

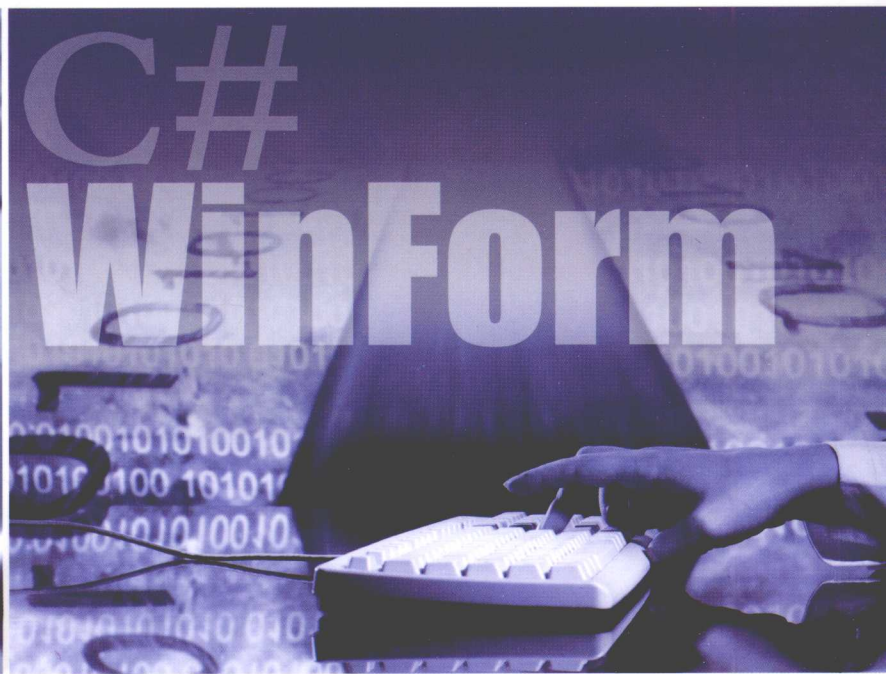
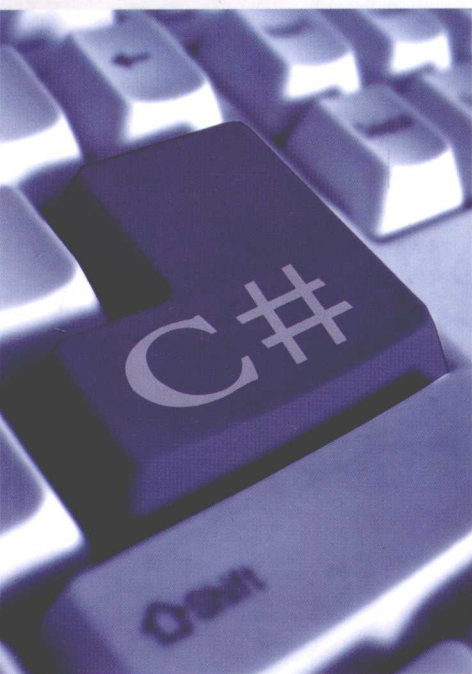


软件职业技术学院“十一五”规划教材

C# WinForm

上机实训指导

主 编 钱 哨 李继哲 李挥剑
副主编 潘静虹 孙 晔 魏丽莉 王曙光



中国水利水电出版社
www.waterpub.com.cn

软件职业技术学院“十一五”规划教材

C# WinForm 上机实训指导

主 编 钱 哨 李继哲 李挥剑

副主编 潘静虹 孙 晔 魏丽莉 王曙光

作 者	主 编 钱 哨 李继哲 李挥剑 副主编 潘静虹 孙 晔 魏丽莉 王曙光
出 版 行	中国水利水电出版社 (北京市西城区右安门内大街1号D座, 100038) 网址: www.waterpub.com.cn E-mail: michael@263.net (水) sales@waterpub.com.cn 电话: (010) 68369528 (营销中心) 65262819 (水)
经 销	全国各城市新华书店均有代售
排 版	北京方正电子技术有限公司
印 刷	北京航空印刷厂
规格	185mm×260mm 16开本 132印张 310千字
版 次	2010年1月第1版 2010年1月第1次印刷
印 数	0001—4000册
定 价	20.00元



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书是《C# WinForm 实践开发教程》的配套实训和上机辅导教材,所有的案例及实训素材都是源于计算机及应用软件教学一线教师多年的实训授课心得,希望读者通过这些实训案例及具体项目的研发,全面系统地掌握 C# WinForm 的知识和具体应用技能。

本书主要面向.NET Framework 2.0 的初、中级用户,可作为高等院校计算机相关专业学生学习 C# WinForm 的上机实训和参考书,特别适合作为《C# WinForm 实践开发教程》的补充教材。

图书在版编目 (C I P) 数据

C# WinForm 上机实训指导 / 钱哨, 李继哲, 李挥剑
主编. — 北京: 中国水利水电出版社, 2010. 1

软件职业技术学院“十一五”规划教材
ISBN 978-7-5084-6996-6

I. ①C… II. ①钱… ②李… ③李… III. ①C语言—
程序设计—高等学校: 技术学校—教学参考资料②窗口软
件, Windows—程序设计—高等学校: 技术学校—教学参
考资料 IV. ①TP312②TP316.7

中国版本图书馆CIP数据核字(2009)第212034号

策划编辑: 石永峰 责任编辑: 张玉玲 加工编辑: 徐 雯 封面设计: 李 佳

书 名	软件职业技术学院“十一五”规划教材 C# WinForm 上机实训指导
作 者	主 编 钱 哨 李继哲 李挥剑 副主编 潘静虹 孙 晔 魏丽莉 王曙光
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×260mm 16开本 12.5印张 310千字
版 次	2010年1月第1版 2010年1月第1次印刷
印 数	0001—4000册
定 价	20.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换
版权所有·侵权必究

序

随着信息技术的广泛应用和互联网的迅猛发展,以信息产业发展水平为主要特征的综合国力竞争日趋激烈,软件产业作为信息产业的核心和国民经济信息化的基础,越来越受到世界各国的高度重视。中国加入世贸组织后,必须以积极的姿态,在更大范围和更深程度上参与国际合作和竞争。在这种形势下,摆在我们面前的突出问题是人才短缺,计算机应用与软件技术专业领域技能型人才的缺乏尤为突出,无论是数量还是质量,都远不能适应国内软件产业的发展和信息化建设的需要。因此,深化教育教学改革,推动高等职业教育与培训的全面发展,大力提高教学质量,是迫在眉睫的重要任务。

2000年6月,国务院发布《鼓励软件产业和集成电路产业发展的若干政策》,明确提出鼓励资金、人才等资源投向软件产业,并要求教育部门根据市场需求进一步扩大软件人才培养规模,依托高等学校、科研院所,建立一批软件人才培养基地。2002年9月,国务院办公厅转发了国务院信息化工作办公室制定的《振兴软件产业行动纲要》,该《纲要》明确提出要改善软件人才结构,大规模培养软件初级编程人员,满足软件工业化生产的需要。教育部也于2001年12月在35所大学启动了示范性软件学院的建设工作,并于2003年11月启动了试办示范性软件职业技术学院的建设工作。

示范性软件职业技术学院的建设目标是:经过几年努力,建设一批能够培养大量具有竞争能力的实用型软件职业技术人才的基地,面向就业、产学结合,为我国专科层次软件职业技术人才培养起到示范作用,并以此推动高等职业技术教育人才培养体系与管理体制和运行机制的改革。要达到这个目标,建立一套适合软件职业技术学院人才培养模式的教材体系显得尤为重要。

高职高专的教材建设已经走过了几个发展阶段,由最开始本科教材的压缩到加大实践教学环节的比重,再到强调实践性教学环节,但是学生在学习时还是反映存在理论与实践的结合问题。为此,中国水利水电出版社在经过深入调查研究后,组织了一批长期工作在高职高专教学一线的老师,编写了这套“软件职业技术学院‘十一五’规划教材”,本套教材采用项目驱动的方法来编写,即全书所有章节都以实例作引导来说明各知识点,各章实例之间并不是孤立的,每个实例都可以作为最终项目的一个组成部分;每一章章末还配有实习实训(或叫实验),这些实训组合起来是一个完整的项目。

采用这种方式编写的图书与市场上同类教材相比更具优越性,学生不仅仅学到了知识点,还通过项目将这些知识点连成一条线,开拓了思路,掌握了知识,达到了面向岗位的职业教育培训目标。

本套教材的主要特点有:

- (1) 课程主辅分明——重点突出,教学内容实用。
- (2) 内容衔接合理——完全按项目运作所需的知识体系结构设置。

(3) 突出实习实训——重在培养学生的专业能力和实践能力，力求缩短人才与企业间的磨合期。

(4) 教材配套齐全——本套教材不仅包括教学用书，还包括实习实训材料、教学课件等，使用方便。

本套教材适用于广大计算机专业和非计算机专业的大中专院校的学生学习，也可作为有志于学习计算机软件技术与开发的工程技术人员的参考教材。

编委会

2006年7月

前 言

本书面向的读者

本书是《C# WinForm 实践开发教程》的配套实训和上机辅导教材,所有的案例及实训素材都是源于计算机及应用软件教学一线教师多年的实训授课心得,希望读者通过这些实训案例及具体项目的研发,全面系统地掌握 C# WinForm 的知识和具体应用技能。

本书主要面向 .NET Framework 2.0 的初、中级用户,可作为高等院校计算机相关专业学生学习 C# WinForm 的上机实训用书。

本书的组织结构

本教材为了配合《C# WinForm 实践开发教程》课程的教学工作而编写,分为 3 个部分:第一部分是第 1~6 章,突出了《C# WinForm 实践开发教程》各章的知识结构和学习要点,编写了一系列的上机案例,以辅助学生进行课堂与课后的上机实验。

第二部分是第 7、8 章,分别设计了两个完整的项目案例,为开展实训工作奠定基础。

第三部分是第 9 章,重点介绍了教师如何开展本课程的实训工作,以及实训工作中的流程和实训管理细节,特别突出了对实训工作的评价体系的设置。

本书由钱哨、李继哲、李挥剑老师任主编,潘静虹、孙晔、魏丽莉、王曙光任副主编。参加本书编写的还有夏永恒、鲁一力、何文、张传立、张继红、陈小全、黄少波、王满师、张光升、李凤等。限于编著者水平,书中有错误或不妥之处,请读者给予批评指正。

编者

2009 年 10 月

目 录

序

前言

第1章 Windows 编程基础上机实训	1	7.1.1 基本功能规定	113
1.1 WinForm 中的常用控件应用上机实验	1	7.1.2 数据库规划设计	116
1.2 SDI 和 MDI 应用上机实验	5	7.2 即时聊天工具详细开发步骤	119
第2章 WinForm 高级控件上机实训	11	7.2.1 建立基本类库文件	119
2.1 选项卡控件上机实验	11	7.2.2 实现登录功能	124
2.2 StatusStrip 控件上机实验	15	7.2.3 用户注册窗体设计	126
2.3 Timer 控件上机实验	19	7.2.4 实现即时聊天主窗体设计	131
2.4 ListView 控件上机实验	25	7.2.5 在线聊天窗体设计	144
2.5 TreeView 控件上机实验	27	7.2.6 个人信息编辑功能设计	149
2.6 流媒体播放器改进实验	30	7.2.7 个人更换头像功能设计	157
第3章 文件处理技术上机实训	33	7.2.8 查找好友功能的设计	164
3.1 文件流 FileStream 上机实验	33	第8章 网络编程上机实训	176
3.2 二进制流和内存流文件操作上机实验	48	8.1 网络编程综合实验	176
第4章 ADO.NET 数据库访问技术上机实训	55	8.1.1 游戏功能要求	176
4.1 学生基本信息管理实验	55	8.1.2 五子棋算法简介	176
4.2 基于 C# 的异构数据源数据整合技术	64	8.1.3 服务端编程	178
4.3 添加、修改、删除、查询数据库基本 表信息	67	8.1.4 客户端编程	181
4.4 基于多窗体的 DataGridView 控件复杂 业务处理	75	第9章 Winform 课程实训步骤	183
第5章 GDI+编程上机实训	86	9.1 Winform 项目实训具体步骤和实现方法	183
5.1 动态绘制统计图形概述	86	9.1.1 筹建项目小组的基本原则	183
5.2 动态绘制统计图形上机实验	88	9.1.2 项目小组任务分工阶段	184
5.2.1 绘制柱状统计图上机实验	88	9.1.3 需求分析阶段（软件及数据库 建模）	187
5.2.2 绘制饼形统计图上机实验	92	9.1.4 软件开发的实施阶段	187
第6章 水晶报表技术上机实训	96	9.1.5 软件测试阶段	189
第7章 即时聊天工具综合开发实训	113	9.1.6 安装部署、答辩阶段	189
7.1 综合项目陈述	113	参考文献	192

第1章

Windows 编程基础上机实训



本章内容

- WinForm 中的常用控件上机实验
 - WinForm 登录窗体实验
 - listBox 控件实验
 - 基本控件综合实验
- SDI 和 MDI 应用上机实验
 - 创建一个 MDI 应用实验
 - 创建一个 MDI 文本编辑器实验
 - 主窗体与子窗体传值实验

1.1 WinForm 中的常用控件应用上机实验

本节上机实验的内容包括：

- WinForm 登录窗体实验
- listBox 控件实验
- 基本控件综合实验



案例学习：WinForm 登录窗体实验

本次实验的目标是快速建立如图 1-1 所示的用户登录窗体，通过该案例使读者快速掌握 WinForm 中的常用控件——标签控件和文本框（密码）框控件。重要属性设置如表 1-1 所示。

图 1-1 用户登录窗体目标界面

表 1-1 登录窗体属性设置

窗体对象名称	属性参数设置	说明
Form1	MaximizeBox = false;	窗体不允许最大化
textBox2	textBox2.PasswordChar = '*'	设置 textBox2 为密码框

(1) 如图 1-1 所示, 从工具箱中拖曳具体的控件到 Form 窗体上, 并更改标签对象和按钮的 text 属性为图 1-1 所示内容。将窗体及文本框的属性按照表 1-1 设置。双击界面, 进入后台代码编辑区, 首先进行初始化代码的编写工作, 如下所示:

```
namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private bool ISLOGIN = false;
        //加在其构造函数的前面稍后使用
        //然后加一个属性, 该属性用来判别是否登录
        public bool isLogin
        {
            get
            {
                return this.ISLOGIN;
            }
        }
        private void Form1_Load(object sender, EventArgs e)
        {
        }
    }
}
```

(2) 在“确定”按钮的 Click 事件处理方法中加入你自己的验证代码, 可以是读数据库, 也可以是其他任何的方法。这里举一个最简单的例子, 代码如下所示:

```
private void button1_Click(object sender, EventArgs e)
{
    if (this.textBox1.Text == "qq" && this.textBox2.Text == "qq")
    {
        this.ISLOGIN = true;
        this.Close(); //登录成功才关闭登录窗体
    }
    else
    {
        MessageBox.Show("非法用户名或密码, 请再重试一次!");
    }
}
```

(3) 建立一个窗体，命名为 Form2。将实验步骤 2 的代码修改如下：

```
private void button1_Click(object sender, EventArgs e)
{
    if (this.textBox1.Text == "qq" && this.textBox2.Text == "qq")
    {
        this.ISLOGIN = true;
        Form2 f2 = new Form2();
        f2.ShowDialog();
        this.Close(); // 登录成功才关闭登录窗体
    }
    else
    {
        MessageBox.Show("非法用户名或密码，请再重试一次！");
    }
}
```



案例学习：listBox 控件实验

本次实验的目标是熟悉并掌握 ListBox 控件。基本要求为：

(1) 单击“显示信息”按钮的时候，下拉列表显示软件部、硬件部、财务部和人事部，并在下面通过 Label 控件显示行数的统计信息。

(2) 单击“插入信息”按钮的时候，可以在选中的某行后插入新的内容为“插入值”信息，并在下面通过 Label 控件重新显示行数的统计信息，如图 1-2 所示。

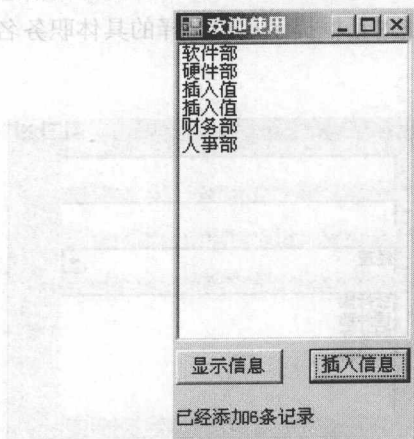


图 1-2 使用 ListBox 控件实验目标界面



参考主要代码

```
// 显示信息的主要代码
private void button1_Click(object sender, EventArgs e)
{

```

```
listBox1.Items.Clear();  
listBox1.Items.Add("软件部");  
listBox1.Items.Add("硬件部");  
listBox1.Items.Add("财务部");  
listBox1.Items.Add("人事部");  
}  
//插入信息主要代码  
private void button2_Click(object sender, EventArgs e)  
{  
    listBox1.Items.Insert(2, "插入值");  
    label1.Text = "已经添加" + listBox1.Items.Count.ToString() + "条记录";  
}
```



案例学习：基本控件综合实验

本次实验的目标是使读者熟悉并掌握 ListBox 控件、ComboBox 控件、Button 控件等，并开始熟练掌握多种控件的联合使用。基本要求为：

(1) 在窗体初始化事件中对 ComboBox 控件和 DropList 控件插入相关信息，并默认 ComboBox 控件中的“经理”为默认值。初始化时窗体所有的控件都处于不可用的状态。设置 DropList 控件的属性为允许多行选择类型。

(2) 单击“添加”按钮时，窗体所有的控件都处于可用的状态，并设置职员姓名的文本框自动获得焦点。

(3) 单击“添加”按钮时，清空“职员姓名”和“地址”文本框的内容，并设置“职员姓名”文本框自动获得焦点。

(4) 单击“退出”按钮时，首先弹出对话框显示用户选择的部门信息，而后退出系统。

(5) 当用户选择某个职务后，弹出对话框提示用户选择的具体职务名称。

本次实验的目标界面如图 1-3 所示。

图 1-3 使用基本控件综合实验的目标界面



参考主要代码

```
//单击“退出”按钮的鼠标单击事件  
private void button3_Click (object sender, System.EventArgs e)
```

```
{
    string str="";
    for(int ctr=0;ctr<=this.lstCurrDeptName.SelectedItems.Count-1;ctr++)
        str += "\n"+this.lstCurrDeptName.SelectedItems[ctr].ToString();
    MessageBox.Show("选定的项目为\n" +str);
    Application.Exit();
}

//改变职务时的选择切换事件
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    MessageBox.Show("您已经选定了" +this.comboBox1.SelectedItem.ToString());
}
```

1.2 SDI 和 MDI 应用上机实验

本节上机实验的内容包括：

- 创建一个 MDI 应用
- 创建一个 MDI 文本编辑器
- 主窗体与子窗体传值实验



案例学习：创建一个 MDI 应用实验

创建 MDI 应用程序，首先要像创建其他应用程序那样，在 Visual Studio 中创建一个 Windows 窗体应用程序。运行后的界面如图 1-4 所示，重要属性如表 1-2 所示。

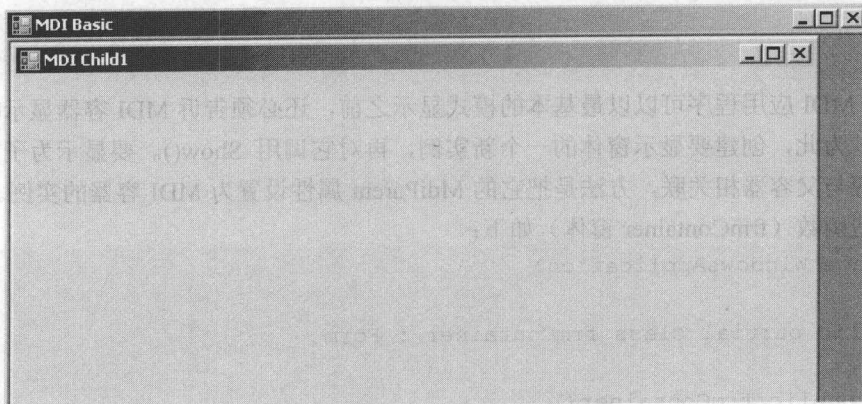


图 1-4 MDI 运行目标界面

(1) 要把应用程序的主窗口从一个窗体改为 MDI 容器，只需把窗体的 `IsMdiContainer` 属性设置为 `True` 即可。改变窗体的背景色，使之表示该窗体现在只有一种背景色，不应放置任何可见的控件（也可以放置控件，在某些情况下这也是合理的）。选择窗体，设置如表 1-2 所示的属性。

表 1-2 登录窗体属性设置 1

属性	值
Name	frmContainer
IsMdiContainer	True
Text	MDI Basic
WindowState	Maximized

(2) 再建立一个窗体，命名为 frmChild，选择窗体，设置如表 1-3 所示的属性。

表 1-3 登录窗体属性设置 2

属性	值
Name	frmChild
IsMdiContainer	False
Text	MDI Child1

在 frmChild 窗体的初始化事件中增加如下代码，特别注意的是定义该窗体时，指定了第一个窗体 frmContainer 为输入对象，命名为 parent:

```
namespace WindowsApplication1
{
    public partial class frmChild : Form
    {
        public frmChild(frmContainer parent)
        {
            InitializeComponent();
            this.MdiParent = parent;
        }
    }
}
```

(3) 在 MDI 应用程序可以以最基本的模式显示之前，还必须告诉 MDI 容器显示哪个窗口，再显示它们。为此，创建要显示窗体的一个新实例，再对它调用 Show()。要显示为子窗口的窗体的构造函数应与父容器相关联，方法是把它的 MdiParent 属性设置为 MDI 容器的实例。修改 MDI 父窗口的构造函数 (frmContainer 窗体) 如下:

```
namespace WindowsApplication1
{
    public partial class frmContainer : Form
    {
        public frmContainer()
        {
            InitializeComponent();
            //创建并实例化一个新的子窗体 frmChild
            frmChild child = new frmChild(this);
            //并将这个子窗体显示出来
            child.Show();
        }
    }
}
```

```
}  
}  
}
```

 案例学习：创建一个 MDI 文本编辑器实验

返回 1.2.1 节中的状态栏示例，把窗体重新命名为 frmEditor，将其 Text 属性改为 Editor。给项目添加一个新窗体 frmContainer.cs，在窗体上设置下述属性，如表 1-4 所示。

表 1-4 frmContainer 窗体属性

属性	值
Name	frmContainer
IsMdiContainer	True
Text	简单文本编辑器
WindowState	Maximized

(1) 在当前的 frmContainer 窗体中添加一个 MenuStrip 菜单控件，如图 1-5 所示。其中文件的“新建”属性配置如图 1-6 所示。

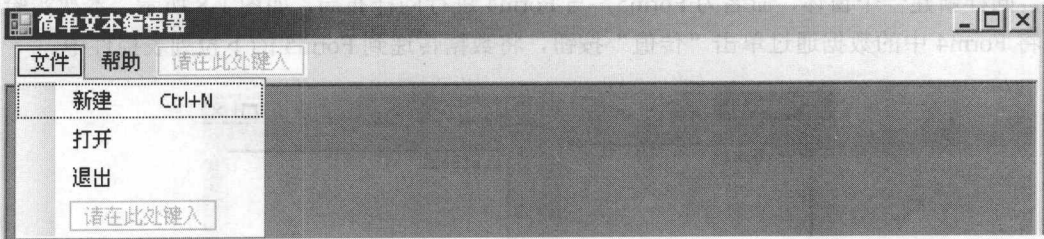


图 1-5 主界面菜单设置

ShortcutKeys	Ctrl+N
ShowShortcutKeys	True
Size	152, 22
Tag	
Text	新建

图 1-6 “新建”属性配置

(2) 双击“新建”菜单进入后台，编码如下所示：

```
private void toolStripMenuItem3_Click(object sender, EventArgs e)  
{  
    frmEditor newForm = new frmEditor(this);  
    newForm.Show();  
}
```

(3) 设置 frmEditor 窗体界面，分别从工具箱中拖曳菜单和工具条控件，构造出如图 1-7 所示的界面。

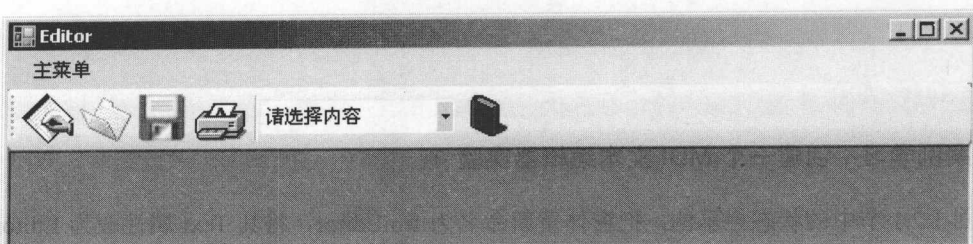


图 1-7 frmEditor 窗体界面



案例学习：主窗体与子窗体的传值实验

主窗体与子窗体传值的方法有很多，此次我们使用 Form 类的 Owner 属性进行传值。若要使某窗体归另一个窗体所有，需要对其 Owner 属性分配一个对将成为所有者的窗体的引用。当一个窗体归另一窗体所有时，它便随着所有者窗体最小化和关闭。例如，如果 Form2 归窗体 Form1 所有，则关闭或最小化 Form1 时，也会关闭或最小化 Form2。并且附属窗体从不显示在其所有者窗体后面。可以将附属窗体用于查找和替换窗口之类的窗口，当选定所有者窗体时，这些窗口不应消失。若要确定某父窗体拥有的窗体，请使用 OwnedForms 属性。

(1) 新建一个窗体，命名为 Form4，从工具箱拖曳一个 Button 按钮和一个 ListBox 下拉列表控件。同样新建一个窗体，命名为 Form5，与 Form4 进行同样布局，如图 1-8 所示。本次实验的目的是将 Form4 中的数据通过单击“传值”按钮，将数据传递到 Form5 的下拉列表控件中。

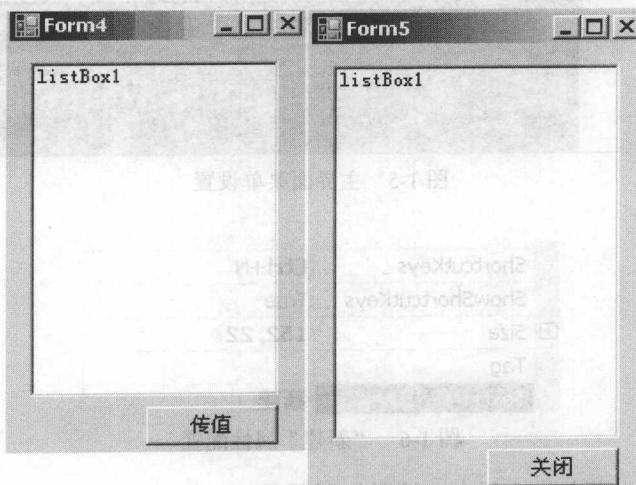


图 1-8 Form4 和 Form5 的窗体配置

(2) 双击 Form4 窗体，进入后台编码区域，键入如下代码，具体说明请参见代码帮助文字内容：

```
.....
using System.Collections;
//注意：必须调用该命名空间，否则无法建立 ArrayList 对象
namespace WindowsApplication1
{
```

```

public partial class Form4 : Form
{
    //Form4 窗体的私有变量 listData1 用来保存 listBox1 下拉列表中的数据
    private ArrayList listData1;
    //Form4 窗体的公共属性。该属性可以被外部所访问，访问的内容就是 Form4 窗体的私有变
    //量 listData1 的内容
    //这里我们采用属性，感觉语法更灵活清楚
    public ArrayList ListData2
    {
        get { return this.listData1; }
    }

    //配置 Form4 构造函数，目的是填充 listBox1 中的具体数据内容
    public Form4()
    {
        InitializeComponent();
        this.listData1 = new ArrayList();
        this.listData1.Add("DotNet");
        this.listData1.Add("C#");
        this.listData1.Add("Asp.net");
        this.listData1.Add("WebService");
        this.listData1.Add("XML");
        this.listBox1.DataSource = this.listData1;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Form5 formChild = new Form5(); //实例化另外一个窗体 Form5
        formChild.Owner = this;
        //我们设置了 formChild.Owner 为 this，这样，子窗体和主窗体就有联系了
        formChild.ShowDialog(); //打开子窗体
        //当然上面两句也可以改为：formChild.ShowDialog(this);
        this.listBox1.DataSource = null;
        this.listBox1.DataSource = this.listData1;
        this.Hide();
        this.Close();
    }
}

```

(3) 双击 Form5 窗体，进入后台编码区域，键入如下代码，具体说明请参见代码帮助文字内容：

```

.....
using System.Collections;
namespace WindowsApplication1
{
    public partial class Form5 : Form

```

```

{
    private ArrayList listData3;
    //建立私有变量, 该变量用于接收复窗体的共有属性数据
    public Form5()
    {
        InitializeComponent();
        //千万别勿 Form5_Load 中的代码放到构造函数里面, 因为在主窗体修改按钮被单击后
        //开始执行 Form5 formChild = new Form5();而在 Form2 的实例化过程中会在构造函数中执行
        //Form4 pareForm = (Form4)this.Owner;而这时的 this.Owner 是没有值的, 为空引用, 子
        //窗体无法接收父窗体传递过来的数据
    }
    private void Form5_Load(object sender, EventArgs e)
    {
        Form4 pareForm = (Form4)this.Owner; //实例化父窗体, 建立父子窗体之间联系
        this.listData3 = pareForm.ListData2; //访问父窗体的公有属性给当前子窗体
        //的私有变量
        foreach (object o in this.listData3) //遍历后将数据显示在 listBox1 中
        {
            this.listBox1.Items.Add(o);
        }
    }
    private void button1_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
}
}

```