

第 5 章 学生成绩管理系统—— 学生登录功能设计

本章要点

- 基本控件的使用：ListBox 控件、ComboBox 控件。
- 消息框的使用。
- ADO.NET 访问数据库。

学习目标

- 理解基本控件的属性及用法。
- 学会使用基本控件。
- 掌握消息框的用法。
- 理解 ADO.NET 概念。
- 掌握 ADO.NET 对象的应用。
- 会使用 ADO.NET 技术访问数据库。

5.1 本章任务

本章任务是完成学生成绩管理系统登录页面中学生登录功能的设计，登录窗体运行效果如图 5-1 所示。在该窗体中给出两种用户登录功能。



图 5-1 学生成绩管理系统登录窗体

学生登录功能：

对用户类别、用户账号和密码进行验证,验证通过之后登录系统进入学生主窗体(如图 5-2 所示)。需要输入的内容有用户类别、用户账号和密码。



图 5-2 学生主窗体

若输入内容出错则会出现以下几种提示信息,见图 5-3。

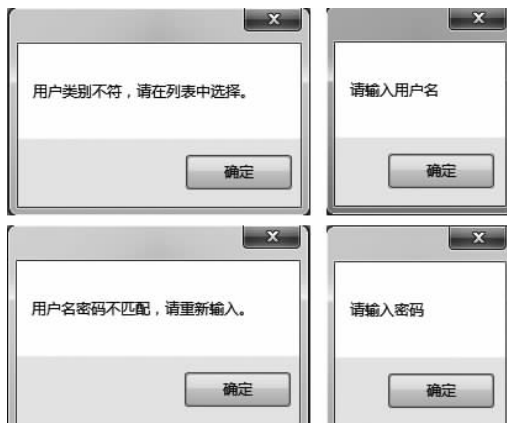


图 5-3 错误提示对话框

5.2 准备工作

学生登录功能涉及的知识内容包括控件、访问数据库。本章从这两个方面进行知识准备。

5.2.1 列表框和组合框

列表框(ListBox)和组合框(ComboBox)都提供一系列选项,用户可以从中做出选择。列表框可以一次选择一个或多个选项,而组合框只能选择一个选项,但可以在 ComboBox 的 TextBox 部分输入新选项。

1. 显示样式

当向窗体中添加列表框控件(ListBox)时,根据可用空间及控件中的选项数量呈现不同的显示效果,如图 5-4 所示。

当向窗体中添加组合框控件(ComboBox)时,通过设置该控件的 DropDownStyle 属性呈现不同的样式,具体如下:

Simple: 使得 ComboBox 的列表部分总是可见的,如图 5-5(a)所示。

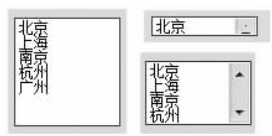


图 5-4 ListBox 控件的外观样式

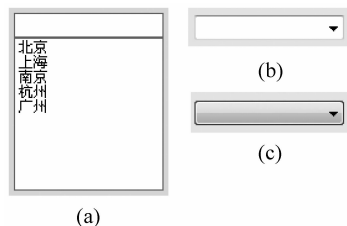


图 5-5 ComboBox 控件的几种外观样式

DropDown: 这个是默认值,使得用户可以编辑 ComboBox 控件的文本框部分,必须单击右侧的箭头才可以显示列表部分,如图 5-5(b)所示。

DropDownList: 外观与 DropDown 的一样,不用的是用户不能编辑 ComboBox 控件的文本框部分,如图 5-5(c)所示。

2. Items 集合

显示在列表框或组合框中的选项列表,属于集合。可以通过从 0 开始的索引引用集合中的选项。

(1) 填充列表

可以使用多种方法来填充列表框和组合框的 Items 集合。

- 使用【属性】窗口

在设计时填充列表。如图 5-6 所示,属性窗口中定义 Items 集合,单击 Items 属性右边的省略号按钮,打开【字符串集合编辑器】对话框,输入列表项,单击【确定】按钮完成填充列表操作,见图 5-7。

- 使用 Items.Add 方法

在运行时添加选项。使用 Items 集合的 Add 方法,该方法的通式:

```
Object.Items.Add(ItemValue);
```

ItemValue 是添加到列表中的新选项,新选项通常出现在列表的最后。

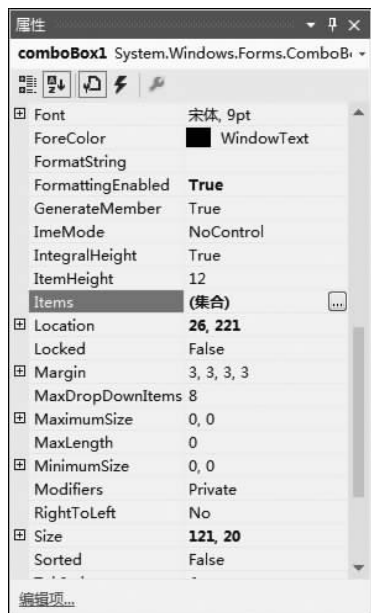


图 5-6 属性窗口



图 5-7 字符串集合编辑器

【实例 5-1】 Items.Add 方法示例。

```
ComboBox1.Items.Add("杭州");
ComboBox1.Items.Add("广州");
ListBox1.Items.Add(textBox1.Text);
ListBox1.Items.Add("西安");
```

- 使用 Items.Insert 方法

Insert 方法可以在列表中的某个位置插入新选项。索引位置从 0 开始。

Items.Insert 方法的通式：

```
Object.Items.Insert(Position, ItemValue);
```

【实例 5-2】 Items.Insert 方法示例。

```
ComboBox1.Items.Insert(2, "南京");
ListBox1.Items.Insert(3, textBox1.Text);
```

(2) 删除选项

通过指定选项的索引或文本,可以从列表中删除单独的选项。

- 使用 Items.RemoveAt 方法

通过指定列表项的索引删除列表项。

Items.RemoveAt 方法的通式：

```
Object.Items.RemoveAt(IndexPosition);
```

【实例 5-3】 Items.RemoveAt 方法示例。

```
ComboBox1.Items.RemoveAt(2); //删除 ComboBox1 组合框中的第 3 项  
ListBox1.Items.RemoveAt(0); //删除 ListBox1 列表框中的第 1 项
```

- 使用 Items.Remove 方法

通过指定列表项的文本删除列表项。

Items.Remove 方法的通式：

```
Object.Items.Remove(TextString);
```

【实例 5-4】 Items.Remove 方法示例。

```
ComboBox1.Items.Remove("南京"); //删除 ComboBox1 组合框中的"南京"  
ListBox1.Items.Remove("西安"); //删除 ListBox1 列表框中的"西安"
```

(3) 清除列表

除了删除单独的列表项以外,还可以删除列表中的全部项。

- 使用 Items.Clear 方法

Items.Clear 方法的通式：

```
Object.Items.Clear();
```

【实例 5-5】 Items.Clear 方法示例。

```
ComboBox1.Items.Clear(); //清除 ComboBox1 组合框  
ListBox1.Items.Clear(); //清除 ListBox1 列表框
```

Items.Count 属性：用来确定列表中的列表项的数量。

【实例 5-6】 使用 for 循环遍历列表框中所有项。

```
for(int i=0;i<ListBox1.Items.Count; i++)  
{  
}
```

引用 Items 集合：

列表中的第一个元素的索引是 0,因此最大的索引是 Items.Count-1。

【实例 5-7】 引用 Items 集合示例。

```
ComboBox1.Items[0]="上海";  
messageLabel.Text=ComboBox1.Items[1].ToString();
```

SelectedIndex 属性：

当程序运行时,如果用户从列表中选择了某个列表项,则该选项的索引编号将被存储在列表框的 SelectedIndex 属性中。如果列表框没有被选中,则 SelectedIndex 属性将被设置为-1。

【实例 5-8】

```
ComboBox1.SelectedIndex=3; //组合框中选择第 4 项  
messageLabel.Text=ComboBox1.Items[ComboBox1.SelectedIndex].ToString();
```

```
//在 messageLabel 标签中显示组合框中当前选中项
```

3. 列表框和组合框的事件

下面介绍列表框和组合框的几个常用的事件。

(1) TextChanged 事件

当用户在组合框的文本框部分输入文本时,TextChanged 事件发生。列表框没有 TextChanged 事件,因为列表框中没有相关的文本框。

(2) Enter 事件

当控件接收到焦点时,Enter 事件发生。当用户使用 Tab 键从一个控件跳转到另一个控件时,每个控件上都会发生 Enter 事件。

(3) Leave 事件

Leave 事件发生在控件丢失焦点之时。Leave 事件的处理程序经常被用于验证输入数据的有效性。当用户从一个控件跳转到另一个控件时,Leave 事件将在下一个控件的 Enter 事件之前被触发。

5.2.2 消息框的使用

当用户输入无效数据或遗漏输入所要求的数据值时,需要程序显示提示消息。可以在消息框(一种特殊类型的窗口)中,向用户显示消息。

使用 MessageBox 对象的 Show 方法显示消息框。MessageBox 对象是预定义的 MessageBox 类的实例,可以在任何需要显示消息的时候使用该对象。

MessageBox 对象的通式:

```
MessageBox.Show(text);
MessageBox.Show(text,caption);
MessageBox.Show(text,caption,MessageBoxButtons);
MessageBox.Show(text,caption,MessageBoxButtons,MessageBoxIcon);
```

其中,text 是显示在消息框中的消息,caption 是显示在消息框窗口的标题栏信息,MessageBoxButtons 成员参数(如表 5-1 所示)指定要显示的按钮,MessageBoxIcon 成员参数(如表 5-2 所示)执行要显示的图标。

表 5-1 MessageBoxButtons 成员参数

| 成员名称 | 说明 |
|------------------|-----------------------|
| AbortRetryIgnore | 消息框包含“中止”、“重试”和“忽略”按钮 |
| OK | 消息框包含“确定”按钮 |
| OKCancel | 消息框包含“确定”和“取消”按钮 |
| RetryCancel | 消息框包含“重试”和“取消”按钮 |
| YesNo | 消息框包含“是”和“否”按钮 |
| YesNoCancel | 消息框包含“是”、“否”和“取消”按钮 |

表 5-2 MessageBoxIcon 成员参数

| 成员名称 | 说 明 |
|-------------|--|
| Asterisk | 该消息框包含一个符号,该符号是由一个圆圈及其中的小写字母 i 组成的 |
| Error | 该消息框包含一个符号,该符号是由一个红色背景的圆圈及其中的白色 X 组成的 |
| Exclamation | 该消息框包含一个符号,该符号是由一个黄色背景的三角形及其中的一个感叹号组成的 |
| Hand | 该消息框包含一个符号,该符号是由一个红色背景的圆圈及其中的白色 X 组成的 |
| Information | 该消息框包含一个符号,该符号是由一个圆圈及其中的小写字母 i 组成的 |
| None | 消息框未包含符号 |
| Question | 该消息框包含一个符号,该符号是由一个圆圈和其中的一个问号组成的 |
| Stop | 该消息框包含一个符号,该符号是由一个红色背景的圆圈及其中的白色 X 组成的 |
| Warning | 该消息框包含一个符号,该符号是由一个黄色背景的三角形及其中的一个感叹号组成的 |

【实例 5-9】 消息框显示效果示例,如图 5-8~图 5-11 所示。

```

MessageBox.Show("请输入密码");
MessageBox.Show("请输入密码","未输入");
MessageBox.Show("确认删除","提示",MessageBoxButtons.OKCancel);
MessageBox.Show("确认删除","提示",MessageBoxButtons.OKCancel,
MessageBoxIcon.Question);

```



图 5-8



图 5-9



图 5-10



图 5-11

5.2.3 ADO.NET 访问数据库文件

目前的大多数数据处理技术和关系数据库文件有关。本书学生成绩管理系统中使用的数据库也存储在名为 ScoreDB 的 SQL Server 关系数据库中。本节介绍如何使用 ADO.NET 技术访问数据库。

1. ADO.NET 简介

ADO.NET 又被称为 ActiveX 数据对象(ActiveX Data Object),允许用户访问很多格式的数据库数据。提供的基本类型有 OLEDB、应用与 SQL Server 的 SQLClient、ODBC 和 Oracle。利用 OLEDB 可以从 Access、Oracle、Sybase 或 DB2 这样的数据源获得数据。本书中的示例将使用 Microsoft 公司的 SSE(SQL Server Express),它将随同 Visual Studio 自动安装。

ADO.NET 对象模型中有 5 个主要的组件,分别是 Connection、Command、DataAdapter、DataSet 以及 DataReader。

2. ADO.NET 体系结构

ADO.NET 支持断开模型,将数据访问与数据处理分离。它是通过两个主要的组件:.NET 数据提供程序和 DataSet 来完成这一操作的。

ADO.NET 体系结构的一个核心元素是 .NET 数据提供程序,它是专门为数据处理以及快速地只进、只读访问数据而设计的组件。它包括 Connection、Command、DataReader 和 DataAdapter 对象的组件。

表 5-3 ADO.NET 体系结构

| 对象名称 | 功 能 | 连接 SQLServer 数据库的对象名称 |
|-------------|---|-----------------------|
| Connection | 提供与数据源的连接 | SqlConnection |
| Command | 用于操作数据库的命令,通常为 SQL 语句或存储过程 | SqlCommand |
| DataReader | 从数据源中提供只进、只读的数据流 | SqlDataReader |
| DataAdapter | 提供连接 DataSet 对象和数据源的桥梁,将数据加载到 DataSet 中,并对 DataSet 中数据的更改与数据源保持一致 | SqlDataAdapter |

DataSet 称为数据集,是指一种由数据所组成的集合。可以把 DataSet 当成内存中的数据库,DataSet 是不依赖于数据库的独立数据集合。独立是指即使断开数据库连接,DataSet 依然是可用的。DataSet 包含一个或多个 DataTable 对象的集合,这些对象由数据行和数据列以及有关 DataTable 对象中数据的主键、外键、约束和关系信息组成。

3. 连接 SQL Server 数据库

连接数据库时,需要在连接字符串中给出 SQL Server 服务器的名称、连接方式、数据库名称等内容。

【实例 5-10】 创建数据库连接。连接到本机的 ScoreDB 数据库,该数据采用信任连接方式。

```
string constr="Data Source=.;Integrated Security=True;database=ScoreDB";
SqlConnection sqlConn=new SqlConnection(constr);
sqlConn.Open();
```

其中,Data Source 指定了 SQL Server 服务器的名称,localhost 或. 表示为本机;Integrated Security 表示采用信任连接方式,即数据库采用 Windows 身份验证;database 表示连接数据库的名称,这里登录的数据库为 ScoreDB。

如果没有采用 Windows 身份验证,则需要在连接字符串中指定 uid 和 pwd。登录 SQL Server 数据库会对此用户的 ID 和口令进行验证。连接字符串如下:

```
string constr="Data Source=.;database=ScoreDB;uid=admin;pwd=123";
```

注意: SqlConnection、SqlCommand、SqlDataReader 和 SqlDataAdapter 在 System.Data.SqlClient 名称空间中,所以在程序中要导入该名称空间: using System.Data.SqlClient;。

4. 读取数据

连接到数据库后,可以读取和操作数据库中的数据。SqlCommand 类可以用来对 SQL Server 数据库执行一个 Transact-SQL 语句或存储过程。

SqlCommand 类常用属性:

- CommandText 属性: 用于获取或设置要对数据源执行的 T-SQL 语句或存储过程。
- CommandTimeout 属性: 用于设置获取或设置在终止执行命令的尝试并生成错误之前的等待时间。
- CommandType 属性: 用户获取或设置该命令的类型,默认是 T-SQL 语句。

SqlCommand 命令对象提供以下几个基本方法来执行命令。

- ExecuteNonQuery: 可以通过该命令来执行不需要返回值的操作,如 UPDATE、INSERT 和 DELETE 等 SQL 命令。该命令不返回任何行,而只是返回执行该命令时所影响到表的行数。
- ExecuteScalar: 它可以执行 Select 查询,但返回的是一个单值,多用户查询聚合值的情况,如使用 count() 或者 sum() 函数的 SQL 命令。
- ExecuteReader: 该方法返回一个 DataReader 对象,内容为与命令匹配的所有行。

【实例 5-11】 连接本机 ScoreDB 数据库,统计 Student 表中学生的人数。

```
SqlCommand mycomm=sqlConn.CreateCommand();
mycomm.CommandText="select count(*) from student";
int sum=mycomm.ExecuteScalar();
```

5. 使用 SqlDataReader

ADO.NET 中的 DataReader 是从数据库中检索只进、只读的数据流。“只进”是指记

录的接收是按照顺序进行的,不能后退。“只读”是指读出的数据不能更新、修改、删除记录。查询结果存储在客户端的网络缓冲区中,直到用户使用 DataReader 的 Read 方法对它们发出请求。

【实例 5-12】 连接本机 ScoreDB 数据库,列出 Student 表中所有学生记录。

```
SqlCommand mycomm=sqlConn.CreateCommand();
mycomm.CommandText="select * from student";
SqlDataReader mydr=mycomm.ExecuteReader();
string str="";
While(mydr.Read())
{
    str+=mydr["ID"];
    str+=mydr["StudentName"];
    str+="\n";
}
lblInfo.Text=str;
```

5.3 完成学生登录功能

启动 Microsoft Visual Studio 2012,在解决方案资源管理器中,右击 ScoreMIS 项目,在弹出的快捷菜单中选择【添加】→【Windows 窗体】命令,打开【添加新项】对话框,输入窗体名称 Login,如图 5-12 所示。单击【确定】按钮。

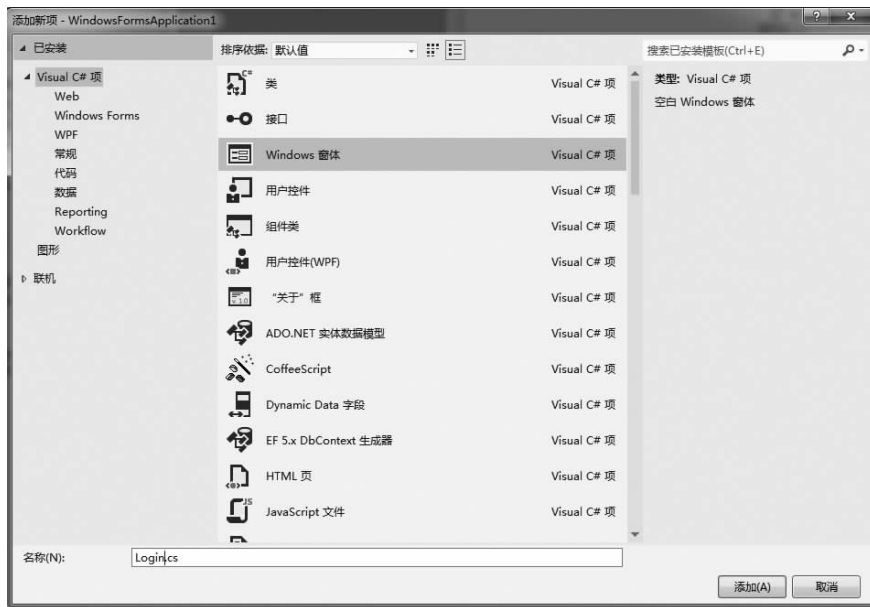


图 5-12 添加 Login 窗体