

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C#程序设计上机指导 与实例解析 (第2版)

C# Programming Guidance and Examples of
Experiments (2nd Edition)

马骏 主编

贾培艳 黄亚博 副主编

- 上机内容重在基础：基本程序设计 + 数据库技术 + 网络编程
- 实验类型包括全面：基础型实验 + 综合型实验 + 设计型实验
- 开发实例突出实用：数据库应用开发实例 + 智能客户端开发实例



精品系列



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C#程序设计上机指导 与实例解析 (第2版)

C# Programming Guidance and Examples of
Experiments (2nd Edition)

马骏 主编

贾培艳 黄亚博 副主编



精品系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

C#程序设计上机指导与实例解析 / 马骏主编. -- 2
版. -- 北京 : 人民邮电出版社, 2011. 2
21世纪高等学校计算机规划教材
ISBN 978-7-115-24178-8

I. ①C… II. ①马… III. ①
C语言—程序设计—高等学校—教学参考资料 IV.
①TP312

中国版本图书馆CIP数据核字(2010)第231470号

内 容 提 要

本书是学生学习 C#的配套教材, 包括 C#程序设计及应用上机指导和 C#网络应用编程上机指导两篇, 每篇上机指导除了基础实验指导外, 还包括综合型或设计型实验指导以及对应的项目开发实例解析。上机实验指导涵盖了 C#的大部分基本技术, 融入了各种常见的设计技巧, 囊括了初学者必须掌握的基本知识; 项目开发实例解析则分别用非常实用的例子详细介绍了相关的系统开发技术和设计技巧。

本书最适用于采用《C#程序设计及应用教程》(第2版)和《C#网络应用编程》(第2版)的读者作为上机练习的参考指导书。对于使用其他 C#教材的读者以及自学 C#的读者, 该书也是一本非常好的辅助教材。

21世纪高等学校计算机规划教材

C#程序设计上机指导与实例解析 (第2版)

-
- ◆ 主 编 马 骏
 - 副 主 编 贾培艳 黄亚博
 - 责任编辑 邹文波
 - 执行编辑 崔细根
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 10.75 2011年2月第2版
字数: 276千字 2011年2月北京第1次印刷

ISBN 978-7-115-24178-8

定价: 22.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

第 2 版前言

要学好并灵活掌握一门程序设计语言,真正达到学以致用目的,除了掌握语言本身的语法知识和基本的编程技巧外,另一种比较快速、有效的学习途径,就是进行大量的上机练习,并尝试开发一些不太复杂的应用程序,特别是对于从未参加过项目开发训练的初学者,这种训练显得非常重要。

本书第 1 版出版后,得到了读者的普遍认可和好评。为了与第 2 版《C#程序设计及应用》(人民邮电出版社,马骏主编)和第 2 版《C#网络应用编程》(人民邮电出版社,马骏主编)的两本教材配套,我们在第 1 版实验指导教材的基础上,编写了针对 VS2008 开发环境的上机指导配套教材。

第 2 版除了与两本教材配套的上机实验指导外,作者还精心编写了与实际应用非常接近的开发实例解析,设计中使用的技术涵盖了教材的很多知识点,是巩固所学内容,将一个个较小的基本编程训练灵活运用于实际项目开发的捷径。

本书由马骏任主编,贾培艳、黄亚博任副主编。参与本书编写和代码调试等工作的还有侯彦娥、秦勉、毋琳、王玉璟、周珂、杨雪、杜军永等。

由于编者水平有限,书中难免存在错误之处,敬请读者批评指正。

编 者

2010 年 11 月

目 录

第 1 篇 C#程序设计及应用上机指导

第 1 章 程序设计基础实验.....2	1.6.1 实验目的.....40
1.1 实验一 VS2008 开发环境练习.....2	1.6.2 实验内容.....40
1.1.1 实验目的.....2	1.6.3 实验要求.....41
1.1.2 实验内容.....3	1.6.4 实验提示.....41
1.1.3 实验要求.....3	1.6.5 实验报告中要求回答的问题.....43
1.1.4 实验提示.....3	第 2 章 程序设计综合型实验.....44
1.1.5 实验报告中要求回答的问题.....7	2.1 综合型实验完成方式及考核方式.....44
1.2 实验二 C#基本编程方法练习.....7	2.1.1 综合型实验完成方式.....44
1.2.1 实验目的.....7	2.1.2 综合型实验考核方式.....45
1.2.2 实验内容.....7	2.2 银行业务管理系统功能要求.....45
1.2.3 实验要求.....8	2.2.1 数据库结构基本要求.....45
1.2.4 实验提示.....8	2.2.2 系统总体功能要求.....47
1.2.5 实验报告中要求回答的问题.....12	2.2.3 存款取款模块基本要求.....53
1.3 实验三 面向对象程序设计练习.....13	2.2.4 汇总查询模块基本要求.....58
1.3.1 实验目的.....13	2.2.5 职员管理模块基本要求.....59
1.3.2 实验内容.....13	2.2.6 其他功能模块基本要求.....59
1.3.3 实验要求.....14	2.2.7 利率设置模块基本要求.....60
1.3.4 实验提示.....14	2.2.8 银行业务管理系统 功能扩展要求.....60
1.3.5 实验报告中要求回答的问题.....22	第 3 章 数据库应用开发实例解析.....61
1.4 实验四 界面设计与文件存取.....22	3.1 系统概述.....61
1.4.1 实验目的.....22	3.1.1 业务流程及基本需求.....61
1.4.2 实验内容.....23	3.1.2 系统总体功能设计.....62
1.4.3 实验要求.....23	3.2 数据库设计.....62
1.4.4 实验提示.....24	3.2.1 数据表划分及逻辑结构.....62
1.4.5 实验报告中要求回答的问题.....31	3.2.2 创建数据库.....64
1.5 实验五 LINQ 与数据库操作练习.....31	3.2.3 存储过程设计.....65
1.5.1 实验目的.....32	3.2.4 LINQ 相关操作.....65
1.5.2 实验内容.....32	3.3 基础模块类设计.....66
1.5.3 实验要求.....32	3.3.1 设计模块公用类.....66
1.5.4 实验提示.....33	3.3.2 设计筛选条件用户控件.....66
1.5.5 实验报告中要求回答的问题.....40	
1.6 实验六 Web 设计基础练习.....40	

3.4 登录模块与主窗体设计	68	3.8.2 查询成绩	79
3.4.1 登录密码加密解密处理	68	3.8.3 统计功能	80
3.4.2 登录窗体界面设计	69	3.9 报表输出模块	81
3.4.3 主功能窗体界面设计	70	3.9.1 基本情况表	81
3.5 编码管理模块	72	3.9.2 成绩表	82
3.5.1 设计编码管理通用类	72	3.9.3 点名册	84
3.5.2 修改主窗体菜单实现各种 编码管理功能	72	3.9.4 学籍表	85
3.6 基本情况管理模块	73	3.10 数据备份与恢复模块	88
3.6.1 录入基本情况	73	3.10.1 备份 SQL Server 数据库	88
3.6.2 成批数据修改	75	3.10.2 恢复数据	89
3.7 课程成绩管理模块	76	3.11 其他功能模块	92
3.8 查询统计模块	78	3.11.1 修改登录密码	92
3.8.1 查询基本情况	78	3.11.2 注销某年级数据	93
		3.12 制作系统安装程序	93

第2篇 C#网络应用编程上机指导

第4章 网络编程基础实验

4.1 实验一 多线程程序设计	98
4.1.1 实验目的	98
4.1.2 实验内容	98
4.1.3 实验要求	99
4.1.4 实验提示	99
4.1.5 实验报告中要求回答的问题	101
4.2 实验二 即时通信	102
4.2.1 实验目的	102
4.2.2 实验内容	102
4.2.3 实验要求	103
4.2.4 实验提示	103
4.2.5 实验报告中要求回答的问题	109
4.3 实验三 网络呼叫应答提醒	109
4.3.1 实验目的	109
4.3.2 实验内容	110
4.3.3 实验要求	110
4.3.4 实验提示	110
4.3.5 实验报告中要求回答的问题	113
4.4 实验四 FTP 文件操作	113
4.4.1 实验目的	113
4.4.2 实验内容	113

4.4.3 实验要求	114
4.4.4 实验提示	114
4.4.5 实验报告中要求回答的问题	116
4.5 实验五 数据加密与解密	116
4.5.1 实验目的	116
4.5.2 实验内容	116
4.5.3 实验要求	116
4.5.4 实验提示	117
4.5.5 实验报告中要求回答的问题	118

第5章 网络编程设计型实验

5.1 设计型实验完成方式及考核方式	119
5.1.1 设计型实验基本功能要求	120
5.1.2 设计型实验完成方式	120
5.1.3 设计型实验考核方式	120
5.2 设计型实验选题参考	121
5.2.1 城镇规划	121
5.2.2 联机画板	122
5.2.3 火炬设计	123
5.2.4 开心农场	123
5.2.5 房间设计	124
5.2.6 电路图制作	125
5.2.7 几何绘图	126

第 6 章 网络应用编程实例解析.....128	6.2.2 服务器与客户端通信描述..... 129
6.1 智能客户端基本知识.....128	6.3 界面设计与代码实现..... 133
6.1.1 通过网站提供客户端软件下载128	6.3.1 服务器端设计..... 133
6.1.2 客户端软件版本测试与自动升级.....129	6.3.2 客户端设计..... 153
6.2 网络对战五子棋游戏功能描述.....129	6.4 智能客户端的发布与升级..... 158
6.2.1 网络对战五子棋游戏规则描述129	6.4.1 发布与传统安装的区别..... 158
	6.4.2 发布智能客户端程序..... 159

第 1 篇

C#程序设计及应用 上机指导

本篇的上机实验指导和开发实例是针对选用第 2 版《C#程序设计及应用教程》的读者而编写的，但也可用于选用其他 C#教材相关内容的上机实验指导。

本部分上机环境要求如下：

操作系统：Windows XP SP3 或者 Window 7。

内存要求：至少 512MB。

开发环境安装要求：

（1）安装 VS2008 简体中文专业版。

（2）安装 VS2008 SP1。

数据库安装要求：

（1）安装 SQL Server 2008 Express。

（2）安装 SQL Server 2008 Express SP1。

第 1 章

程序设计基础实验

程序设计实验指导共包含 7 个上机实验，其中前 6 个实验为基础型实验，第 7 个实验为综合型实验。本章主要介绍基础型上机实验内容。

上机实验环境

操作系统：Windows XP SP3 或者 Windows 7。

内存要求：至少 512MB。

开发环境要求：安装了 VS 2008 SP1 的 VS 2008 Professional 简体中文版。

实验报告要求

(1) 使用统一的实验报告纸，要求字迹工整，内容清晰。注意填写必要的信息：姓名、学号、班级和辅导教师。

(2) 必须认真填写实验题目、实验目的等；实验步骤中要求列出当次实验中自己认为有意义的操作过程及各种必要的输入输出情况；写出主要的功能模块划分、设计界面和关键源代码，以及上机调试过程中遇到的问题和解决办法。

1.1 实验一 VS2008 开发环境练习

本实验主要是为了让读者熟悉 Visual Studio 2008 开发环境，掌握如何在此开发环境下使用最基本的 Windows 窗体控件，以及对窗体中的多个控件快速进行排列布局的方法，为后续的实验打下基础。建议用 2 小时完成本实验。

1.1.1 实验目的

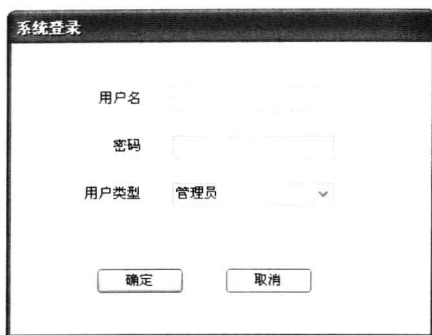
通过本实验提供的实验提示，理解设计 Windows 窗体界面的基本操作技术和技巧，掌握最常用的基本控件的用法。

读者通过本实验，应达到如下目标。

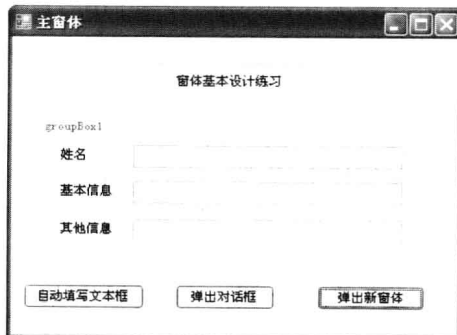
- (1) 掌握创建 Windows 窗体应用程序的基本操作步骤。
- (2) 掌握解决方案资源管理器、工具箱、属性和事件的基本用法。
- (3) 掌握修改窗体标题、窗体大小、窗体样式的方法。
- (4) 掌握文本框 (TextBox)、标签 (Label) 和下拉框 (ComboBox) 的基本用法。
- (5) 掌握对窗体中的多个控件进行快速布局的基本方法。
- (6) 掌握观察项目保存的目录和文件结构，以及生成的可执行文件的位置的方法。

1.1.2 实验内容

设计一个如图 1-1 (a) 所示的系统登录窗体界面, 当用户单击【确定】按钮时, 登录窗体自动消失, 然后显示如图 1-1 (b) 所示的主窗体界面。



(a) 登录窗体



(b) 主窗体

图 1-1 实验一的运行效果

如果用户在登录窗体中单击【取消】按钮, 则直接退出程序。

在主窗体界面中, 用户可以单击不同的按钮, 实现对应的功能。

1.1.3 实验要求

由于读者在练习本实验时, 还没有学习本书后续章节的内容, 所以本实验的重点是让读者关注如何设计和快速排列窗体界面。至于如何用 C# 语言实现要求的各种功能, 实验步骤中会给出相应的代码, 读者只需要按照实验提示操作, 得到正确的运行结果即可。

1.1.4 实验提示

(1) 运行 Microsoft Visual Studio 2008, 选择【新建项目】, 在弹出的新建项目类型模板中, 选择“Windows 窗体应用程序”模板, 在【名称】文本框中将项目名称更改为 WindowsFormsExec, 在【位置】文本框中输入保存项目的目录位置, 例如 C:\CSharpExperiment, 确认选中了【创建解决方案的目录】复选框, 如图 1-2 所示, 然后单击【确定】按钮。

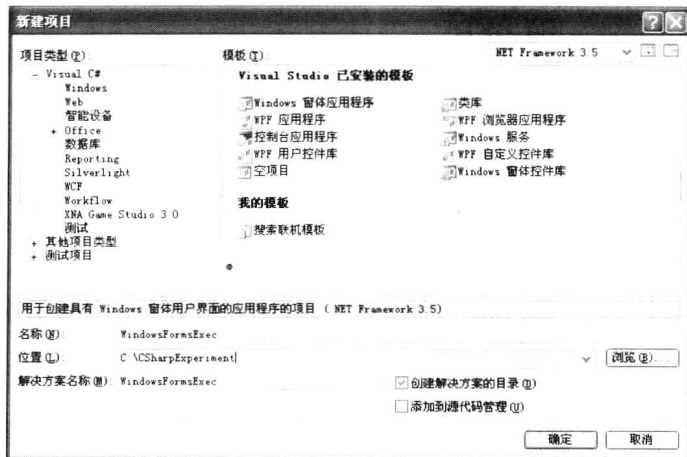


图 1-2 新建 Windows 窗体应用程序项目

注意不要使用#之类的符号作为项目名称，应该用英文单词或者拼音的组合给项目起名，并且使用有意义的名称。另外还要注意不要将【位置】设置为网络映射驱动器，要确保保存位置为本地的硬盘，这是因为设置为网络映射驱动器后，可能会因为没有相应的读写权限，导致无法正常运行和调试程序。

(2) 在【解决方案资源管理器】中，单击 Form1.cs，重命名 Form1.cs 为 LoginForm.cs，然后分别修改窗体的【Text】属性、【Size】属性和【FormBorderStyle】属性，观察修改后窗体显示的效果。

(3) 在【解决方案资源管理器】中，鼠标右击项目名“WindowsFormsExec”，选择【添加】→【Windows 窗体】，添加一个名为 MainForm.cs 的窗体。

(4) 打开 Program.cs，将代码改为下面的内容。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
namespace WindowsFormsExec
{
    static class Program
    {
        public static bool isValidUser;
        /// <summary>
        /// 应用程序的主入口点。
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new LoginForm());
            if (isValidUser == true)
            {
                Application.Run(new MainForm());
            }
        }
    }
}
```

(5) 打开 LoginForm.cs 的设计界面，按照下面的步骤，设计系统登录功能。

1) 将窗体的【ControlBox】属性改为“False”，目的是为了不让该窗体显示最大化、最小化和关闭按钮。

2) 从【工具箱】中向设计窗体拖放 1 个 Label 控件，选中该 Label 对象，按住<Ctrl>键，用鼠标左键将该对象按纵向排列复制 2 个，然后分别修改 3 个 Label 控件的【Text】属性为“用户名”、“密码”和“用户类型”。

3) 选择【工具箱】，双击 TextBox 控件 2 次，观察窗体中添加的控件情况，再分别拖动窗体中的 2 个 TextBox 对象，放在对应的 Label 对象右边，并修改其【Name】属性为“textBoxUserName”和“textBoxPassword”。

4) 从【工具箱】中向设计窗体拖放一个 ComboBox 控件，放在对应的 Label 对象的右边，然后在【属性】窗口中将其【Name】属性修改为“comboBoxLoginType”。

5) 按住<Shift>键,用鼠标同时选中2个 TextBox 控件和1个 ComboBox 控件,利用快捷工具栏的“使大小相同”按钮,将3个控件的大小变为相同,然后将其拖动至合适大小。

6) 从【工具箱】中向窗体拖放两个 Button 控件,分别修改其【Name】属性为“buttonOK”和“buttonCancel”,【Text】属性为“确定”和“取消”。

7) 鼠标双击【确定】按钮,以便让其自动添加 Click 事件,然后将该事件代码改为下面的内容。

```
private void buttonOK_Click(object sender, EventArgs e)
{
    Program.IsValidUser = true;
    this.Close();
}
```

注意由于读者还没有学习数据库操作,这里并没有对用户身份及权限进行实际验证,当用户在运行时单击【确定】按钮后,这段代码的功能是直接设置验证成功标志,然后关闭此窗体。

8) 鼠标双击【取消】按钮,以便让其自动添加 Click 事件,然后将该事件代码改为下面的内容。

```
private void buttonCancel_Click(object sender, EventArgs e)
{
    Program.IsValidUser = false;
    this.Close();
}
```

这段代码的功能是设置验证失败标志,然后关闭此窗体。

9) 将构造函数改为下面的内容。

```
public LoginForm()
{
    InitializeComponent();
    comboBoxLoginType.Items.AddRange(new object[] { "管理员", "一般用户" });
    comboBoxLoginType.SelectedIndex = 0;
}
```

这段代码的功能是自动添加下拉框中提供的选项。

(6) 按<F5>键运行程序,观察系统登录窗体的运行效果。

(7) 在【解决方案资源管理器】中,鼠标右击项目名“WindowsFormsExec”,选择【添加】→【Windows 窗体】,添加一个名为 OtherForm.cs 的窗体,窗体内容自定。

(8) 打开 MainForm.cs 的设计界面,设计主窗体功能,界面效果如图 1-1(b)所示。

1) 按照与系统登录窗体相似的设计步骤,从工具箱中向设计窗体添加对应的控件,并修改相应的属性。

2) 将显示信息为“窗体基本设计练习”的 Label 控件的【AutoSize】属性改为“False”,【BorderStyle】属性改为“Fixed3D”,【TextAlign】属性改为“MiddleCenter”,然后拖动该控件右下角,使其变为合适大小。

3) 将3个 TextBox 的【Name】属性分别设置为“textBoxName”、“textBoxBaseInfo”和“textBoxOtherInfo”;将3个按钮的【Name】属性分别设置为“buttonFillText”、“buttonShowDialog”和“buttonShowNewForm”。

4) 同时选中 GroupBox 内的多个控件,分别练习快捷工具条中的排列布局按钮提供的功能,例如“使宽度相同”、“使垂直间距相等”、“使大小相同”等功能,观察调整后的情况。如果布局结果不是自己希望的效果,可以按<Ctrl>+Z 组合键取消最近一次的调整操作。

5) 分别双击“自动填写文本框”、“弹出对话框”和“弹出新窗体”3个按钮,添加对应的 Click 事件代码。

```

private void buttonFillText_Click(object sender, EventArgs e)
{
    textBoxName.Text = "张三";
    textBoxBaseInfo.Text = "男, 20岁";
    textBoxOtherInfo.Text = "业余爱好: 足球";
}
private void buttonShowDialog_Click(object sender, EventArgs e)
{
    MessageBox.Show("OK");
}
private void buttonShowNewForm_Click(object sender, EventArgs e)
{
    OtherForm otherForm = new OtherForm();
    otherForm.ShowDialog();
}

```

这一步的目的主要是为了让读者了解如何通过代码修改文本框内容，如何弹出对话框以及如何弹出新的 Windows 窗体。

(9) 按<F5>键运行程序，在系统登录窗体中，单击【确定】按钮，观察随后显示的主窗体的运行效果。在主窗体中，依次按<Tab>键，观察光标焦点顺序是否和想像的一致，然后最后结束程序运行。

(10) 选择【视图】→【Tab 键顺序】命令，分别按照自己希望的顺序依次单击各个 TextBox 对象，完成后，按<Esc>键结束<Tab>键顺序设置。注意：如果不希望某个控件用<Tab>键获取焦点，可以设置其【TabStop】属性为“False”。

(11) 在 LoginForm.cs 的代码编辑窗体中，利用鼠标左键选中下面的代码：

```

private void buttonCancel_Click(object sender, EventArgs e)
{
    Program.IsValidUser = false;
    this.Close();
}

```

然后单击快捷工具栏的【注释选中行】按钮，将这几行代码作为注释。

(12) 再次运行应用程序，观察出错提示。在提示的对话框中单击【否】按钮。注意出现编译错误时不要单击【是】按钮，因为继续运行没有意义。

(13) 选中刚才变为注释的行，单击快捷工具栏的【取消对选中行的注释】符号，再次运行程序，观察是否还会出现编译错误。

(14) 在【解决方案资源管理器】中，双击 MainForm.Designer.cs，观察自动生成的代码，体会设计窗体与自动生成的代码之间的关系，注意观察时不要修改自动生成的任何代码。

(15) 选择【文件】→【退出】命令，结束项目编辑。

(16) 定位到 C:\CsharpExperiment\WindowsFormsExec 文件夹，依次查看该文件夹下的各个子文件夹以及子文件夹下的文件，体会哪些是源代码文件、哪些是自动生成的文件、哪些是编译后生成的可执行文件。

(17) 将 C:\CsharpExperiment\WindowsFormsExec 文件夹下的所有内容全部备份到分配给自己的网络映射驱动器空间中，或者直接备份到 U 盘中。注意不要只备份其中某一个文件，应该将整个 WindowsFormsExec 文件夹全部备份。

另外还要注意，以后再次打开这个项目前，仍需要先将该文件夹下的所有文件和子文件夹复制到本地硬盘中，不要在 U 盘或者网络映射驱动器中直接打开。

1.1.5 实验报告中要求回答的问题

(1) 在 Windows 窗体应用程序项目的界面布局中,你总结出了哪些布局技巧?还发现了哪些你认为有价值的布局技巧?

(2) 从 MainForm.Designer.cs 中的代码和 MainForm.cs 的设计窗体的对照比较中,你能得到什么结论?受到哪些启发?

1.2 实验二 C#基本编程方法练习

本实验主要是为了让读者练习如何用 Windows 窗体应用程序实现指定的功能,掌握用 C#语言编写应用程序代码的基本方法。

建议用 4 小时完成本实验。

1.2.1 实验目的

用 Windows 窗体应用程序实现超级存款计算器的功能。读者通过本实验,应达到如下目标:

- (1) 练习 C#中变量声明和赋值的方法。
- (2) 练习类型转换的方法。
- (3) 练习分支语句的基本用法。
- (4) 练习循环语句的基本用法。

1.2.2 实验内容

为银行个人存款客户提供一个“超级存款计算器”,以简单直观的操作界面为客户提供一个银行存款本息到期金额结算查询程序,以便客户决定选择哪种存款方式。

程序运行效果如图 1-3 所示。

输入信息		结果
存款金额(元)	2000	存款金额: 2000元
年利率(%)	3.23	年利率: 3.23%
存期(年)	2	存期: 2年
利息计算方式	按月计息	到期结算金额:
		2133.28元

图 1-3 超级存款计算器运行效果

当用户输入存款金额及相应信息后,单击【计算】按钮,程序能自动在【到期结算总额】中显示到期应得的本金和利息合计总金额。如果输入内容不符合要求,弹出对话框显示对应的提示信息。

具体要求如下。

- (1) 初始存款金额不能低于 100 元,否则不进行计算并弹出对话框显示相应的提示信息。

(2) 用下拉框显示利息计算方式, 下拉框提供“按月计息”、“按季度计息”和“按年计息”3种选项。

按月计息是指每月计算一次应得的利息, 并将应得的利息作为新存款, 自动添加到用户的现有存款金额中。例如, 存款人第一次存入金额 1 000 元, 年利率为 2%, 则第一个月的利息为 $1\,000 \times (0.02 \div 12)$ 元, 第二个月的利息为 $(1\,000 + \text{第一个月的利息}) \times (0.02 \div 12)$ 元, 依次类推。

按季度计息是指每季度计算一次应得的利息, 并将应得的利息作为新存款添加到用户存款金额中。例如, 存款人第一次存入金额 1 000 元, 年利率为 2%, 则第一个季度的利息为 $1\,000 \times (0.02 \div 4)$ 元, 第二个季度的利息为 $(1\,000 + \text{第一个季度的利息}) \times (0.02 \div 4)$ 元, 依次类推。

按年计息是指每年计算一次应得的利息, 并将应得的利息作为新存款添加到用户存款金额中。例如, 存款人第一次存入金额 100 元, 年利率为 2%, 则

第一年的利息 x_1 : 100×0.02 元, 第一年结算余额 y_1 : $100 + x_1$

第二年的利息 x_2 : $y_1 \times 0.02$ 元, 第二年结算余额 y_2 : $y_1 + x_2$

第三年的利息 x_3 : $y_2 \times 0.02$ 元, 第三年结算余额 y_3 : $y_2 + x_3$

.....

(3) 到期结算的总金额四舍五入到小数点后两位。

1.2.3 实验要求

要求程序中至少提供下列方法。

① Caculate 方法

功能: 计算到期结算金额。

方法原型:

```
private float Caculate(float startAmount, float rate, int count)
```

其中, startAmount 表示存款金额, rate 表示利率, count 表示叠加次数。返回值为到期金额。

② ConvertStringToNumber 方法

功能: 将大于零的字符串转换为 32 位整数或者 64 位浮点数, 并指明转换是否成功。

方法原型 (提供两种重载形式):

```
private bool ConvertStringToNumber(string s, bool mustGreatThanZero, out int result)
```

```
private bool ConvertStringToNumber(string s, bool mustGreatThanZero, out float result)
```

其中, s 为被转换的字符串, mustGreatThanZero 为是否要求被转换的内容必须大于零, result 为转换后的 32 位整数或者 64 位浮点数。

1.2.4 实验提示

(1) 创建一个名为 SuperCalculator 的 Windows 窗体应用程序项目。

(2) 通过窗体的 Shown 事件, 让窗体界面显示时光标默认在存款金额文本框中闪烁。

```
private void FormMain_Shown(object sender, EventArgs e)
{
    textBoxStartAmount.Focus();
}
```

(3) 通过 GroupBox 控件的 Enter 事件, 保证修改输入信息中任何一个内容时, 到期结算金额中都不能显示值, 而只有单击【计算】按钮才显示结算结果。

```
private void groupBox1_Enter(object sender, EventArgs e)
```

```
{
    //保证修改任一输入值时，不显示计算结果
    LabelResult.Text = string.Empty;
}
```

(4) 在【计算】按钮的 Click 事件中，先判断输入信息是否符合要求，然后根据利息计算方式计算到期结算金额。

实现代码中可以利用 ComboBox 控件的【SelectedItem】属性判断用户选择的值，利用【SelectedIndex】属性判断是否选择了提供的选项。例如

```
if (comboBoxCalculateType.SelectedIndex == -1)
{
    MessageBox.Show("请选择提供的利息计算方式");
    return;
}
```

即如果没有选择任一个选项，SelectedIndex 属性返回-1。

(5) 如果希望通过代码设置程序开始运行时窗体的起始位置在屏幕中间，可以在构造函数中添加如下代码。

```
public MainForm( )
{
    InitializeComponent( );
    this.StartPosition = FormStartPosition.CenterScreen;
}
```

这里有一个输入技巧：输入完 this.StartPosition 后面的“=”后，直接按空格键，系统会自动出现 FormStartPosition，然后直接键入“.”，在选择希望的枚举值后按回车键。凡是以枚举类型出现的都可以采用这种办法提高键入代码的速度。再举一个例子，输入

```
MessageBox.Show("aa", "bb", MessageBoxButtons.YesNo, MessageBoxIcon.Asterisk);
```

输入技巧为：输入“bb”后面的“,”后，按空格键，出现 MessageBoxButtons，选择 YesNo，按回车键，然后键入逗号，再按空格键，出现 MessageBoxIcon，选择 Asterisk，按回车键。最后输入其他内容。

如果按空格键后没有出现希望的选项，可以先用上下箭头键选择合适的重载方法，然后按空格键。

另外，在已有代码的前面增加内容时，如果希望输入时就有智能提示，当输入一个字母后还没有提示，可以先输入一个空格，然后在空格的前面添加内容就有提示了。

(6) 使用已经声明过的变量、对象名，或者输入类名时，或者键入“.”后，如果应该有对应的智能提示却没有看到智能提示时，说明前面的输入肯定有问题，这时不要再继续输入内容了，因为再输入的内容肯定也是错的，而应该思考一下问题在哪，这样才能有效地提高编写和调试程序的效率。

(7) 将重复使用的功能，用单独的方法实现，并使用 XML 注释方式给方法添加注释。使用 XML 注释方式时，注意一定要先定义好方法及参数，然后再在其上面按“///”添加注释。

(8) 下面列出 MainForm.cs 的部分参考代码。

```
public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();
        this.StartPosition = FormStartPosition.CenterScreen;
    }
}
```

```
string[] cac1Type = { "按月计息", "按季度计息", "按年计息" };
comboBoxCalculateType.Items.AddRange(cac1Type);
comboBoxCalculateType.SelectedIndex = 0;
labelResult.Text = string.Empty;
}
private void MainForm_Shown(object sender, EventArgs e)
{
    textBoxStartAmount.Focus();
}
private void buttonOK_Click(object sender, EventArgs e)
{
    //存款金额
    int startAmount;
    //年利率
    float yearRate;
    //存期
    int years;
    if (!ConvertStringToNumber(textBoxStartAmount.Text, true, out startAmount))
    {
        MessageBox.Show("存款金额输入有错!", "提示",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    if (startAmount < 100)
    {
        MessageBox.Show("存款金额不能小于100元", "提示",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    if (ConvertStringToNumber(textBoxYearRate.Text, true, out yearRate) == false)
    {
        MessageBox.Show("年利率输入有错!", "提示",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    yearRate /= 100.0f;
    if (ConvertStringToNumber(textBoxYears.Text, true, out years) == false)
    {
        MessageBox.Show("存期(年)输入有错!", "提示",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    if (comboBoxCalculateType.SelectedIndex == -1)
    {
        MessageBox.Show("请选择提供的利息计算方式", "提示",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    labelParameter.Text =
        string.Format("存款金额: {0}元{3}{3}年利率: {1}%{3}{3}存期: {2}年",
            startAmount, yearRate * 100, years, Environment.NewLine);
    switch (comboBoxCalculateType.SelectedItem.ToString())
    {

```