

21 世纪

计算机应用技术系列规划教材

C#网络应用编程

实验指导与 开发实例

◎ 马骏 编著 ◎



人民邮电出版社
POSTS & TELECOM PRESS

21 世纪计算机应用技术系列规划教材

C#网络应用编程实验 指导与开发实例

马 骏 编著

人 民 邮 电 出 版 社

北 京

图书在版编目 (CIP) 数据

C#网络应用编程实验指导与开发实例/马骏编著. —北京: 人民邮电出版社, 2007.3
(21 世纪计算机应用技术系列规划教材)

ISBN 978-7-115-15765-2

I. C... II. 马... III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 007073 号

内 容 提 要

本书是《C#网络应用编程基础》和《C#网络应用高级编程》的配套教材, 内容包括上述两本教材的上机实验指导和 3 个比较大的应用开发实例。

上机实验指导部分充分考虑到学生学习的进度和编程思路, 精心设计在 VS2005 开发平台下使用 C#编写的有代表意义的上机实验内容, 并在实验步骤提示中介绍如何逐步完成所需的功能, 使学生通过提示和上机练习, 快速掌握 C#编程方法和基本的设计技巧。

应用开发实例分别介绍数据库应用、网站开发以及智能客户端的具体设计与实现技术。从项目分析、界面设计、代码实现到安装程序的制作与发布, 实例部分均作了详细介绍。

本书可作为高等院校计算机及相关专业的教材, 也适合初、中级程序员以及想利用 C#进行更复杂的应用编程的人员阅读。

21 世纪计算机应用技术系列规划教材

C#网络应用编程实验指导与开发实例

-
- ◆ 编 著 马 骏
责任编辑 张 鑫
 - ◆ 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>
三河市海波印务有限公司印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 17.25
字数: 420 千字
印数: 1 ± 3 000 册
 - 2007 年 3 月第 1 版
2007 年 3 月河北第 1 次印刷

ISBN 978-7-115-15765-2/TP

定价: 25.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

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,还有一种比较快速、有效的学习途径,就是练习开发一些实际的项目。对于从未参加过项目开发的初学者,这种训练显得非常重要。

本书正是为了这个目的而编写的,除了与教材配套的上机实验指导外,作者还精心编写了3个实际应用开发的例子,设计中使用的技术涵盖了《C#网络应用编程基础》和《C#网络应用高级编程》教材中的很多知识点。掌握这些技术是巩固所学内容,进而做到举一反三,将一个个较小的基本编程训练灵活运用于实际开发的捷径。

本书的所有程序均在 Microsoft VS2005 Professional 简体中文版开发环境下调试通过。读者通过阅读、上机练习和调试运行,能快速理解用 C#设计实际应用项目的方法和技巧,为提高实际开发能力打下良好的基础。另外,由于本书是《C#网络应用编程基础》和《C#网络应用高级编程》教材中的综合应用配套教材,因此需要读者具有一定的 C#编程基础。实际动手进行调试是掌握本书知识的一个非常重要的环节,希望引起读者的高度重视。

本书由马骏编著。参与代码验证以及文字校对等工作的还有陈明、王芳、杨韶华、张瑞青、郑珂等。为了配合学习需要,作者还提供了与本书配套的所有源程序代码。读者可以到人民邮电出版社网站(<http://www.ptpress.com.cn>)下载,也可以与马骏联系(mj@henu.edu.cn)。

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
2006 年 12 月

目 录



第1章 上机实验指导	1
1.1 实验一 创建简单的.NET 应用程序	1
1.2 实验二 C#基本编程方法	5
1.3 实验三 面向对象的编程基础	8
1.4 实验四 界面设计与文件存取	21
1.5 实验五 ADO.NET 与数据操作	23
1.6 实验六 ASP.NET 网页设计	25
1.7 实验七 ASP.NET 网站开发	28
1.8 实验八 简单网络聊天系统	34
1.9 实验九 网络呼叫应答提醒系统	35
1.10 实验十 文件数据加密与解密	39
第2章 数据库应用开发实例	41
2.1 高校成人教育学籍管理系统概述	41
2.1.1 业务流程及基本需求	41
2.1.2 系统总体功能设计	42
2.2 数据库设计	42
2.2.1 数据表划分及逻辑结构	42
2.2.2 存储过程设计	44
2.3 基础模块类设计	46
2.3.1 CommonMethod 类	46
2.3.2 UserControlJbqk 用户控件	48
2.3.3 UserControlCj 用户控件	51
2.4 登录模块与主窗体设计	56
2.4.1 登录密码加密解密处理	56
2.4.2 登录窗体界面设计	59
2.4.3 主功能窗体界面设计	63

2.5 编码管理模块	64
2.5.1 设计编码管理通用类	64
2.5.2 修改主窗体菜单实现各种编码管理功能	69
2.6 基本情况管理模块	69
2.6.1 录入基本情况	70
2.6.2 成批数据修改	77
2.7 课程成绩管理模块	78
2.8 查询统计模块	88
2.8.1 查询基本情况	88
2.8.2 查询成绩	89
2.8.3 统计总人数	91
2.8.4 统计不及格人数	91
2.8.5 统计学期总成绩	91
2.9 报表输出模块	94
2.9.1 基本情况表	94
2.9.2 成绩表	97
2.9.3 点名册	99
2.9.4 学籍表	101
2.10 数据备份与恢复模块	104
2.10.1 备份 SQL Server 数据库	105
2.10.2 恢复数据	108
2.11 其他功能模块	113
2.11.1 修改登录密码	113
2.11.2 注销某年级数据	115
2.12 制作成人教育学籍管理系统安装程序	116
第3章 网站开发实例	121
3.1 辅助学习网站功能描述	121
3.1.1 基本需求简介	121
3.1.2 模块功能划分	121
3.2 设计网站使用的数据库与存储过程	122
3.2.1 数据库中的表结构设计	122
3.2.2 存储过程设计	123
3.3 网站配置	125
3.4 样式设计	126
3.4.1 再谈 CSS 样式	126
3.4.2 主题和外观	127
3.4.3 定义网站主题	128
3.5 网页布局中的元素定位	130

3.6 母版页设计.....	131
3.7 CommonMethods 类.....	133
3.8 Global.asax 文件	137
3.9 用户注册登录模块	138
3.9.1 用户登录模块	138
3.9.2 用户注册模块	141
3.10 网站主页.....	146
3.10.1 防止用户绕过登录页面	146
3.10.2 主页设计	147
3.11 课件下载模块.....	152
3.12 技术论坛模块	154
3.12.1 主论坛	154
3.12.2 子论坛	157
3.12.3 发布与修改论坛信息	162
3.13 编程开发模块	167
3.13.1 文章分类索引模块	167
3.13.2 文章内容显示模块	170
3.14 资料上传模块	172
3.15 智能搜索模块	175
3.15.1 站内搜索	176
3.15.2 站外搜索	180
3.16 搞笑图片模块	183
3.17 奇闻趣事模块	185
3.18 音频视频模块	187
3.18.1 音乐欣赏	188
3.18.2 Flash 欣赏	190
3.18.3 视频欣赏	193
3.19 人气统计模块	195
3.20 请您留言模块	197
3.21 发布辅助学习网站	199
第4章 智能客户端应用开发实例	202
4.1 智能客户端基本知识	202
4.1.1 通过网站提供客户端软件下载	202
4.1.2 客户端软件版本测试与自动升级	203
4.2 网络对战五子棋游戏功能描述	203
4.2.1 网络对战五子棋游戏规则描述	203
4.2.2 服务器与客户端通信描述	203
4.3 网络对战五子棋游戏的界面设计与代码实现	208

4.3.1	设计五子棋房间控件	208
4.3.2	网络对战五子棋服务器端设计	214
4.3.3	网络对战五子棋客户端设计	237
4.4	网络对战五子棋的发布与升级	263
4.4.1	发布与传统安装的区别	263
4.4.2	发布网络对战五子棋客户端程序	264

第1章

上机实验指导



网络应用开发的例子非常多,为了让读者对 C# 网络应用编程的各个方面都有一个大概的认识,本章上机实验指导以实际业务处理为素材,根据课程进度选取了有代表意义的部分内容,经过简化和变形处理,整理为一个个单独的实验。

考虑到学习的进度,在安排每个上机实验内容时,功能需求及实现步骤均与实际的业务处理有非常大的差别,而且有些步骤是多余的,或者实现的办法不是最简单的。但完成实验的目的不仅仅是为了实现需要的功能,还要通过这些步骤领会涉及的多种技巧。另外,由于各个实验之间既有独立性,又相互关联,所以读者按照实验顺序完成一个个实验后,既能对网络应用编程形成一个连贯的思路,又能快速掌握和巩固涉及的技术。

1. 上机实验环境

操作系统: Windows XP SP2 或者 Windows 2000 SP4。

开发工具: VS2005 Professional 简体中文版。

内存要求: 至少 256MB。

2. 实验报告要求

(1) 使用专用的统一的实验报告纸,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、设计界面及关键源代码,以及上机调试过程中遇到的问题和解决办法。

1.1 实验一 创建简单的.NET 应用程序

1. 实验目的

熟悉 VS2005 开发环境,掌握如何在此开发环境下开发简单的.NET 应用程序,以及调试程序的基本操作技巧。

2. 建议用时

建议用 2 小时完成本实验,或者根据自己的理解情况调整实际需要的时间。

3. 实验内容

分别创建不同类型的.NET 应用程序项目，体会基本的设计与编程方法。

4. 实验要求

(1) 通过实验掌握【工具箱】、【属性】窗口、【解决方案资源管理器】等的用法和基本操作技巧。

(2) 通过实验观察各种应用程序的程序结构及特点。

(3) 通过实验观察生成的可执行文件的存放位置，掌握项目备份与恢复的方法。

(4) 通过实验掌握利用断点进行程序调试的方法。

5. 实验步骤 1: 创建一个简单的 Windows 应用程序项目

(1) 运行 Microsoft Visual Studio 2005，创建一个 Windows 应用程序项目，在【名称】文本框中将项目名改为 SimpleWindowsApplication，在【位置】文本框中输入保存的目录位置 C:\CSharpExperiment，选中【创建解决方案的目录】复选框，如图 1-1 所示，然后单击【确定】按钮。

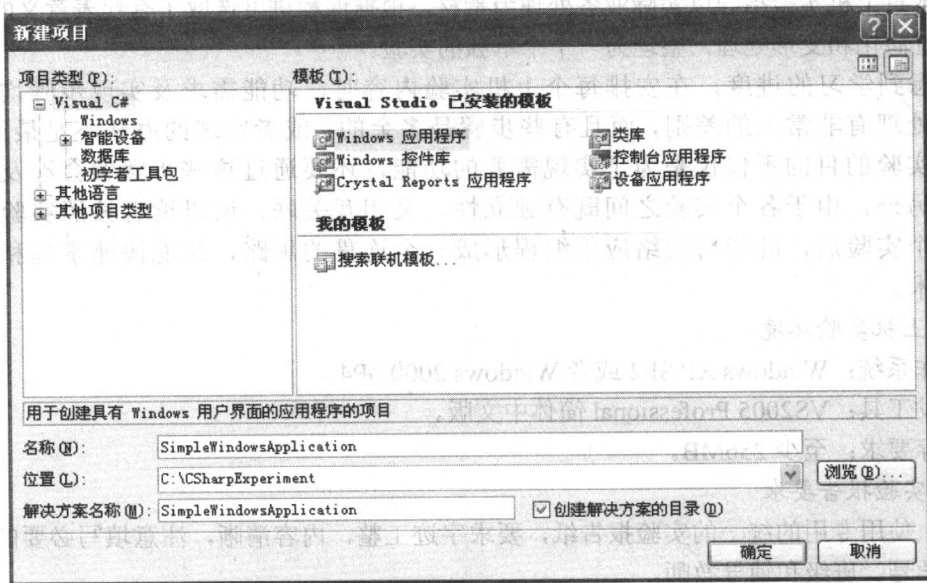


图 1-1 新建 Windows 应用程序项目

注意：千万不要将【位置】设置为网络映射的驱动器，一定要确保保存位置为本地硬盘的逻辑驱动器。原因有两点：一是用本地硬盘速度快；二是为了确保有足够的读写权限。特别是以后对数据库的操作以及网站设计，要求必须是本地硬盘，而网络映射的驱动器可能会因为没有相应的读写权限导致无法调试。

另外还要注意，不要使用#之类的符号作为项目名称，最好用英文单词或者汉语拼音的组合给项目起名，并且使用有意义的名称。

(2) 在【解决方案资源管理器】中，单击 Form1.cs，重命名 Form1.cs 为 FormMain.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) 单击【**BackgroundImage**】属性右边的“…”，导入一个本地资源类型的图片。然后选择 Form1.cs 的设计窗体，分别选择不同的【**BackgroundImageLayout**】属性值，观察窗体的背景图布局效果，然后删除该背景图片。

(4) 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 3 个，然后分别修改其【**Text**】属性为“编号”、“姓名”、“年龄”和“出生日期”。

(5) 选中 label1 对象，修改【**AutoSize**】属性为“False”，拖动该控件右下角改变大小，观察变化，然后设置其【**Font**】属性为“隶书，2 号，斜体”，选择【**ForeColor**】属性为红色，再分别选择不同的【**TextAlign**】属性值，观察显示的效果，并思考如何使某行的信息居中显示。

(6) 选择【工具箱】，双击 TextBox 控件 4 次，观察窗体中添加的情况，然后分别拖动窗体中的 4 个 TextBox 对象，放在对应的 Label 对象右边，并分别修改其【**Name**】属性为“textBoxId”、“textBoxName”、“textBoxAge”和“textBoxBirthday”。

(7) 设置 textBoxName 的【**Size**】属性为“200, 21”，然后按住鼠标左键，同时选中 4 个 TextBox 对象，在快捷工具栏中，分别选择“使宽度相同”、“使垂直间距相等”，观察变化。再选择“顶部对齐”，观察 4 个对象是不是叠在了一起，然后按<Ctrl>+<Z>组合键取消这次操作。

(8) 按<F5>键编译并运行，依次按<Tab>键，观察光标焦点顺序是否和想象的一致。然后结束程序运行。

(9) 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”。

(10) 从【工具箱】中向设计窗体拖放 1 个 Button 控件，然后双击该对象，观察自动生成的代码，并在 button1_Click 事件中添加一行代码：this.Close();。

(11) 运行该应用程序，然后单击【button1】按钮结束程序运行。

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

```
private void button1_Click(object sender, EventArgs e)
{
    this.Close();
}
```

然后单击快捷工具栏的【注释选中行】符号，将这几行代码作为注释。

(13) 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：调试时只要出现编译错误，不要单击【是】按钮，因为继续运行没有意义。

(14) 切换到 FormMain.cs 的代码编辑窗体，同时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，然后单击【button1】按钮结束应用程序运行。

(15) 在 FormMain.cs 的设计窗体中，修改 button1 的【**Name**】属性为“buttonOK”，【**Text**】属性为“确定”，然后切换到 FormMain.cs 的代码编辑窗体，修改 button1_Click 为 buttonOK_Click，运行观察编译出错提示，然后单击【否】按钮放弃运行。

(16) 切换到 FormMain.cs 的设计窗体，单击【确定】按钮，然后单击【属性】窗口中的雷电图标，在“Click”事件右边的下拉列表中选择“buttonOk_Click”，再次运行程序。

(17) 切换到 FormMain.cs 的代码编辑窗体, 在 buttonOK_Click 上方单击鼠标右键, 在弹出的快捷菜单中选择【重构】→【重命名】命令, 将 buttonOK_Click 改为 buttonExit_Click, 重新运行程序, 观察是否有编译错误, 然后结束程序运行, 体会重构的作用。

(18) 在【解决方案资源管理器】中, 双击 FormMain.Designer.cs, 观察自动生成的代码, 了解设计窗体与自动生成的代码之间的关系。注意: 不要修改自动生成的任何代码。

(19) 在【解决方案资源管理器】中, 双击 Program.cs, 观察 Mian 方法内的代码。

(20) 选择【文件】→【退出】命令, 结束项目编辑。

(21) 找到 C:\CSharpExperiment, 用鼠标右键单击该目录, 从快捷菜单中选择【复制】命令, 将该目录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: 不要只备份某一个文件, 应该将整个目录全部备份。

另外还要注意: 以后再次打开这个项目前, 仍需要先将该目录复制到本地硬盘中, 不要在 U 盘或者网络映射驱动器中直接打开。

(22) 依次查看 C:\CSharpExperiment 目录下的各个子目录以及子目录下的文件, 体会哪些是源代码文件, 哪些是自动生成的文件。

6. 实验步骤 2: 创建一个简单的控制台应用程序项目

(1) 重新运行 VS2005, 新建一个名为 SimpleConsoleApplication 的控制台应用程序。

(2) 编写程序完成下列功能:

◆ 从键盘接收一个字符串, 如 “my friend”;

◆ 输出相应的欢迎信息, 如 “Welcome: my friend”。

提示: 输入字母时, 智能帮助会自动提示相应的信息, 可以直接按回车键或者<Tab>键接受提示的信息。例如,

输入到 conso 时智能帮助已经提示到 Console, 直接按回车键即可。

输入 f 时智能帮助已经提示到 for, 直接按回车键, 然后按<Tab>键, for 语句的整个结构就自动出来了, 再依次按<Tab>键修改各参数的内容。如果不需要修改了, 可以直接按回车键, 此时光标会直接转到循环体内部。

还有, 如果将最后一个大括号删除, 再重新添加上, 系统会自动对整个代码按照嵌套层次进行统一的缩进处理, 轻松解决了手工调整的麻烦。

其他编辑技巧请实验者自己总结。

实现这一步功能的代码如下:

```
static void Main(string[] args)
{
    Console.Write("请输入一个字符串: ");
    string welcomeString = Console.ReadLine();
    Console.WriteLine("Welcome: {0}", welcomeString);
    Console.ReadLine();
}
```

(3) 在此基础上再增加一些语句, 然后设置一些断点, 体会利用断点调试程序的方法。

(4) 结束项目调试, 将该项目备份到分配给自己的网络空间或者 U 盘中。

7. 实验步骤 3: 创建一个简单的网站

(1) 重新运行 VS2005, 新建一个网站。注意, 在弹出的窗体中, 【位置】下拉列表中的选项应该选择为“文件系统”, 【位置】的右边是要求将网站保存到的具体目录, 在其中输入

C:\CSharpExperiment\SimpleWebSite, 单击【确定】按钮。

(2) 修改<body>为<body style="text-align:center;">。

(3) 修改<div>为<div style="width:760px; height:400px;">。

注意: 键入“width:760px;”后, 按空格键, 然后键入“height”, 键入过程中智能帮助会提示相应的选项。

(4) 切换到设计视图, 在 div 区域内输入一句话, 如“我的第一个网站”, 然后在其下面添加 1 个 Button 控件, 设置其【Width】属性为“70%”, 【ID】属性为“buttonOK”, 【Text】属性为“确定”。

(5) 用鼠标右键单击该标签, 从快捷菜单中选择【样式】命令, 将字体大小设为“20pt”, 选择颜色为自定义的红色。

(6) 用鼠标拖动调整 div 区域的宽度, 观察【确定】按钮的变化。

(7) 单击鼠标右键, 从快捷菜单中选择【在浏览器中查看】命令, 观察结果。注意: 第一次浏览时系统需要建立缓存, 因此速度比较慢, 缓存以后就快了。

(8) 双击【确定】按钮, 在代码隐藏类的 buttonOK_Click 事件中输入:

```
Response.Write("<script>alert('哈哈');</script>");
```

(9) 按<F5>键调试运行, 观察弹出的警告对话框提示信息, 然后直接单击【确定】按钮, 观察运行结果。

8. 实验报告中要求回答的问题

(1) 在 Windows 应用程序项目的基本操作中, 你总结出了哪些操作技巧? 还发现了哪些你认为有价值的操作技巧?

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

1.2 实验二 C#基本编程方法

1. 实验目的

(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。

2. 建议用时

建议用 4 小时完成本实验, 或者根据自己的理解情况调整实际需要的时间。

3. 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。要求初始界面如图 1-2 所示。

图 1-2 超级存款计算器设计界面

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。具体要求如下。

(1) 存款金额不能低于 100 元,否则不进行计算并弹出对话框提示相应信息。

(2) 利息计算方式提供按月算息、按季度算息和按年算息 3 种形式。

按年算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 : $y_2 + x_3$

.....

按季度算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。

按月算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。

(3) 到期结算总金额,要求输出结果四舍五入到小数点后两位。

4. 实验要求

(1) 要求用 `startAmount` 表示初始存款金额。

(2) 要求用 `yearRate` 表示年利率。

(3) 要求用 `years` 表示年数。

(4) 要求用 `calculateFrequency` 保存用户选择的计算方式,即“按月算息”、“按季度算息”和“按年算息”。当用户在【利息计算方式】中选择某个计算方式后,程序会根据选择结果对 `calculateFrequency` 赋以相应的字符串值,如赋值为“按月算息”。

(5) 要求用 `rate` 表示按选择的利息计算方式使用的利率。

(6) 要求将计算出的结算总金额赋给 `total` 变量,并在只读的 `textBoxTotal` 中显示结果。

(7) 要求程序中提供下列方法。

① Caculate 方法

功能: 计算到期结算金额。

方法原型:

```
private double Caculate(double startAmount, double 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 str, bool mustGreatThanZero, out int result)
```

```
private bool ConvertStringToNumber(string str, bool mustGreatThanZero, out double result)
```

其中, `str` 为被转换的字符串, `mustGreatThanZero` 为是否有必须大于零的要求, `result` 为转换后的 32 位整数或者 64 位浮点数。

5. 实验步骤提示

(1) 创建一个名为 `SuperCalculator` 的 Windows 应用程序, 修改 `Form1.cs` 为 `FormMain.cs`, 然后完成图 1-2 的设计界面。

(2) 在 `comboBoxCalculateFrequency` 的 **【Items】** 属性中输入按月计算、按季度计算和按年计算三个选项。

(3) 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();
}
```

(4) 想办法用一个事件, 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
    textBoxTotal.Clear();
}
```

(5) 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 **【SelectedItem】** 属性判断选择的值, 利用 **【SelectedIndex】** 属性判断是否选择了提供的选项。例如

```
if (comboBoxCalculateFrequency.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。

(6) 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 `FormMain` 的构造函数中添加如下代码。

```
public FormMain()
{
    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。

(7) 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。

(8) 将重复使用的功能,用单独的方法实现,并使用 XML 注释方式给方法功能以及提供的参数添加对应的注释。使用 XML 注释方式时,注意一定要先定义好方法及参数,然后在其上面按“///”添加注释。

6. 实验报告中要求回答的问题

(1) 画出包含输入和输出数据的程序运行界面示例。

(2) 运行设计的程序,在实验报告中列出计算后的表 1-1 的内容,并说明程序计算结果和手工计算结果是否符合。

表 1-1 部分存款计算结果

初始金额(元)	利率(%)	年 数	计 算 方 式	到期结算总额(元)
1 000	2	5	按月计息	
			按季度计息	
			按年计息	
3 500	3.3	7	按月计息	
			按季度计息	
			按年计息	
5 000	6.25	10	按月计息	
			按季度计息	
			按年计息	

(3) 写出实验中遇到的问题及解决方法。

(4) 所有功能实现后,与参考解答对照,写出自己设计的程序与参考解答相比有什么优缺点。

1.3 实验三 面向对象的编程基础

1. 实验目的

- (1) 练习如何创建类和对象。
- (2) 练习如何为定义的类编写相应的方法。
- (3) 练习如何通过属性访问对象中的数据。
- (4) 练习如何创建基类及其派生类。

2. 建议用时

建议用 6 小时完成本实验,或者根据自己的理解情况调整实际需要的时间。

3. 实验说明

本实验对大部分非本科生来说,独立完成实现的功能可能有一定难度,教师可以根据实际情况决定是否要求非本科生完成此实验。如果要求完成,最好在实验之前先讲解一下参考解答的程序。

4. 实验内容

在个人银行存款业务中,不同银行规定有不同的账户类型,如整存整取、零存整取、存本取息、通知存款、定额定期、定活两便和活期储蓄等。本实验不处理这么复杂的内容,而是假定只提供两种账户,一种是活期存款账户,另一种是定期存款账户。

实验要求为个人银行存款账户定义两个类,一个是活期存款账户 `CheckingCustom` 类,另一个是定期存款账户 `FixedCustom` 类。要求用户操作界面如图 1-3 所示。

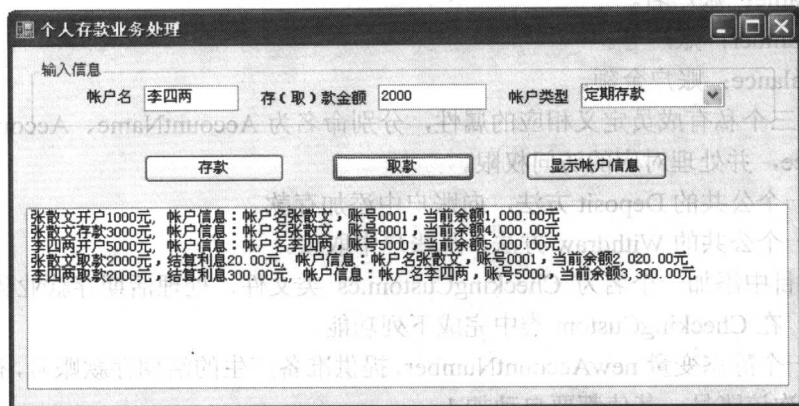


图 1-3 个人存款业务处理界面示例

为了简化处理过程。假定实验中的“活期存款”和“定期存款”业务规定及功能实现要求如下。

(1) 不论是活期存款账户还是定期存款账户,都可以随时存款和取款,而且规定一个人最多只能有一个活期账户和一个定期账户。创建活期账户时,必须提供账户名、账户号和开户金额,业务处理均以账户名为关键字。

(2) 活期存款账户号的范围必须为 0001~4999 (包括 0001 和 4999)。取款时,不论存款时间有多长,一律按 0.5% 计算利息。

(3) 定期存款账户号的范围必须为 5000~9999 (包括 5000 和 9999)。取款时,不论存款时间有多长,一律按下列方法计算利息:当存款余额大于 500 时利息为 6%,否则利息为 3%。

(4) 每次取款之前,都要先根据当前余额结算一次利息,并将利息附加到余额中,然后再从中取出指定的款数。向现有账户追加存款时,不进行结算。

(5) 要允许用户随时查询所有账户的信息。

(6) 设计的程序要易于扩充,即需要增加存款业务类型时能够利用已经实现的功能,通过尽量少的代码快速实现,不要全部从头开始设计。

5. 实验步骤提示

由于要求程序易于扩充,因此需要找出各种存款业务类型共有的功能,然后用一个基类