

任务：

- (1) 优化窗体控件布局。
- (2) 调整窗体中控件输入次序。
- (3) 窗体控件的特殊设置。
- (4) 新建窗体。
- (5) 完成练习。

3.1 概 述

3.1.1 理解 Visual Studio 2010 对象

Visual Studio 2010 是一个面向对象的应用程序开发环境,它包含大量对象,例如窗体、菜单、数据窗口对象、各种控件等。无论什么对象,它们基本上都具有这样三种特性:属性、对象函数(方法)、事件。个别对象可能只具备其中的一种或两种特性。下面介绍 Visual Studio 2010 对象属性、函数、事件的基本概念。

1. 属性

属性用来描述对象的特征,描述一个对象不同于另一个对象的地方。Visual Studio 2010 的对象也有属性,少至三五个,多达数十个。对可视对象(如窗口、控件、菜单等)来说,查看对象属性的方法很简单:鼠标右击对象,则屏幕上弹出一个弹出式菜单,选择 Properties 菜单项以打开属性对话框,这样对象属性就分门别类地呈现在屏幕上,用户可以根据需要尽情设置和修改。

每个对象都有一个对象名,可以在属性对话框中定义,对象的大多数属性在程序代码中可以动态地修改(不允许修改的属性称作只读属性),其格式为:对象名.属性=属性值。例如,某段程序代码有一行语句为 `cb_find.text="继续查找"`,当执行这一语句后,程序就把按钮 `cb_find` 的标题修改为“继续查找”。其中,`cb_find` 是按钮名称,`text` 是按钮对象的标题属性。

2. 对象函数

在大多数面向对象的编程语言中,也将对象函数称作“方法”。对象函数作用于对象,

其引用方式形如：对象名·函数名(参数 1, 参数 2, …)。当然有的函数可以没有参数, 大多数对象函数都有返回值, 该值往往用来指示函数的执行情况, 现举一个简单的例子, 用 Hide() 函数将名为 cb_yes 的按钮对象隐藏起来, 其调用形式是 cb_yes · Hide ()。

3. 事件

事件是对象的一个重要内在特征, 它是指由用户操作或程序执行所引起相关对象的某些事情发生, 比如当单击菜单对象时, 菜单对象的单击(Clicked)事件就发生了。Visual Studio 2010 的大多数对象都有自己的事件, 应用有应用的事件, 菜单、窗口、控制件也有它们各自的事件, Visual Studio 2010 为这些对象都预先定义了一组事件。针对每个发生的事件, 如果设计人员希望做相应处理的话, Visual Studio 2010 提供了每个事件的响应方法入口, 可在属性窗口的事件列表项中双击所选中的事件, 即可进入事件响应方法入口, 允许用户编写一段代码并存储, 作为事件处理程序。当然, 并非每个发生的事件都需要编写一段事件处理程序, 若不需要, 可不编写。

3.1.2 控件概述

控件是构成 Visual Studio 2010 应用程序图形界面、完成数据输入/输出的强有力的工具, 是应用程序图形界面的重要组成部分。一方面, 控件丰富美化了窗口界面; 另一方面, 控件也为用户操纵应用程序以及应用程序表现数据提供了灵活多样的手段。Visual Studio 2010 的控件基本上分成两类: 一类能够响应用户的操作, 它们具有事件(如按钮、编辑框等控件); 另一类不响应用户的操作, 它们没有事件(如文本框、绘图控件)。无论哪一类控件, 都不能单独保存, 需要开发人员将其放置到窗口上, 与窗口融为一体, 保存到应用库中, 才能发挥其作用。

前面介绍了静态文本框、命令按钮、单行编辑框等控件, 接下来说明控件操控及其他常用控件的概念和用途。

控件操控包括放置控件、控件命名、选择窗体中控件、移动控件、删除控件、对齐控件、控件常用属性的设置、控件的焦点与 Tab 次序的设置等。

将控件放置到窗体上的操作步骤为: ①用窗体画板打开欲放置控件的窗体; ②在 Visual Studio 2010 窗体图标工具栏上, 有一个工具箱图标(图案是锤子与扳手交叉摆列), 单击该图标, 系统会显示出所有的控件图标, 通过单击选择要放置的控件; ③将鼠标指针移动到窗体上要放置所选控件的位置, 单击鼠标左键, 所选控件即被放置到窗体上。控件放置到窗体上后, 系统在状态栏中给出所放控件的名称、位置、大小等信息。

往窗体上放置控件时, 系统自动为控件赋予一个唯一的名称, 该名称由两部分组成: 第一部分是控件的默认名称前缀, 如命令按钮的默认名称前缀是 cb_, 单行编辑框前缀是 sle_; 第二部分是数字标号, 该数字保证控件名称的唯一性。默认的控件名称不直观, 没有反映具体控件的作用, 一般情况下都要对控件重新命名。

最初放置到窗体上的控件, 位置通常不会恰到好处, 控件的大小也是系统默认的尺寸, 不一定符合应用程序的要求, 因此应通过移动控件位置和改变控件的大小来合理布局窗体。当某个控件不再需要时, 可以将其删除。

在窗体上放置了多个控件后,往往需要让控件按某种方式排列整齐,以使应用程序更加美观。

大多数控件都有一个标题,用于向用户提示控件的功用,缺省时,控件的标题由系统自动设置为“None”,开发人员必须予以修改,有些控件我们希望根据程序的运行情况决定用户能否访问它,这时可以通过设置控件的 Visible 或 Enabled 属性来实现。

焦点是接收用户鼠标或键盘输入的能力,当控件有焦点时,才能接收用户的输入。在窗体上放置多个控件后,程序运行时哪些控件能获得输入焦点、哪些控件不能获得输入焦点,以及用户按键盘上的 Tab 键时从一个控件跳到哪一个控件的次序等问题对应用程序来说都是至关重要的。控件或窗口得到焦点的途径有三条:①运行时用户用鼠标选择对象;②运行时用户使用访问键选择对象;③在某事件代码中使用对象函数 SetFocus()使对象获得焦点。

将控件放置到窗体上后,Visual Studio 2010 自动给每个能够响应用户操作的控件赋予一个 TabIndex 值为 0,1,2,⋯,该值决定了默认情况下程序运行时用户按 Tab 键输入焦点的跳转次序,如果默认的跳转次序不能满足应用程序的需要,可重新设置。

编辑屏蔽控件 MaskedTextBox 是一种“聪明”的单行编辑框,使用户按指定的格式输入数据,例如,输入某人的出生日期时要让用户按“日/月/年”的格式输入,此时,可以使用格式为 DD/MM/YY 的编辑屏蔽控件。

列表框 ListBox 在一个矩形框,可以显示多行文本,用列表的方式为用户提供多种选择。

下拉列表框 ComboBox 又叫作组合框,它把单行编辑框与列表框的功能结合起来,具有这两种控件的特点。

单选钮 RadioButton、复选框 CheckBox 是为用户做出抉择而设计的,单选钮用来表示一组互斥的选项,用户只能从中选择一项。复选框用来表示是/否或真/假两种选择,常常成组使用,同一组中可以有多个复选框被选中。单选钮、复选框的分组通过组框 GroupBox 来实现,组框与单选钮、复选框常常同时出现。组框有两个用途:一是装饰界面,二是对单选钮进行分组,这样在一个窗体中就可以有几组单选钮。

标签控件 Tab Control 是在 Visual Studio 2010 开发环境中使用最多的控件之一,当设置某个对象的属性时,打开的属性对话框使用的就是标签控件。标签控件能够把多种数据和程序选项封装在一起,使用户通过简单的操作访问这些数据和选项。一般来说,标签控件由多个标签页组成。用户能够看到的总是当前标签页。

数据窗口是 Visual Studio 2010 提供给开发人员快速开发应用程序的强有力的工具,数据窗口包括两个方面的内容:数据窗口对象和数据窗口控件。数据窗口对象主要以多种形式表现数据,并允许用户更新数据;数据窗口对象通过数据窗口控件在用户界面上展现数据,即先在窗口上放置数据窗口控件,再将数据窗口对象连接到数据窗口控件上。

特别提示:对某 MaskedTextBox 控件设置日期时,为了使设置能适应各操作系统和数据库系统的环境,建议在该控件属性的 Mask 标签页中做如下设置:设置 Type 为 Date,设置 Masks 为 [date]-International,设置 Mask 为 yyyy-mm-dd。

3.2 实 验

实验一 优化窗体控件布局

(1) 进入 Stucour 窗体应用项目后,选择 w_student 窗体设计(双击项目中的 w_student.cs),并完成如图 3.1 所示的窗口。其中,“出生年月”右边的控件为 Toolbox 窗口的 Common Controls 中的 MaskedTextBox 控件,按钮为 Button 控件,下方深灰色矩形框为 Toolbox 窗口的 Data 中的 DataGridView 控件(用来显示数据)。

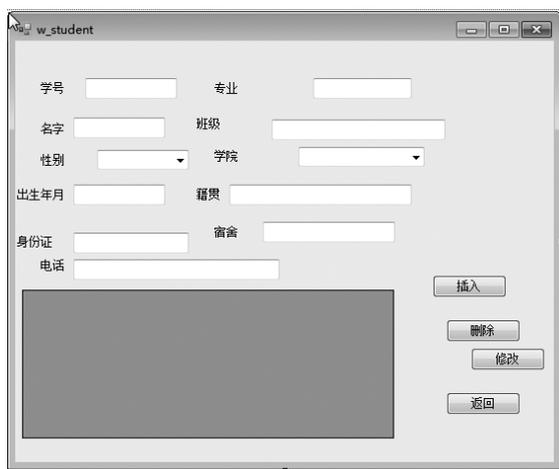


图 3.1 控件分布界面示意图

(2) 如图 3.2 所示,单击“学号”的 Label 控件后,按 Ctrl 键,并用鼠标逐个单击图中所示的各个 Label 控件,最后单击 Make Vertical Spacing Equal 图标。

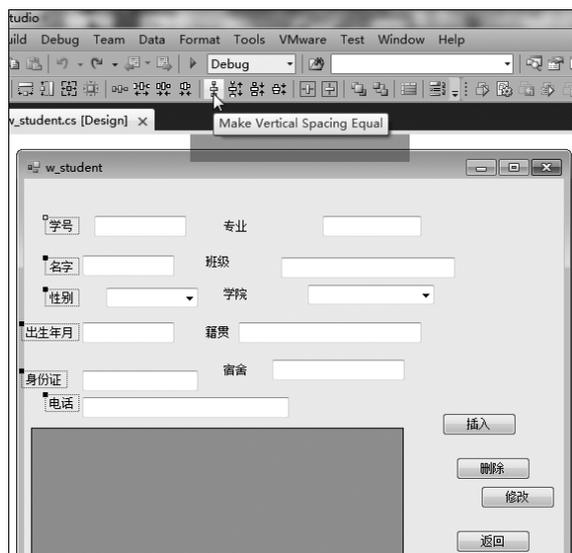


图 3.2 垂直居中示意图

(3) 如图 3.3 所示,单击 Align Rights 图标。

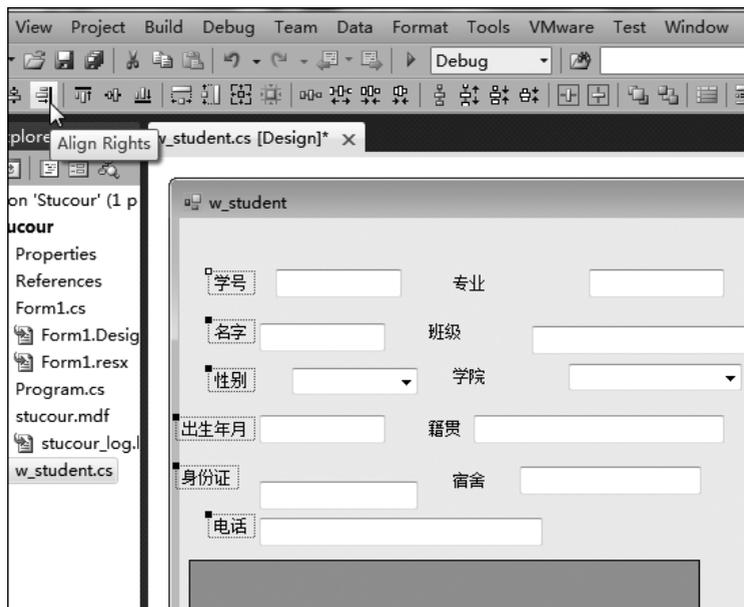


图 3.3 控件右对齐示意图

(4) 如图 3.4 所示,单击“学号”右边的 TextBox 控件后,按 Ctrl 键,并逐个单击图中所示的各个 TextBox 控件,最后单击 Make Vertical Spacing Equal 图标。



图 3.4 控件多选示意图

(5) 如图 3.5 所示,单击 Align Lefts 图标,再单击 Make Same Width 图标。

(6) 如图 3.6 所示,类似于上述操作方法,得到窗体整齐排列的控件。

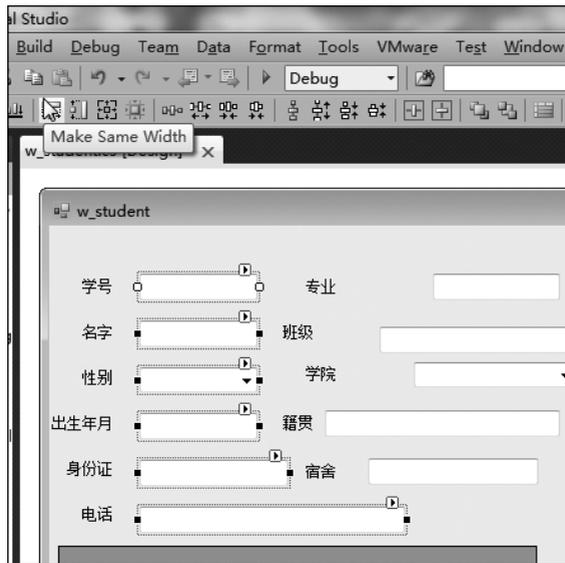


图 3.5 右对齐以及宽度相等示意图



图 3.6 窗口调整结果图

实验二 调整窗体中控件输入次序

(1) 如图 3.7 所示,单击第一个输入控件(学号右边的 TextBox 控件),并在屏幕右下角的 Properties 窗口中选择 TabIndex 项,在其对应输入区输入“1”。

(2) 单击右边的第一个输入控件(“专业”右边的 TextBox 控件),并在屏幕右下角的 Properties 窗口中选择 TabIndex 项,在其对应输入区输入“2”。以此类推,单击“名字”右

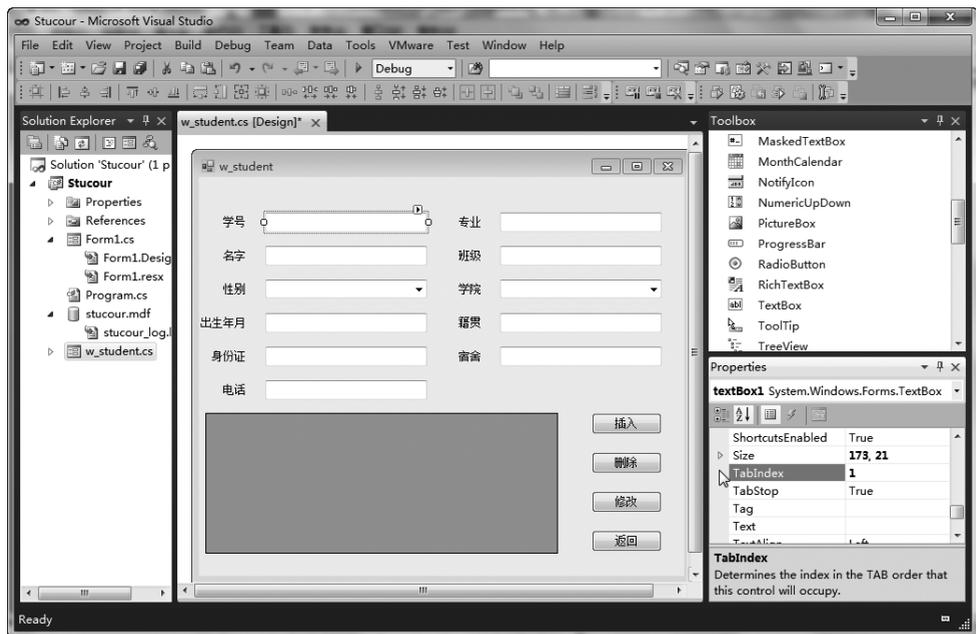


图 3.7 TabIndex 位置示意图

边的 TextBox 控件, Properties 窗口中选择 TabIndex 项, 在其对应输入区输入“3”, …, 单击“电话”右边的 TextBox 控件, Properties 窗口中选择 TabIndex 项, 在其对应输入区输入“11”, 如图 3.8 所示。当该窗体运行时数据输入次序是以 TabIndex 所标记的值从小到大的次序(即在输入时, 按 Tab 键时, 则以 TabIndex 所标记的值从小到大的次序切换)。

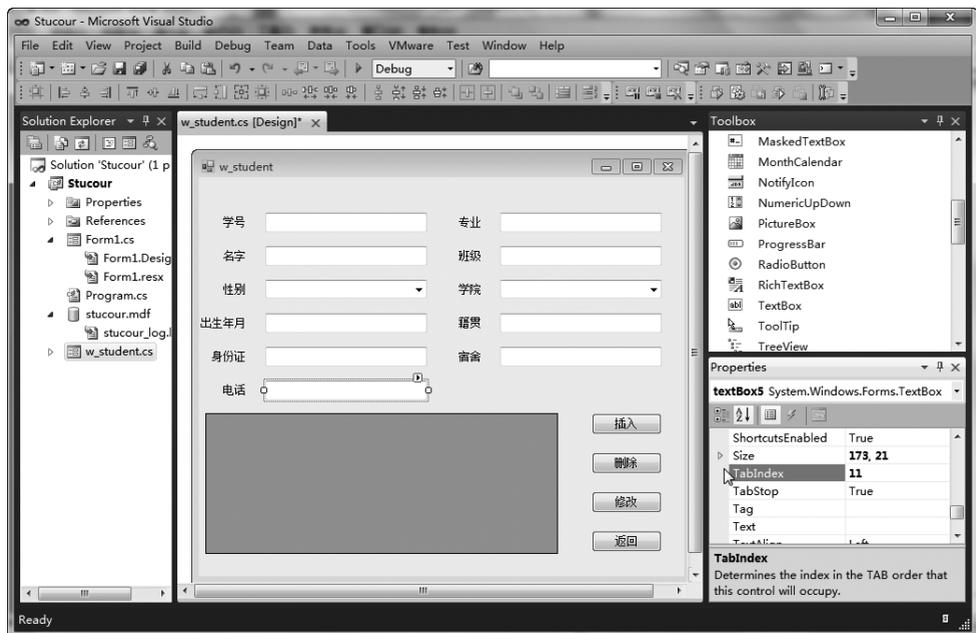


图 3.8 电话的 TabIndex 属性值

实验三 窗体控件的特殊设置

(1) 如图 3.9 所示,单击“出生年月”所指的 MaskedTextBox 控件,并单击 Properties 窗口中 Mask 项右边的小图标。

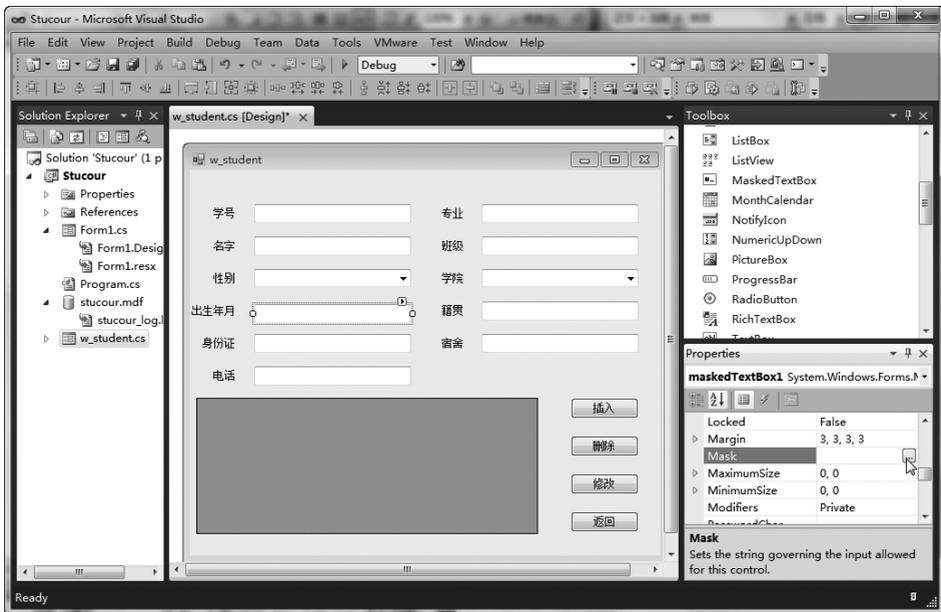


图 3.9 MaskedTextBox 控件位置示意图

(2) 如图 3.10 所示,在弹出的 Input Mask 对话框的掩码列表中选择“短日期格式”,并单击 OK 按钮。

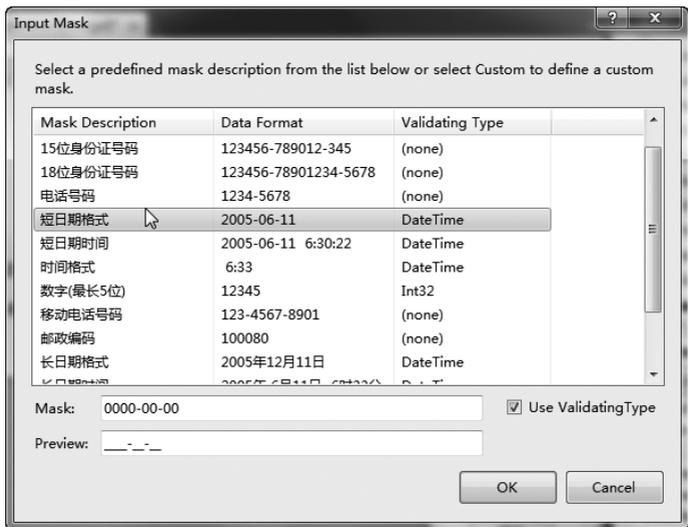


图 3.10 短日期格式设置

(3) 如图 3.11 所示,单击“性别”所指的 ComboBox 控件,并单击 Properties 窗口中 Items 项右边的小图标。

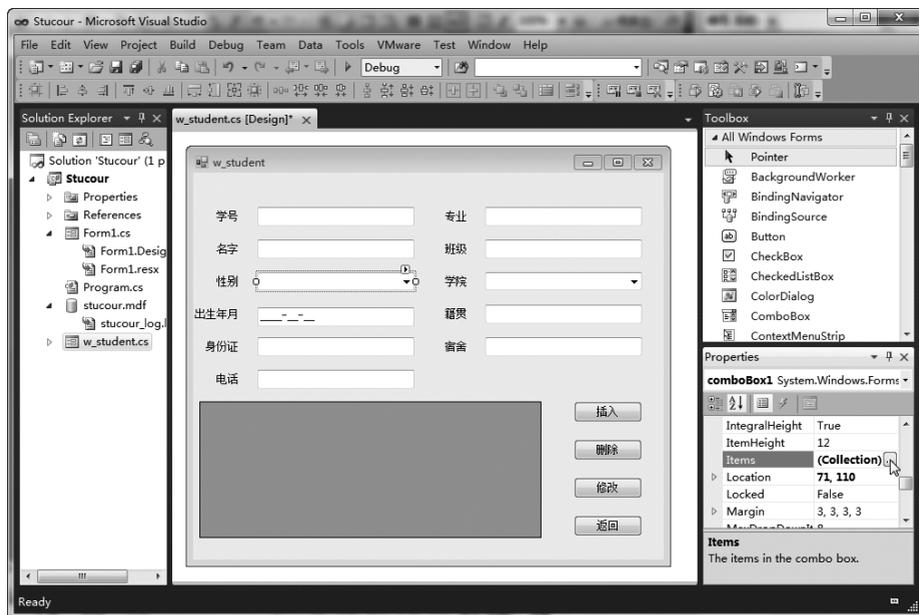


图 3.11 ComboBox 控件位置示意图

(4) 如图 3.12 所示,在弹出的对话框中输入“男”“女”,并单击 OK 按钮。

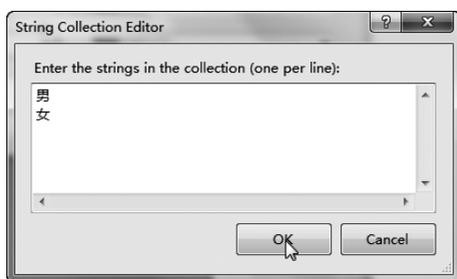


图 3.12 Item 内容填写示意图

(5) 类似前两步操作,在“学院”所指的 ComboBox 控件 Items 项中输入“管理学院”“经济学院”“理学院”,如图 3.13 所示。

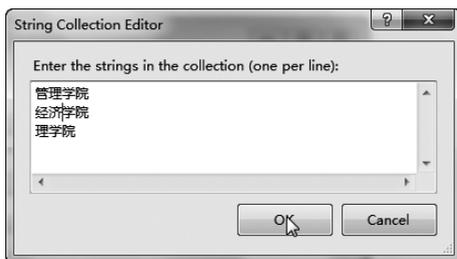


图 3.13 学院的 ComboBox 控件填写示意图

实验四 新建窗体

(1) 如图 3.14 所示,选择菜单 Project→Add Windows Form。

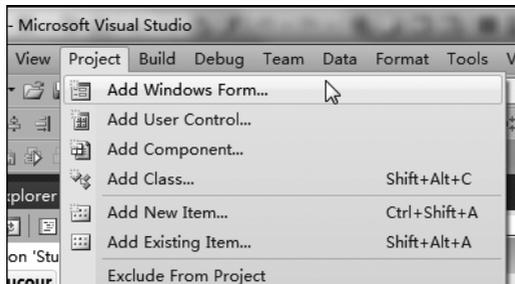


图 3.14 创建 Windows 窗体示意图

(2) 如图 3.15 所示,在弹出的对话框中选择 Windows Form,并在 Name 输入项中输入“w_systable.cs”,单击 Add 按钮。

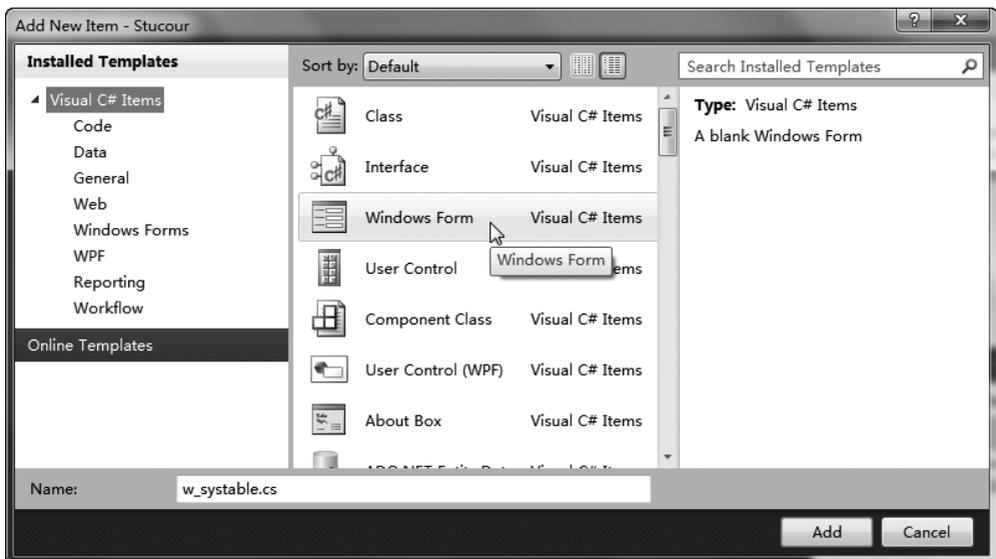


图 3.15 添加 Windows Form 窗体

(3) 如图 3.16 所示,将窗体拉伸,以方便添加控件。

(4) 如图 3.17 所示,单击屏幕右边 Toolbox 中 Containers 中的 TabControl 控件,并单击 w_systable 窗体空白处。

(5) 将 TabControl 控件拉伸,并单击 Properties 窗口中 TabPages 项右边的小图标,如图 3.18 所示。

(6) 单击左边列表项中的 tabPage1,单击弹出式窗口右边属性项的 Text,输入对应值“学院”。单击左边列表项中的 tabPage2,单击弹出式窗口右边属性项的 Text,输入对应值“班级”。然后,单击 OK 按钮。(在图 3.19 中,在左边 Members 下方,可以单击 Add

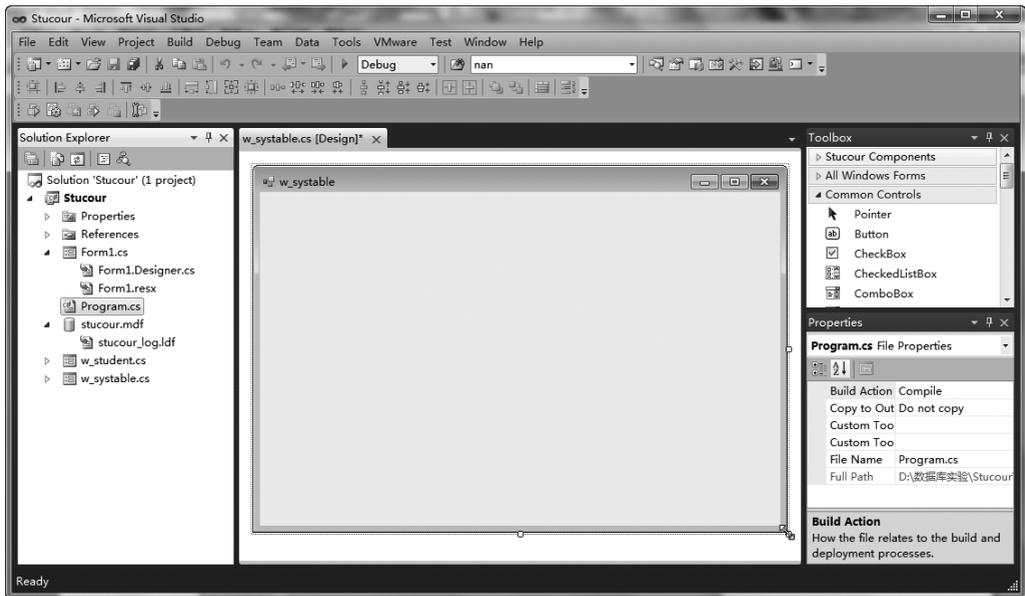


图 3.16 窗体拉伸

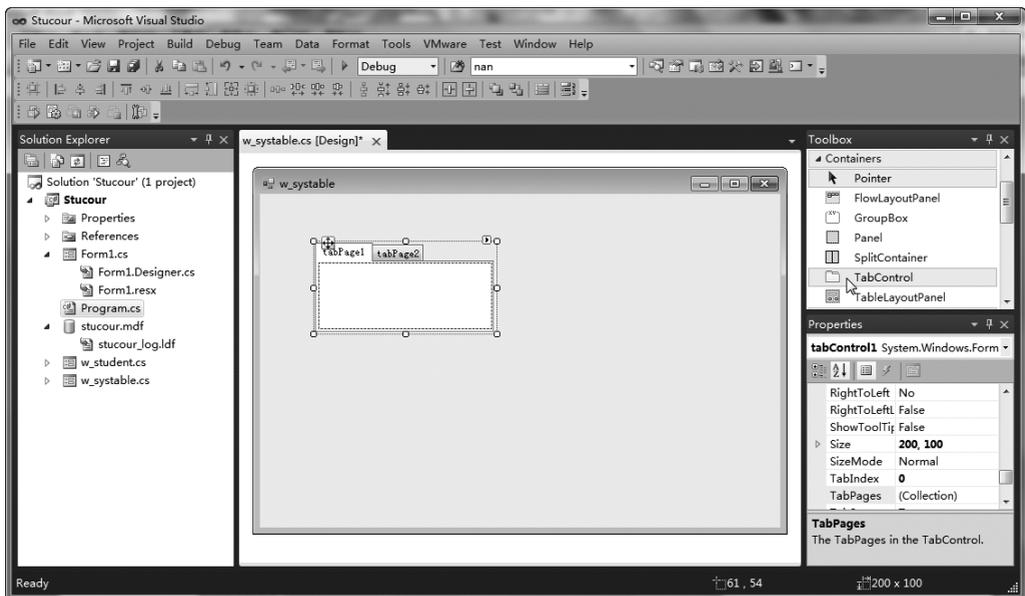


图 3.17 添加 TabControl 控件示意图

按钮,添加新的 tabPage,如单击一下将生成 tabPage3),如图 3.19 所示。

(7) 单击屏幕右边 Toolbox 中的 Data 中的 DataGridView 项,并单击 w_systable 窗体空白处,然后拉伸 DataGridView,如图 3.20 所示。

(8) 如图 3.21 所示,为窗体添加 4 个 Button 控件,并修改其 Text。

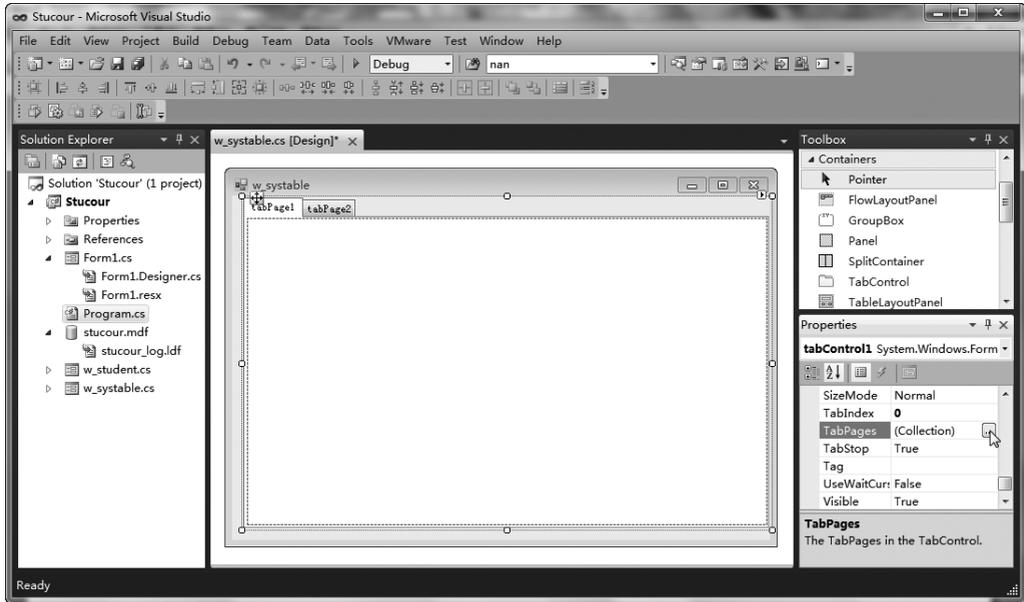


图 3.18 TabPages 位置示意图

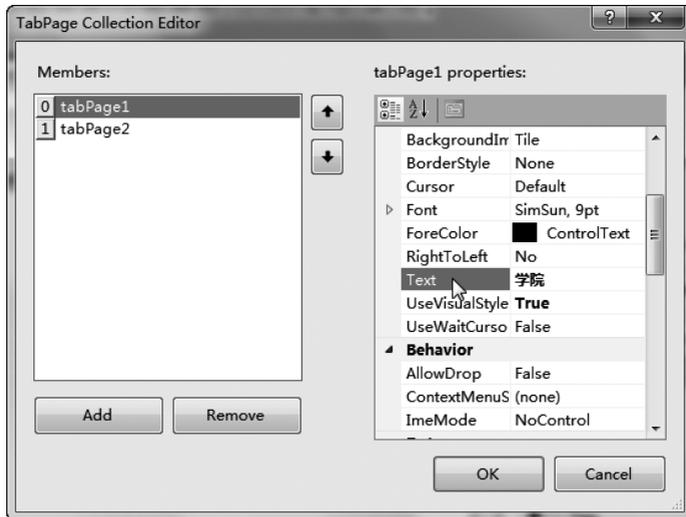


图 3.19 添加新的 tabPage 示意图

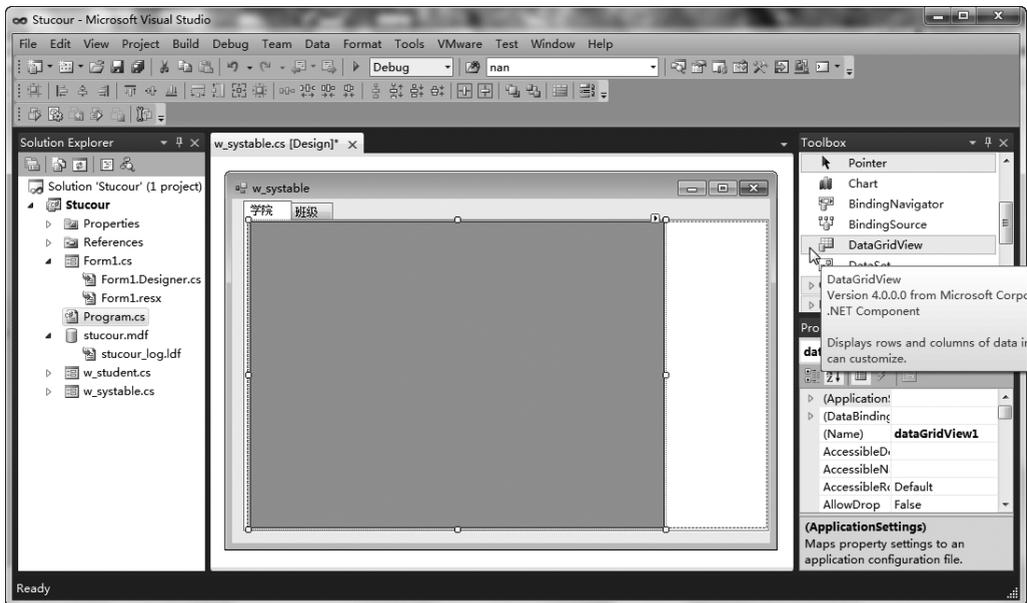


图 3.20 拉伸 DataGridView 效果示意图

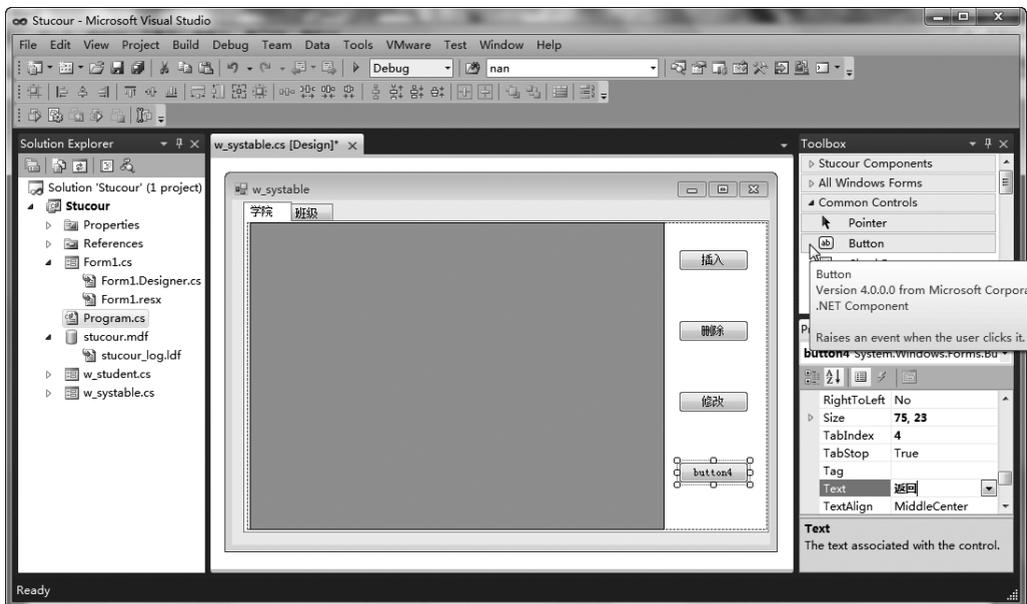


图 3.21 添加 Button 控件效果示意图

实验五 完成练习

要求完成 Stucour 应用的所有窗口并把所有窗口连到菜单上。

1. w_clastu 窗体

说明：该窗体挂在“报表输出”菜单的子菜单项“各班学生列表”上，即要进入“报表输出”菜单的“各班学生列表”子菜单的 Click 事件代码编辑区，插入下列三行代码。

```
Form a =new w_course();  
a.MdiParent =this;  
a.Show();
```

其中，“选择输出范围”为 Toolbox 中 Containers 类型的 GroupBox 控件，“班级”为 Toolbox 中 Common Controls 类型的 CheckBox 控件，如图 3.22 所示。

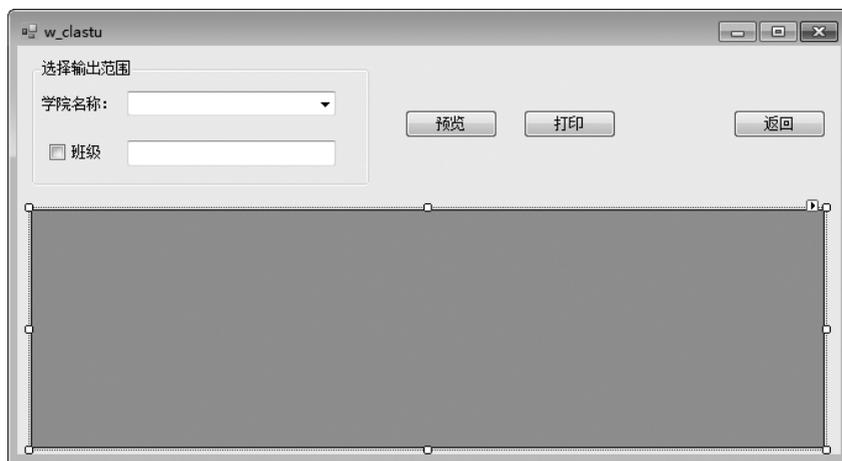


图 3.22 w_clastu 窗体示意图

2. w_collcour 窗体

说明：该窗体挂在“报表输出”菜单的子菜单项“各院课程列表”上，即要进入“报表输出”菜单的“各院课程列表”子菜单的 Click 事件代码编辑区，插入下列三行代码。

```
Form a =new w_collcour();  
a.MdiParent =this;  
a.Show();
```

w_collcour 窗体如图 3.23 所示。

3. w_stugrade 窗体

说明：该窗体挂在“报表输出”菜单的子菜单项“学生各科成绩表”上，即要进入“报表输出”菜单的“学生各科成绩表”子菜单的 Click 事件代码编辑区，插入下列三行代码。

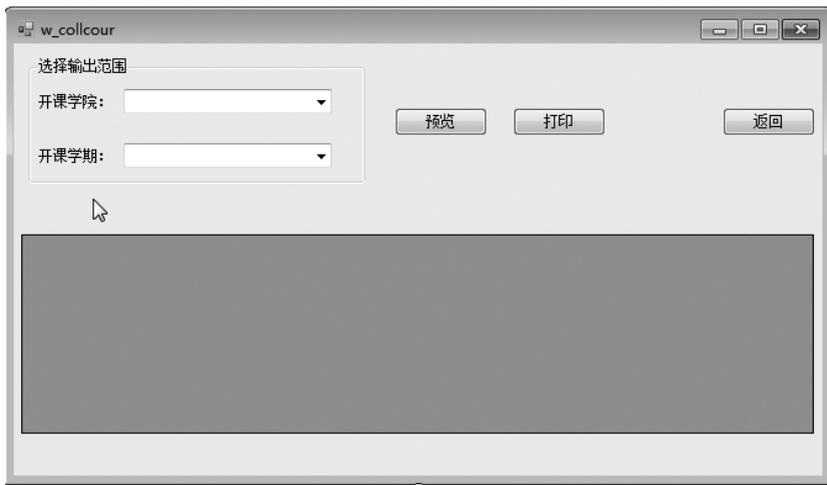


图 3.23 w_collcour 窗体

```
Form a =new w_stugrade();
a.MdiParent =this;
a.Show();
```

w_stugrade 窗体如图 3.24 所示。

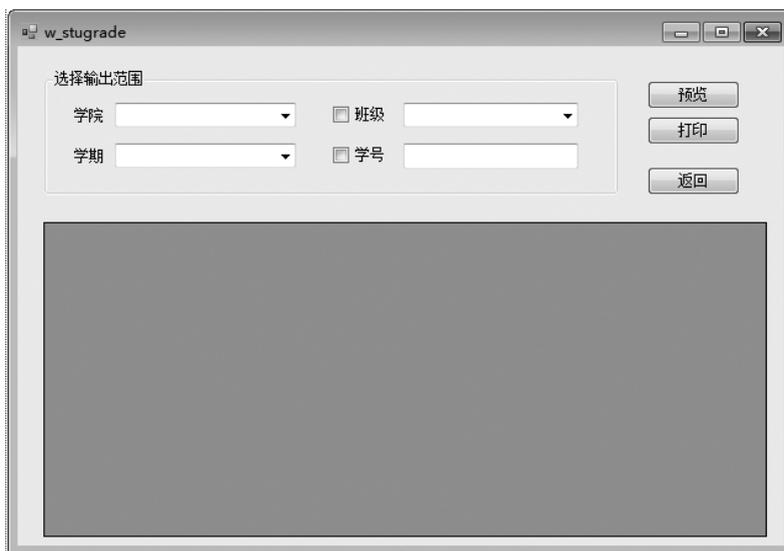


图 3.24 w_stugrade 窗体

4. w_courgrade 窗体

说明：该窗体挂在“报表输出”菜单的子菜单项“各科学生成绩表”上，即要进入“报表输出”菜单的“各科学生成绩表”子菜单的 Click 事件代码编辑区，插入下列三行代码。

```
Form a =new w_courgrade();  
a.MdiParent =this;  
a.Show();
```

w_courgrade 窗体如图 3.25 所示。

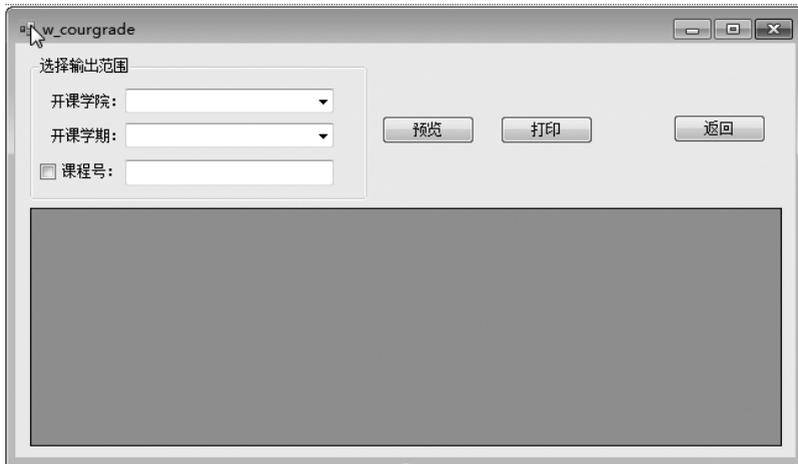


图 3.25 w_courgrade 窗体

5. w_courmark 窗体

说明：该窗体挂在“数据输入”菜单的子菜单项“课程成绩”上，即要进入“数据输入”菜单的“课程成绩”子菜单的 Click 事件代码编辑区，插入下列三行代码。

```
Form a =new w_courmark();  
a.MdiParent =this;  
a.Show();
```

w_courmark 窗体如图 3.26 所示。



图 3.26 w_courmark 窗体

6. w_course 窗体

说明：该窗体挂在“数据输入”菜单的子菜单项“课程数据”上，即要进入“数据输入”菜单的“课程数据”子菜单的 Click 事件代码编辑区，插入下列三行代码。

```
Form a =new w_courmark();  
a.MdiParent =this;  
a.Show();
```

w_course 窗体如图 3.27 所示。



图 3.27 w_course 窗体

7. w_enroll 窗体

说明：该窗体挂在“数据输入”菜单的子菜单项“学生选课数据”上，即要进入“数据输入”菜单的“学生选课数据”子菜单的 Click 事件代码编辑区，插入下列三行代码。

```
Form a =new w_enroll();  
a.MdiParent =this;  
a.Show();
```

w_enroll 窗体如图 3.28 所示。

8. w_systable 窗体(此窗体已在示例中完成一部分)

说明：该窗体挂在“数据输入”菜单的子菜单项“初始设置”上，即要进入“数据输入”菜单的“初始设置”子菜单的 Click 事件代码编辑区，插入下列三行代码。

```
Form a =new w_systable();  
a.MdiParent =this;  
a.Show();
```



图 3.28 w_enroll 窗体

w_systable 窗体如图 3.29~图 3.31 所示。

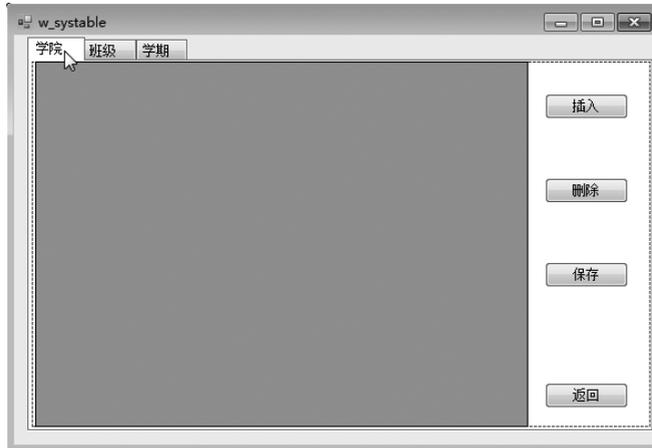


图 3.29 w_systable 窗体的“学院”选项卡

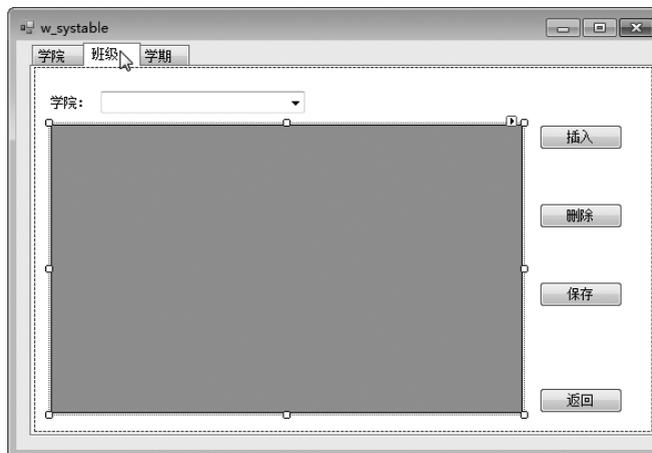


图 3.30 w_systable 窗体的“班级”选项卡

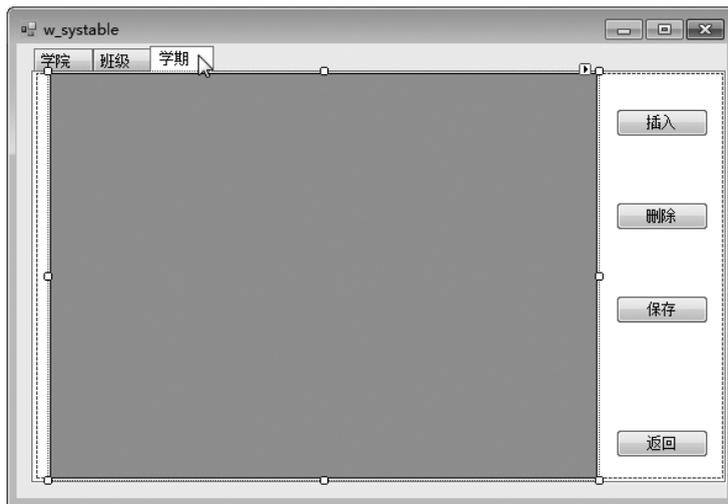


图 3.31 w_systable 窗体的“学期”选项卡

任务：

- (1) DataGridView 连接与数据库表的连接。
- (2) 有引用关系的表的关系定义。
- (3) 创建 student、enroll 及 course 连接的视图。
- (4) 有引用关系表的数据源定义及数据显示。
- (5) 完成练习。

4.1 概 述

4.1.1 何为 DataGridView

DataGridView 控件,也称为数据网格视图控件,支持各种数据源的表格式数据显示。通过 DataGridView 控件,可以显示和编辑表格式的数据,而这些数据可以取自多种不同类型的数据源。DataGridView 控件具有很高的可配置性和可扩展性,提供了大量的属性、方法和事件,可以用来对该控件的外观和行为进行自定义。当需要在 Form 应用程序中显示表格式数据时,可以优先考虑 DataGridView。如果要在小型网格中显示只读数据,或者允许用户编辑数以百万计的记录,DataGridView 将提供一个易于编程和良好性能解决方案。

DataGridView 的主要功能如下。

- (1) 支持多种列类型,这些列类型能够满足大部分常见需要,且易于扩展或替换。
- (2) 实现多种数据显示方式,即 DataGridView 能够显示非绑定的数据、绑定的数据源,或者同时显示绑定和非绑定的数据,以及自定义的数据等。
- (3) 提供了自定义数据显示的多种方式,包括很多属性和事件,用于数据的格式化和显示。
- (4) DataGridView 使用户能够以多种方式操作单个网格组件。例如,可以冻结行和列,避免它们因滚动而不可见;隐藏行、列、表头;改变行、列、表头尺寸的调整方式等。

4.1.2 在实验中使用 DataGridView

在运用 Visual Studio 2010 进行数据库系统开发过程中,Visual Studio 2010 数据网