

## 项目 3 设计制作网络调查系统

本项目制作一个简单的网络调查系统。本网络调查系统的设计思路为：首先，用户设置调查项并生成调查问卷，对网上的浏览者进行调查。网上浏览者对调查问卷进行相应的选择和回答后，提交一个调查结果。调查结果在服务器端汇总，得出调查结论。本项目将为用户提供一个迅速、便捷地得到用户反馈的方式，及时地了解用户对某些问题的真实态度，为后面的决策提供参考和依据。

在本项目的设计中，我们将使用到 ASP.NET 的一些常用控件，比如单选控件、复选控件和选择控件组；另外，还会用到 ASP.NET 的常见内置对象，如 Response、Request、Server、Application、Session 等。同时还将接触到其他一些控件。

### 本项目学习目标

#### 1. 知识目标

- (1) 了解 ASP.NET 基本控件的特点。
- (2) 了解 ASP.NET 常见内置对象的特点。
- (3) 了解 ASP.NET 单选及复选控件的特点。
- (4) 了解 ASP.NET 下拉列表控件的特点。

#### 2. 能力目标

- (1) 掌握 ASP.NET 常见控件的使用方法。
- (2) 掌握 ASP.NET 内置对象的使用方法。
- (3) 掌握 Calendar 控件的使用方法。
- (4) 掌握 C# 基本语句的编写和调试程序的方法。

#### 3. 综合素质目标

- (1) 引导学生了解编程规范的重要性。
- (2) 培养学生的职业素质和道德规范。
- (3) 培养学生养成反复调试、精益求精的态度。

## 任务 3.1 使用常见控件

### 3.1.1 使用 RadioButton 和 RadioButtonList 控件

#### 1. 要求与目的

要求：制作如图 3-1 所示的学生注册界面，要求用户填写姓名，选择自己的性别，并选

择年级,年级包括“2007级”“2008级”和“2009级”。填写完成,单击“提交”按钮,显示出用户所填内容。



使用 RadioButton 和  
RadioButtonList 控件

图 3-1 学生注册界面

目的:

- (1) 掌握单选按钮控件 RadioButton 常用的属性及编程方法。
- (2) 掌握单选列表控件 RadioButtonList 常用的属性及编程方法。

## 2. 设计步骤

第一步: 界面设计。

打开 Visual Studio 2012,新建一个 ASP.NET 空网站,语言选择 C#,然后添加一个 ASP.NET 页面。首先做一个布局表格,在表格中输入如下文本“用户注册”“用户名:”“性别:”“年级:”。在对应的位置添加如下控件:1 个 TextBox 文本框控件,用于存放用户名;2 个 RadioButton 按钮控件,用于让用户选择性别;1 个 RadioButtonList 控件,用于让用户选择年级,如图 3-2 所示。



图 3-2 设计界面

布局完成后,修改控件的属性,首先分别右击 2 个 RadioButton 控件,修改其 Text 属性为“男”和“女”。

再单击 RadioButtonList 控件的智能标签,选择“编辑项”选项,出现如图 3-3 所示的界面。

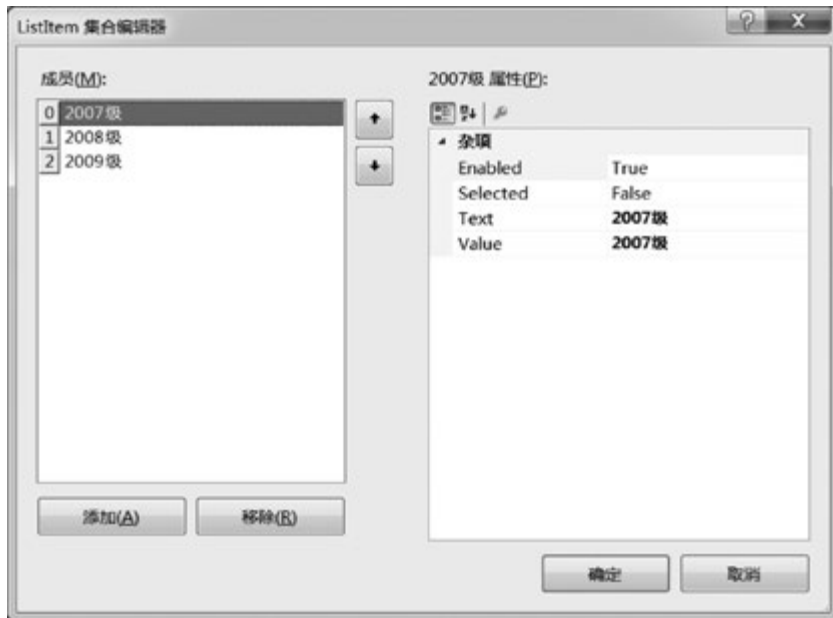


图 3-3 编辑单选项

在“ListItem 集合编辑器”对话框中添加 3 项,分别是“2007 级”“2008 级”“2009 级”,添加完成后单击“确定”按钮。

最后右击 Button 按钮,修改其 Text 属性为“注册”。

界面设计好之后,主要控件的 HTML 源文件如代码 3-1 所示。

#### 代码 3-1 主要控件的 HTML 源文件

```
<table class="style1">
  <tr>
    <td colspan="2" style="text-align: center;">
      用户注册</td>
  </tr>
  <tr>
    <td class="style2">
      用户名: </td>
    <td>
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    </td>
  </tr>
  <tr>
    <td class="style2">
      性别: </td>
    <td>
      <asp:RadioButton ID="RadioButton1" runat="server" Text="男" />
      <asp:RadioButton ID="RadioButton2" runat="server" Text="女" />
    </td>
  </tr>
</table>
```

```

</tr>
<tr>
  <td class="style2">
    年级: </td>
  <td>
    <asp:RadioButtonList ID="RadioButtonList1" runat="server">
      <asp:ListItem>2007 级</asp:ListItem>
      <asp:ListItem>2008 级</asp:ListItem>
      <asp:ListItem>2009 级</asp:ListItem>
    </asp:RadioButtonList>
  </td>
</tr>
<tr>
  <td colspan="2" style="text-align: center">
    <asp:Button ID="Button1" runat="server" Height="25px" Text="注册"
      Width="87px" />
  </td>
</tr>
</table>

```

第二步：编写事件代码。

双击 Button 控件,进入该按钮的单击事件,编写的代码如代码 3-2 所示。

### 代码 3-2 Button 控件的单击事件

```

protected void Button1_Click(object sender, EventArgs e)
{
    string a1=TextBox1.Text;
    string a2="";
    if (RadioButton1.Checked)
    {
        a2="男";
    }
    else if (RadioButton2.Checked)
    {
        a2="女";
    }
    string a3=RadioButtonList1.SelectedItem.ToString();
    Response.Write("用户名是"+a1+"<br>");
    Response.Write("性别是"+a2+"<br>");
    Response.Write("年级是"+a3+"<br>");
}

```

第三步：编译、运行并测试程序。

代码编写好之后,按 F5 键编译、运行该程序,并输入数据测试。

## 3. 相关知识点

(1) RadioButton 控件常用的属性说明如下。

① AutoPostBack 属性：用于设置当单击 RadioButton 控件时是否自动回送信息到服务器, True 表示回送, False(默认)表示不回送。

② Checked 属性：用于获取或设置单选按钮的选中状态。True 表示选中, False(默认) 表示不选中。

③ Text 属性：用于获取或设置单选按钮的文本。

④ TextAlign 属性：用于指定单选按钮显示文本的显示位置。Right(默认) 表示文本显示在单选按钮的右边；Left 表示文本显示在单选按钮的左边。

(2) RadioButton 控件常用的事件说明如下。

CheckedChange 事件：当向服务器回送信息时, 如果 RadioButton 控件属性值发生了变化, 将引发该事件。

(3) RadioButtonList 控件常用的属性说明如下。

① ID：为控件提供唯一标识, 方便在服务器端代码中引用和操作该控件, 确保不同控件在页面内不会混淆。

② runat：值设为“server”, 使控件成为服务器端控件, 这样服务器能够对其进行处理, 如接收用户操作并进行相应响应。

③ DataSource：指定控件的数据来源, 它可以是实现 IEnumerable 接口的对象, 像 DataTable 等, 通过它能为控件动态填充数据。

④ DataTextField：在使用 DataSource 绑定时, 指定数据源中用于显示在单选按钮上的字段名称。

⑤ DataValueField：绑定数据时, 指定数据源中用于存储每个单选按钮对应值的字段名称, 方便后续获取选中值。

⑥ RepeatDirection：设置单选按钮的排列方向, 有“Horizontal”(水平排列) 和“Vertical”(垂直排列) 两种可选, 以满足不同布局需求。

⑦ RepeatColumns：结合 RepeatDirection 使用, 当水平排列时指定每行按钮数量, 垂直排列时指定每列按钮数量。

⑧ CssClass：应用 CSS 类到控件上, 可通过 CSS 样式自定义控件的外观, 如字体、颜色、边框等。

⑨ SelectedIndex：获取或设置当前选中单选按钮的索引, 索引从 0 开始, 若未选中任何项则为 -1。

⑩ SelectedValue：获取或设置当前选中单选按钮的值, 未选中时为 null, 便于获取用户选择信息。

(4) RadioButtonList 控件常用的事件说明如下。

SelectIndexChange 事件：当用户选择了列表中的任意选项时, 都将触发该事件。

### 3.1.2 使用 CheckBox 和 CheckBoxList 控件

#### 1. 要求与目的

要求：在 3.1.1 小节案例的基础上增加用户注册的功能, 添加用户提交的信息。增添一个“个人爱好”选择项, 以及“本学期选修课程”选择项, 效果如图 3-4 所示。



使用 CheckBox 和 CheckBoxList 控件

目的:

- (1) 了解 CheckBox 控件的常用属性及编程方法。
- (2) 了解 CheckBoxList 控件的常用属性及编程方法。

## 2. 操作步骤

第一步: 界面设计。

在 3.1.1 小节案例的基础上添加“个人爱好”和“本学期选修课程”选项区。其中“个人爱好”选项区使用 4 个 CheckBox(复选按钮)控件来实现,本学期所选课程使用一个 CheckBoxList(复选按钮列表)控件来实现。效果如图 3-5 所示。



图 3-4 运行效果预览

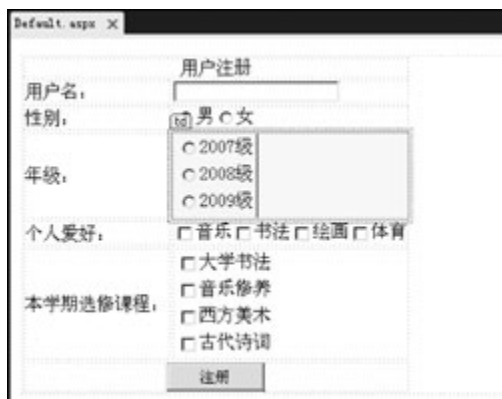


图 3-5 设计界面

在做好布局之后,修改控件的属性。首先分别右击 4 个 CheckBox 控件,修改其 Text 属性分别为“音乐”“书法”“绘画”“体育”。单击 CheckBoxList 控件的智能标签,选择“编辑项”选项,打开如图 3-6 所示的“ListItem 集合编辑器”对话框。

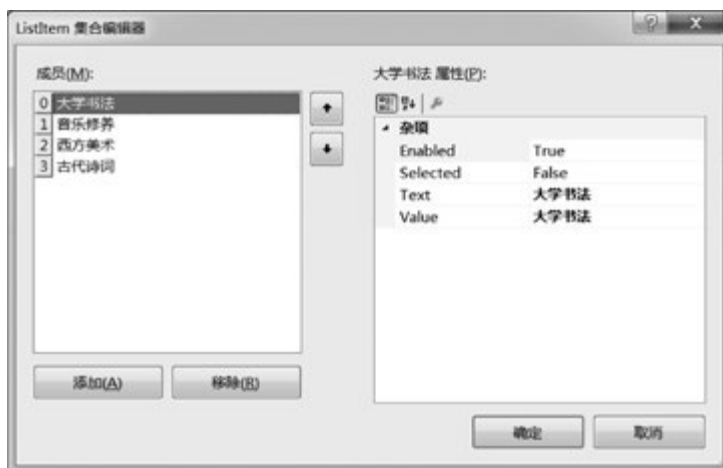


图 3-6 “ListItem 集合编辑器”对话框

在该编辑器对话框中添加四个选项,分别是“大学书法”“音乐修养”“西方美术”“古代诗词”。

界面设计之后,主要控件的 HTML 源文件如代码 3-3 所示。

### 代码 3-3 主要控件的 HTML 源文件

```
<table class="style1">
  <tr>
    <td colspan="2" style="text-align: center">
      用户注册</td>
    </tr>
  <tr>
    <td class="style2">
      用户名: </td>
    <td>
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    </td>
  </tr>
  <tr>
    <td class="style2">
      性别: </td>
    <td>
      <asp:RadioButton ID="RadioButton1" runat="server" Text="男" />
      <asp:RadioButton ID="RadioButton2" runat="server" Text="女" />
    </td>
  </tr>
  <tr>
    <td class="style2">
      年级: </td>
    <td>
      <asp:RadioButtonList ID="RadioButtonList1" runat="server">
        <asp:ListItem>2007 级</asp:ListItem>
        <asp:ListItem>2008 级</asp:ListItem>
        <asp:ListItem>2009 级</asp:ListItem>
      </asp:RadioButtonList>
    </td>
  </tr>
  <tr>
    <td class="style2">
      个人爱好: </td>
    <td>
      <asp:CheckBox ID="CheckBox1" runat="server" Text="音乐" />
      <asp:CheckBox ID="CheckBox2" runat="server" Text="书法" />
      <asp:CheckBox ID="CheckBox3" runat="server" Text="绘画" />
      <asp:CheckBox ID="CheckBox4" runat="server" Text="体育" />
    </td>
  </tr>
  <tr>
    <td class="style2">
      本学期选修课程: </td>
    <td>

```

```

        <asp:CheckBoxList ID="CheckBoxList1" runat="server">
            <asp:ListItem>大学书法</asp:ListItem>
            <asp:ListItem>音乐修养</asp:ListItem>
            <asp:ListItem>西方美术</asp:ListItem>
            <asp:ListItem>古代诗词</asp:ListItem>
        </asp:CheckBoxList>
    </td>
</tr>

<tr>
    <td colspan="2" style="text-align: center">
        <asp:Button ID="Button1" runat="server" Height="25px" Text="注册"
            Width="87px" />
    </td>
</tr>
</table>

```

第二步：编写事件代码。

双击“注册”按钮，进入该按钮的单击事件，编写事件代码如代码 3-4 所示。

#### 代码 3-4 “注册”按钮的单击事件

```

protected void Button1_Click(object sender, EventArgs e)
{
    string a1=TextBox1.Text;
    string a2="";
    if (RadioButton1.Checked)
    {
        a2="男";
    }
    else if (RadioButton2.Checked)
    {
        a2="女";
    }
    string a3=RadioButtonList1.SelectedItem.ToString();
    string a4="";
    if (CheckBox1.Checked)
    {
        a4+=CheckBox1.Text+",";
    }
    if (CheckBox2.Checked)
    {
        a4+=CheckBox2.Text+",";
    }
    if (CheckBox3.Checked)
    {
        a4+=CheckBox3.Text+",";
    }
    if (CheckBox4.Checked)
    {
        a4+=CheckBox4.Text;
    }
}

```

```

string a5="";
Response.Write("用户名是"+a1+"<br>");
Response.Write("性别是"+a2+"<br>");
Response.Write("年级是"+a3+"<br>");
Response.Write("爱好是"+a4+"<br>");
for(int i=0;i<CheckBoxList1.Items.Count;i++)
{
    if(CheckBoxList1.Items[i].Selected==true)
    {
        a5+=CheckBoxList1.Items[i].Value.ToString()+",";
    }
}
Response.Write("选修的课程是"+a5);
}

```

第三步：编译、运行并测试。

代码编写完之后进行保存，编译、运行并测试该程序，得到最终效果。

### 3. 相关知识

CheckBox 和 CheckBoxList 控件的属性和事件的用法请参考 RadioButton 和 RadioButtonList 控件的相关内容。

#### 3.1.3 使用 DropDownList 控件

DropDownList 控件用于在 Web 页中可以创建下拉列表，并允许用户从预定义的下拉列表中选择一项。

(1) DropDownList 控件常用的属性说明如下。

① AutoPostBack 属性：用于设置当改变选项内容时是否自动回送信息到服务器。True 表示回送，False(默认)表示不回送。

② DataSource 属性：用于指定填充列表控件的数据源。

③ DataTextField 属性：用于指定 DataSource 中的一个字段，该字段的值对应于列表项的 Text 属性。

④ DataValueField 属性：用于指定 DataSource 中的一个字段，该字段的值对应于列表项的 Value 属性。

⑤ Items 属性：表示列表中各个选项的集合，如 DropDownList.Items(i) 表示第 i 个选项，i 从 0 开始。每个选项都有以下 3 个基本属性。

- Text 属性：表示每个选项的文本。
- Value 属性：表示每个选项的选项值。
- Selected 属性：表示该选项是否被选中。

⑥ Count 属性：通过该属性可获得 DropDownList 控件的选项数。

⑦ Add() 方法：通过该方法可以向 DropDownList 控件添加选项。

⑧ Remove() 方法：通过该方法可从 DropDownList 控件中删除指定的选项。

- ⑨ Insert()方法：通过该方法可将一个新的选项插入 DropDownList 控件中。
- ⑩ Clear()方法：通过该方法可以清空 DropDownList 控件中的选项。
- ⑪ SelectedIndex 属性：用于获取下拉列表中选项的索引值。如果未选定任何项，则返回值为-1。
- ⑫ SelectedItem 属性：用于获取列表中的选定项。通过该属性可获得选定项的 Text 属性值和 Value 属性值。
- ⑬ SelectedValue 属性：用于获取下拉列表中选定项的值。
- (2) DropDownList 控件常用的事件说明如下。
- SelectedIndexChanged 事件：当用户选择了下拉列表中的任意选项时，都将引发 SelectedIndexChanged 事件。

### 3.1.4 使用 ListBox 控件

ListBox 控件用于在 Web 页中创建列表框，并允许用户从列表中选择一项或多项。

(1) ListBox 控件常用的属性说明如下。

- ① Rows 属性：表示所能显示的列表项的行数。如果列表项的数目大于 Rows 值，则会自动产生滚动条，以便选择更多的选项。
- ② SelectionMode 属性：指明选择模式，其属性值有下列 4 个。
- MultiSimple：可以简单选择多项。
  - MultiExtended：可以选择多项，并且用户可以使用 Shift 键、Ctrl 键和箭头键来进行多行项选择。
  - None：无法选择。
  - One：只能选择一项。
- ③ SelectedIndex 属性：用于获取列表中索引值最小的选项。如果未选择任何项，则返回值为-1。
- ④ SelectedItem 属性：用于获取列表中索引值最小的选项。通过该属性可获得选项的 Text 和 Value 属性值。
- ⑤ SelectValue 属性：用于获取列表中索引值最小的选项的值。
- (2) ListBox 控件常用的事件说明如下。
- SelectedIndexChanged 事件：当用户选择了列表中的任意选项时，将引发该事件。

## 任务 3.2 熟悉 ASP.NET 内置对象

ASP.NET 提供了 7 个内置对象，这些对象可以在页面中直接使用，通过 ASP.NET 内置对象，在 ASP.NET 页面上以及页面之间可方便地进行获取、输出、传递、保留各种信息等操作，以完成各种复杂的功能。各对象的描述如表 3-1 所示。