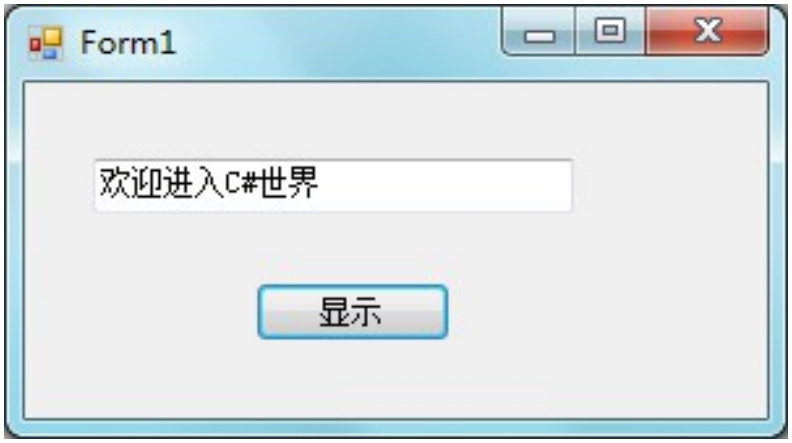
- **例 8.1** Button 控件的 Click 事件演示。
- 程序操作步骤：
  - （1）在 Visual Studio 2012 开发环境中，选择“文件”→“新建”→“项目”命令，弹出“新建项目”对话框。
  - （2）在左边选中“ Visual C#”，右边选中“ Windows 窗体应用程序”选项，然后在该对话框下方的“名称”文本框中，输入该项目的名称，如“ ch08-1”，在“位置”文本框中，输入保存该项目的位置，也可单击“浏览”按钮来选定保存位置。
  - （3）单击“确定”按钮，在 Visual Studio 2012 的编辑窗口中将显示一个空白窗体。
  - （4）打开工具箱，双击 Button 控件一次，在属性面板上选择 Name 属性，改为“ btnShow”，将 Text 属性改为“显示”，再双击 TextBox 控件一次，在属性面板上选择 Name 属性，改为“ txtOuput”。效果图如下页图所示。


ch08-1 - Microsoft Visual Studio(管理员)
快速启动 (Ctrl+Q)
文件(F) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 团队(M) SQL(Q) 格式(O) 工具(T) 测试(S) 体系结构(C) 分析(N) 窗口(W) 帮助(H)
启动 Debug
Form1.cs [设计]
Form1
显示
属性
Form1 System.Windows.Forms.Form
Localizable False
Location 0, 0
Locked False
MainMenuStrip (无)
MaximizeBox True
MaximumSize 0, 0
MinimizeBox True
MinimumSize 0, 0
Opacity 100%
Padding 0, 0, 0, 0
RightToLeft No
RightToLeftLayout False
ShowIcon True
ShowInTaskbar True
Size 300, 166
SizeGripStyle Auto
StartPosition WindowsDefaultLocation
Tag
Text Form1
Text
与控件关联的文本。
错误列表
0 个错误 0 个警告 0 个消息
搜索错误列表
说明 文件 行 列 项目
已保存的项
15 , 15
300 x 166

程序代码

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace ch08_1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void btnShow_Click(object sender, EventArgs e)
        {
            txtOuput.Text = " 欢迎进入 C# 世界 ";
        }
    }
}
```

运行结果



TextBox 控件

- 文本框（ TextBox ）经常用于获取用户输入或显示文本，通常用于可编辑文本，也可以设定其成为只读控件。文本框能够显示多行数据，并添加基本的格式设置。
- Text 属性是文本框最重要的属性，要显示的文本就包含在 Text 属性中。 Text 属性可以在设计窗口时使用属性窗口设置，也可以在运行时用代码设置或者通过用户输入设置，同样也可以在运行时通过读取 Text 属性来获得文本框的当前内容。

TextBox 控件的常用属性

名称	说明
AutoSize	获取或设置一个值，该值指示当更改分配给控件的字体时，是否自动调整控件的高度。此属性与此类无关。
BackColor	获取或设置控件的背景色。
CausesValidation	获取或设置一个值，该值指示控件是否会引起在任何需要在接收焦点时执行验证的控件上执行验证。
CharacterCasing	获取或设置 TextBox 控件是否在字符键入时修改其大小写格式。
DataBindings	为该控件获取数据绑定。
Enabled	获取或设置一个值，该值指示控件是否可以对用户交互作出响应。
MaximumSize	获取或设置大小，该大小是 GetPreferredSize 可以指定的上限。
MaxLength	获取或设置用户可在文本框控件中键入或粘贴的最大字符数。
MinimumSize	获取或设置大小，该大小是 GetPreferredSize 可以指定的下限。
Multiline	已重写。获取或设置一个值，该值指示此控件是否为多行 TextBox 控件。
Name	获取或设置控件的名称。
PasswordChar	获取或设置字符，该字符用于屏蔽单行 TextBox 控件中的密码字符。
ReadOnly	获取或设置一个值，该值指示文本框中的文本是否为只读。
ScrollBars	获取或设置哪些滚动条应出现在多行 TextBox 控件中。
SelectedText	获取或设置一个值，该值指示控件中当前选定的文本。
SelectionLength	获取或设置文本框中选定的字符数。
SelectionStart	获取或设置文本框中选定的文本起始点。
Text	已重写。获取或设置 TextBox 中的当前文本。
TextAlign	获取或设置 TextBox 控件中文本的对齐方式。
TextLength	获取控件中文本的长度。
Visible	获取或设置一个值，该值指示是否显示该控件。
WordWrap	指示多行文本框控件在必要时是否自动换行到下一行的开始。

TextBox 控件常用的事件

名称	说明
Enter	进入控件时发生。
GotFocus	在控件接收焦点时发生。
Leave	在输入焦点离开控件时发生。
Validating	在控件正在验证时发生。
Validated	在控件完成验证时发生。
LostFocus	当控件失去焦点时发生。
KeyDown	在控件有焦点的情况下按下键时发生。
KeyPress	在控件有焦点的情况下按下键时发生。
KeyUp	在控件有焦点的情况下释放键时发生。
TextChanged	在 Text 属性值更改时发生。

RadioButton 与 CheckBox 控件

- 单选按钮（ RadioButton ）通常成组出现，用于为用户提供两个或多个互相排斥的选项，如下图所示。单选按钮和后来将要介绍的复选框（ CheckBox ）控件类似，但也存在重要的区别，即从一组单选按钮中必须且只能选择一个，而在一组复选框中这可以同时选择多个选项。
- 把单选按钮组合在一起，使它们创建一个逻辑单元，此时必须使用 GroupBox 控件。首先在窗体上拖放一个 GroupBox 控件（组框），再把需要的 RadioButton 按钮放在组框的边界内，RadioButton 按钮知道如何改变自己的状态，以反应组框中唯一被选中的选项。

RadioButton 与 CheckBox 控件

- 复选框（ CheckBox ）指示某特定条件是打开的还是关闭的。当用户希望选择一个或多个选项时，就需要使用复选框。一个复选框如下图所示。

RadioButton 控件属性

属性名称	说明
Appearance	获取或设置一个值，该值用于确定 RadioButton 的外观。
AutoCheck	获取或设置一个值，它指示：在单击控件时，Checked 值和控件的外观是否自动更改。
CheckAlign	获取或设置 RadioButton 的复选框部分的位置。
Checked	获取或设置一个值，该值指示是否已选中控件。
Enabled	获取或设置一个值，该值指示控件是否可以对用户交互作出响应。
FlatStyle	获取或设置按钮控件的平面样式外观。
Name	获取或设置控件的名称。
Text	与控件关联的文本
TextAlign	获取或设置 RadioButton 控件上的文本对齐方式。

RadioButton 控件事件

事件名称	说明
Click	在单击控件时发生。
CheckedChanged	当 Checked 属性的值更改时发生。

CheckBox 控件属性

属性名称	说明
CheckState	获取或设置 CheckBox 的状态。
ThreeState	获取或设置一个值，该值指示此 CheckBox 是否允许三种复选状态而不是两种。

CheckBox 控件事件

事件名称	说明
CheckedChanged	当 Checked 属性的值更改时发生。
CheckStateChanged	当 CheckState 属性的值更改时发生。
Click	在单击控件时发生。

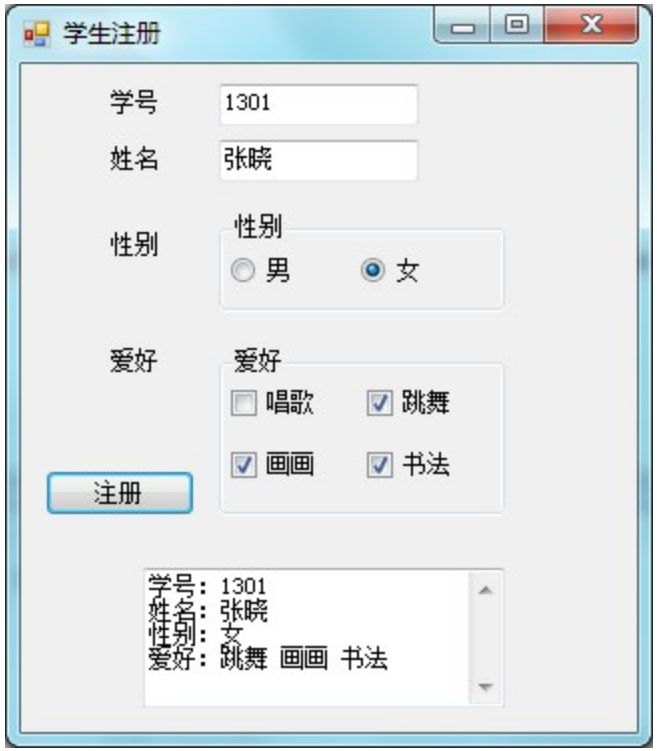
一个关于 RadioButton 控件以及 CheckBox 控件的使用的简单例子

- 程序界面

程序代码

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace ch08_3
{
    public partial class Form1 : Form
    {
        public Form1()
        {    InitializeComponent();     }
        private void btnRegister_Click(object sender, EventArgs e)
        {   string sex="";
            string hobby="";
            if (rdoMale.Checked)
                sex = " 男 ";
            else
                sex = " 女 ";
            if (ckSing.Checked)
                hobby += " 唱歌   ";
            if (ckDance.Checked)
                hobby += " 跳舞   ";
            if (ckDraw.Checked)
                hobby += " 画画   ";
            if (ckWrite.Checked)
                hobby += " 书法 ";
            txtOutput.Text = " 学号：  " + txtSno.Text + "\r\n 姓名：  " + txtName.Text + "\r\n 性别：  " + sex + "\r\n 爱好：  " + hobby;      }
    }
}
```

程序运行结果


学生注册
学号
1301
姓名
张晓
性别
性别
男
女
爱好
爱好
唱歌
跳舞
画画
书法
注册
学号：1301
姓名：张晓
性别：女
爱好：跳舞 画画 书法

ListBox 控件

- 列表框用于显示一组字符串，可以一次从中选择一个或多个选项。例如，在设计期间，如果不知道用户要选择的数值个数，或者列表中的值非常多的时候，就应考虑使用列表框。

ListBox 控件属性

名称	说明
DataBindings	为该控件获取数据绑定。
DataSource	获取或设置此 ListControl 的数据源。
Items	获取 ListBox 的项。
Name	获取或设置控件的名称。
SelectedIndex	获取或设置 ListBox 中当前选定项的从零开始的索引。
SelectedItem	获取或设置 ListBox 中的当前选定项。
SelectedItems	获取包含 ListBox 中当前选定项的集合。
SelectedValue	获取或设置由 ValueMember 属性指定的成员属性的值。
SelectionMode	获取或设置在 ListBox 中选择项所用的方法。
Sorted	获取或设置一个值，该值指示 ListBox 中的项是否按字母顺序排序。
Text	获取或搜索 ListBox 中当前选定项的文本。

ListBox 控件常见方法

方法名称	说明
ClearSelected	取消选择 ListBox 中的所有项。
FindString	查找 ListBox 中以指定字符串开始的第一个项。
FindStringExact	查找 ListBox 中第一个精确匹配指定字符串的项。
GetItemText	返回指定项的文本表示形式。
GetSelected	返回一个值，该值指示是否选定了指定的项。
SetSelected	选择或清除对 ListBox 中指定项的选定。
ToString	返回 ListBox 的字符串表示形式。

ListBox 控件常见事件

名称	说明
TextChanged	当 Text 属性更改时发生。
SelectedIndexChanged	在 SelectedIndex 属性更改后发生。

ComboBox 控件

- 与 ListBox 不同，组合框（ ComboBox ）从来都不能在列表中选择多个选项，但可以在 ComboBox 的 TextBox 部分输入新选项。
- 通常，组合框比 ListBox 节省空间，因为组合框中可见的部分只有文本框和按钮部分。

ComboBox 控件属性

名称	说明
DropDownStyle	获取或设置指定组合框样式的值。
DroppedDown	获取或设置一个值，该值指示组合框是否正在显示其下拉部分。
Items	获取一个对象，该对象表示该 ComboBox 中所包含项的集合。
MaxDropDownItems	获取或设置要在 ComboBox 的下拉部分中显示的最大项数。
MaxLength	获取或设置组合框可编辑部分中最多允许的字符数。
Name	获取或设置控件的名称。
SelectedIndex	获取或设置指定当前选定项的索引。
SelectedItem	获取或设置 ComboBox 中当前选定的项。
SelectedText	获取或设置 ComboBox 的可编辑部分中选定的文本。
SelectedValue	获取或设置由 ValueMember 属性指定的成员属性的值。
SelectionLength	获取或设置组合框可编辑部分中选定的字符数。
SelectionStart	获取或设置组合框中选定文本的起始索引。
Sorted	获取或设置指示是否对组合框中的项进行了排序的值。
Text	获取或设置与此控件关联的文本。

ComboBox 控件事件

名称	说明
DropDown	当显示 ComboBox 的下拉部分时发生。
SelectedIndexChanged	在 SelectedIndex 属性更改后发生。
SelectedValueChanged	当 SelectedValue 属性更改时发生。
KeyDown	在控件有焦点的情况下按下键时发生。
KeyPress	在控件有焦点的情况下按下键时发生。
KeyUp	在控件有焦点的情况下释放键时发生。
TextChanged	在 Text 属性值更改时发生。

ListView 控件

- ListView 是 Windows 列表视图控件，用于显示来自应用程序、数据库或文本文件的信息或者获取来自用户的信息。在标准列表视图对话框中可以进行各种查看操作，如：图标、详细视图等。
- 列表视图通常用于显示数据，用户可以对这些数据和显示方式进行某些控件，可以把包含在控件中的数据显示为列和行，或者显示为一列，或者先是为图标形式。
- ListView 控件的主要属性就是 Items，该属性是一个包含控件所显示的项的集合，可用于在列表视图中的添加和移除项。 SelectedItems 属性则包含控件中当前选定项的集合。如何将 MultiSelect 属性设置为 True，用户就可以同时选择多项。 ListViewItem 类用于表示列表视图中的项，这些项可以包含子项，子项包含与父项相关的信息。
- 在应用程序中，我们经常使用方法和事件为列表视图提供附加功能。BeginUpdate 和 EndUpdate 方法用于为列表视图添加许多项，而且在每次添加项时并不显示控件的重新绘制，这样就提高了性能。

ListView 控件属性

名称	说明
Activation	获取或设置用户激活某个项必须要执行的操作的类型。
Alignment	获取或设置控件中项的对齐方式。
AllowColumnReorder	获取或设置一个值，该值指示用户是否可拖动列标头来对控件中的列重新排序。
AutoArrange	获取或设置图标是否自动进行排列。
CheckBoxes	获取或设置一个值，该值指示控件中各项的旁边是否显示复选框。
CheckedIndices	获取控件中当前选中项的索引。
CheckedItems	获取控件中当前选中的项。
Columns	获取控件中显示的所有列标头的集合。
FocusedItem	获取当前具有焦点的控件中的项。
FullRowSelect	获取或设置一个值，该值指示单击某项是否选择其所有子项。
GridLines	获取或设置一个值，该值指示：在包含控件中项及其子项的行和列之间是否显示网格线。
HeaderStyle	获取或设置列标头样式。
HoverSelection	获取或设置一个值，该值指示当鼠标指针在项上停留几秒钟时是否自动选定该项。
Items	获取包含控件中所有项的集合。
LabelEdit	获取或设置一个值，该值指示用户是否可以编辑控件中项的标签。
LabelWrap	获取或设置一个值，该值指示当项作为图标在控件中显示时，项标签是否换行。
LargeImageList	获取或设置当项以大图标在控件中显示时使用的 ImageList。
MultiSelect	获取或设置一个值，该值指示是否可以选择多个项。
Scrollable	获取或设置一个值，该值指示在没有足够空间来显示所有项时，是否给滚动条添加控件。
SelectedIndices	获取控件中选定项的索引。
SelectedItems	获取在控件中选定的项。
SmallImageList	获取或设置 ImageList，当项在控件中显示为小图标时使用。
Sorting	获取或设置控件中项的排序顺序。
StateImageList	获取或设置与控件中应用程序定义的状态相关的 ImageList。
TopItem	获取或设置控件中的第一个可见项。

ListView 控件方法

名称	说明
BeginUpdate	避免在调用 EndUpdate 方法之前描述控件。
Clear	从控件中移除所有项和列。
EndUpdate	在 BeginUpdate 方法挂起描述后，继续描述列表视图控件。
EnsureVisible	确保指定项在控件中是可见的，必要时滚动控件的内容。
GetItemAt	检索位于指定位置的项。

ListView 控件事件

名称	说明
AfterLabelEdit	当用户编辑项的标签时发生。
BeforeLabelEdit	当用户开始编辑项的标签时发生。
ColumnClick	当用户在列表视图控件中单击列标头时发生。
ItemActivate	当激活项时发生。

8.2 用户自定义控件

- 虽然 Visual Studio.NET 附带了大量的控件，但仍不能满足各个应用程序的特殊需要。比如说， Visual Studio .NET 自带的控件不能以我们希望的方式绘制自己，或者控件只能以某种方式使用，而我们却希望把控件的功能和界面一起封装，或者需要的控件不存在。此时，我们就需要自己开发一个新的控件。自定义控件基本思想是允许开发人员生成新的功能，把现有的控件聚集到一个公共控件上，使之可以在应用程序种重复使用，或通过组织在多个应用程序种重复使用。为此， Microsoft 提供了创建满足需要的控件方式。 Visual Studio.NET 提供了一个工程类型 Windows Control Library ，使用它可以创建自己的控件。

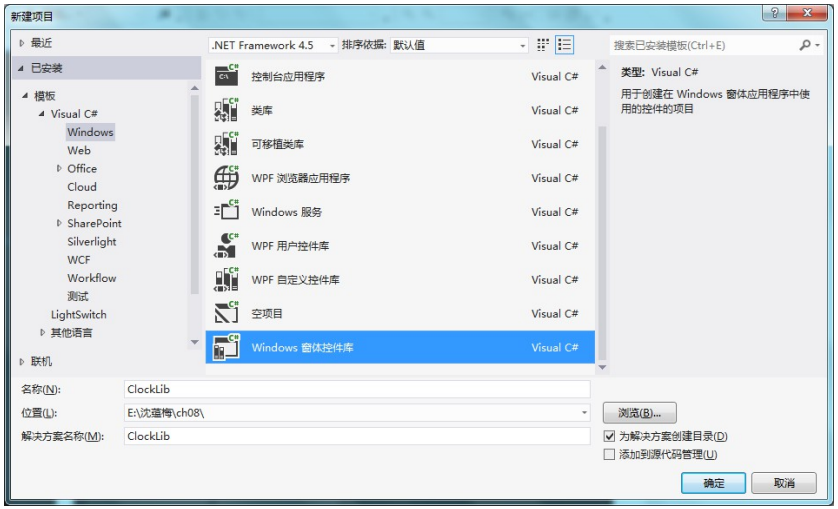
开发定制控件的三种方法

- 1. 从 Windows 窗体控件继承
- 2. 从 UserControl 类继承
- 3. 从 Control 类继承

定制控件示例 ----- 从 UserControl 类继承

- 下面的示例说明了如何通过组合控件的方式来自定义控件。本示例将 Label 和 Timer 两个控件绑定到一起，实现通过标签显示系统当前时间，每秒刷新一次。

- （1）在 Visual Studio 2012 中创建一个新的 C# 工程，选择【 Windows 控件库】，把新工程命名为“ ClockLib”，如下图所示。

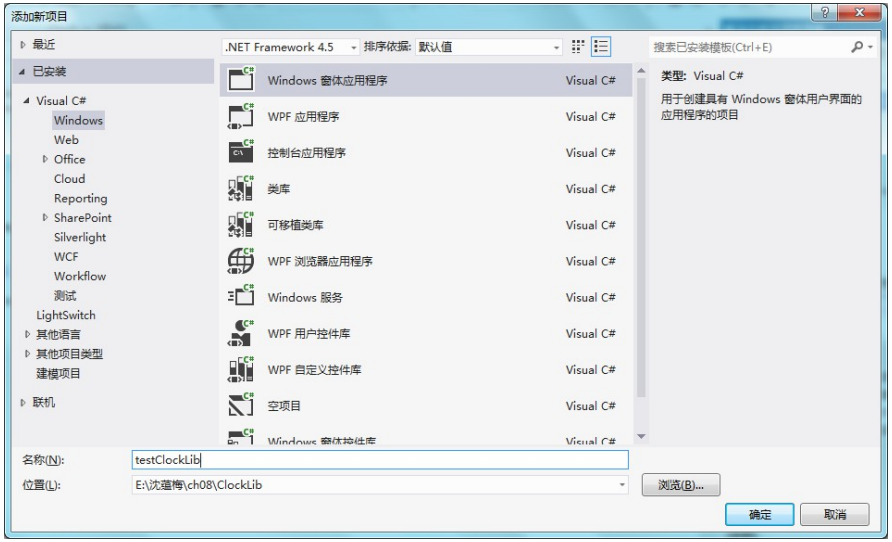


- （2）在“解决方案资源管理器”中右击“UserControl1.cs”，再从快捷菜单中选择“重命名”。将文件名更改为 Clock.cs 。系统询问是否重命名对代码元素“UserControl1”的所有引用时，单击“是”按钮 。
- （3）打开控件设计器的代码，找到 public partial class StringOfButton : UserControl 。默认情况下，用户控件从系统提供的 UserControl 类继承。 UserControl 类提供所有用户控件所要求的功能，并实现标准方法和属性。
- （4）在用户控件中加入标签和计时器两个控件。在“解决方案资源管理器”中，切换到 Clock 控件设计器，在“工具箱”中单击“所有 Windows 窗体”选项卡，然后为 Clock 控件设计器添加一个 Label ，名为 Label1 的标签控件被添加到用户控件设计器上的控件中。

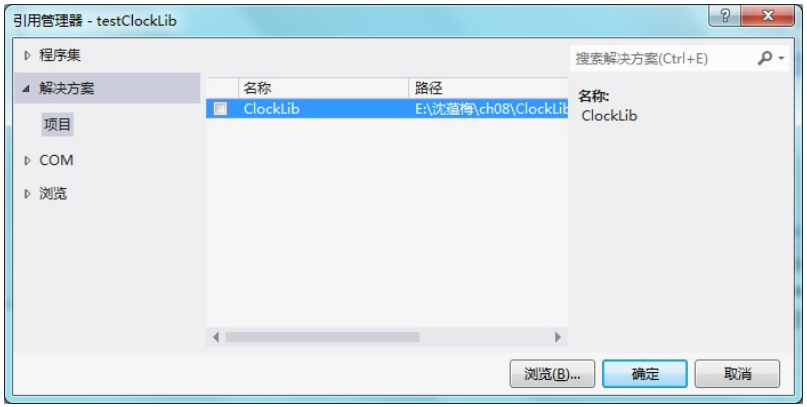
- 选中 timer1 控件，切换到“事件窗口”，双击“ Tick”，为 time1 控件添加一个 timer1_Tick 事件。切换到代码编辑器，找到 timer1_Tick 事件的代码，将代码修改如下：
  - protected virtual void timer1_Tick(object sender, EventArgs e)
  - {
  - // 在标签中显示当前的时间
  - lblDisplay.Text = System.DateTime.Now.ToLongTimeString();
  - }
- 修饰符从 Private 更改为 Protected，用 Virtual 关键字修改该方法使其可被重写。

- （5）从“文件”菜单中，选择“全部保存”命令来保存项目。
- （6）生成控件。在“生成”菜单中单击“生成 ClockLib”命令，输出窗体提示生成是否成功。
- （7）创建测试项目。由于定制的控件不是独立的项目，它们必须寄宿在容器中。因此，必须提供一个运行该控件的测试项目，来进行测试控件。

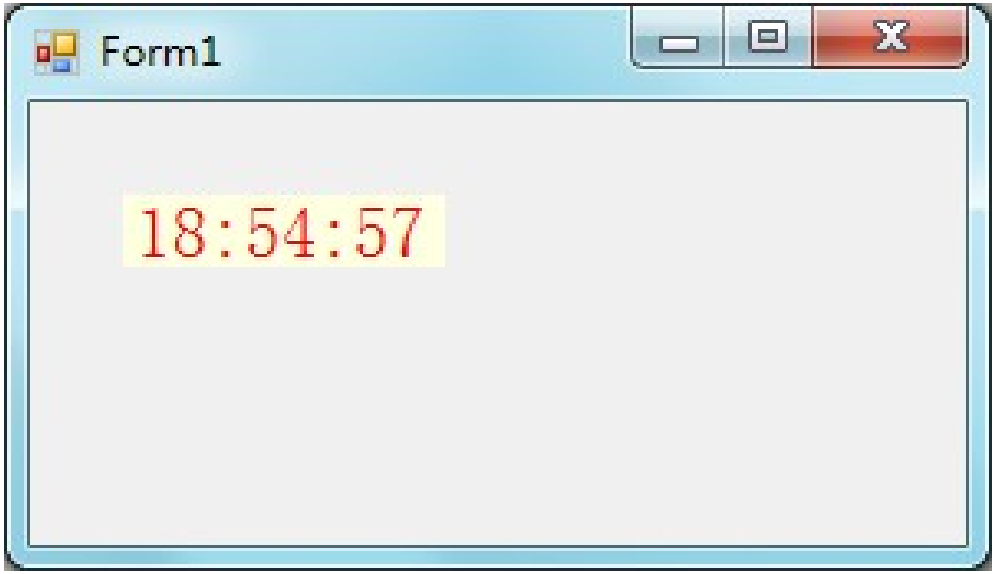
- 在“文件”菜单上，指向“添加”，然后单击“新建项目”打开“添加新项目”对话框。在“ Visual C#”节点下选择“ Windows”节点，再单击“ Windows 窗体应用程序”。在“名称”框中键入 testClockLib, 如下图所示。



- （8）添加引用后，需要将新控件添加到工具箱。
- 在解决方案资源管理器中，右键单击 testClockLib 项目的“添加引用”节点命令以打开【添加引用】对话框。如图 8.34 所示。选中“ ClockLib”，单击【确定】按钮。在“解决方案资源管理器”中，右击“ testClockLib”并选择“生成”。



- （9）将 Clock 控件添加到 testClockLib 的窗体设计器上，并调整到适当的大小。窗体中显示一个名为“ clock1”的定制控件。
- （10）在解决方案资源管理器中，右键单击 testClockLib 然后从快捷菜单中选择“设为启动项目”。
- （11）按 F5 键运行该项目，出现 Form1。效果图如下页图所示：


Form1
18:54:57

小结

- 本章首先介绍了 Windows 应用程序常用的控件，及其相关的属性、方法和事件，开发人员可以使用这些控件编写复杂的应用程序，使用这些控件进行开发，可以减少开发者很多的重复性工作。
- .NET 允许开发者自己根据实际需求创建出新的控件，并提供了三种常用的定制控件开发方式，我们根据需要，来选择一种合适自己的定制方式。

谢谢