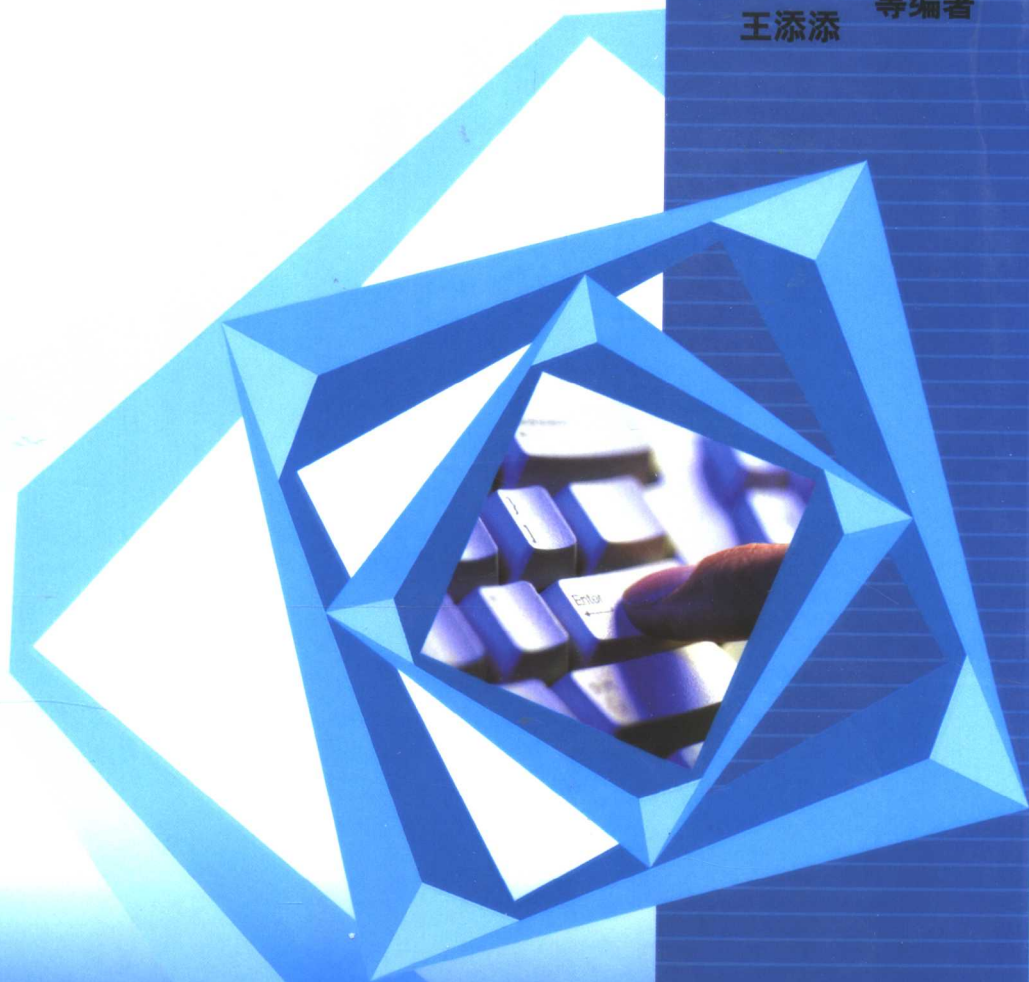


21世纪高等院校课程设计丛书

C#

课程设计案例精编

王振江 等编著
王添添



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校课程设计丛书

C#课程设计案例精编

王振江 王添添 等编著

中国水利水电出版社

内 容 提 要

C#是 Microsoft 在推出.NET 平台的同时推出的基于.NET 平台的托管语言,它简洁高效,继承了 C++和 Java 的诸多优秀特性,使得它卓尔不群。它与 C++和 Java 的高度相似性又大大减少了程序员从其他平台转移的时间,所以它一经推出,马上就受到广大程序员的青睐。

本书从实际应用的角度出发,详细介绍了使用 C#开发.NET 应用程序的方法。书中的 9 个案例由浅入深、从简单到复杂地介绍了使用 C#开发程序的关键技术和编程方法。

本书是在校大学生学习使用 C#进行课程设计的好帮手和重要参考资料,也是研究生撰写毕业论文时的优秀参考文献,同时它还可供广大教师、计算机专业编程人员参考使用。此外,也可以作为中级开发人员的参考书。

本书所有案例的完整源代码可从中国水利水电出版社网站 (<http://www.waterpub.com.cn/softdown/>) 下载。

图书在版编目 (CIP) 数据

C#课程设计案例精编 / 王振江等编著. —北京:中国水利水电出版社,2005
(21 世纪高等院校课程设计丛书)
ISBN 7-5084-3438-2

I. C... II. 王... III. C 语言—程序设计—高等学校:技术学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 138892 号

书 名	C#课程设计案例精编
作 者	王振江 王添添 等编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (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	北京北医印刷厂
规 格	787mm×1092mm 16 开本 13.75 印张 331 千字
版 次	2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷
印 数	0001—5000 册
定 价	22.00 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

丛书序

课程设计是教学计划中的一个重要环节。通过课程设计,学生能够得到系统的技能训练,从而巩固和加强所学的专业理论知识,其目的是培养学生的综合运用能力,使学生成为具有扎实的理论基础和较强的独立动手能力的专业人才。

随着社会对复合型、应用型人才的需要,各高校对学生在课程设计上的要求越来越高,课程设计的选题也越来越需要结合实际情况。考虑到目前课程设计重视实际操作的需要,我们组织具有丰富实际开发和教学经验的老师编写了本套丛书。书中所选的案例皆取自作者平时所做的项目,具有相当强的实用性和可操作性,其中有些案例项目稍加扩充,即可成为一个功能完整的项目系统。

我们编写本套丛书的目的是给学生和老师在做课程设计的时候提供一个参考,老师可以先让学生按照本书例子的引导完成整个案例的制作,然后要求学生仿照该案例或课后习题有所扩充,最后再独立地以自己的想法做出各具特色的课程设计的实例。因为知识的应用过程就是一个熟练的过程,你可以先照着别人的例子作,在亲自实践后才能了解到其中的实际问题,从而在解决问题中掌握更多实用的知识,并巩固和加强所学的专业理论知识。这样才能在走入工作岗位后更好地将学到的知识应用到实际开发中去。

本套丛书具有以下特点:

内容全面、综合:精心选取各课题开发中具有代表性的若干个案例,全面覆盖各课题开发技术中的重点和难点,内容详实。

选例实用、典型:案例的选取具有代表性,是在实际开发工作中经常能遇到的。书中所有的例子都经过验证实现,读者可通过实例的学习对相应技术点有清晰直观的了解。

紧扣课设、实用:丛书最大程度地强调课程设计的特色,书中所有的例子尽量做到按学生课程设计的思路编排,以达到更易读懂、更实用的目的。

注释清楚、明了:对于学习过程中易出现问题的部分都加了详细的注释说明,以便学生实际课程设计制作的过程中思路更清晰、明了。

课后练习选取有代表性:为了方便学生练习和老师布置作业,部分案例后还选取了具有代表性的实例题目,并作了简单的实现思路说明,适合不同层次的学生练习。

通过对本套丛书相应课程设计实例的学习,相信您一定能轻松完成自己的课程设计,做出满意的课程设计作业。

本套丛书目前涉及到的课程设计课题有: Delphi、Visual FoxPro、Visual Basic、Visual C++、PowerBuilder、C、汇编语言、Java、ASP、SQL Server、Access、JSP、Oracle、数据库技术、网页制作等。在以后的时间里我们将加入更多、更实用的课程设计课题供广大读者和老师参考。也敬请广大读者及时和我们沟通,从而使我们能更好地为大家服务。

真诚地感谢参与本套丛书编写的老师们,是他们认真敬业的精神保证了本套丛书能符

合课程设计的要求，能更好地为读者学习应用。也非常感谢中国水利水电出版社万水公司的图书策划编辑，是他们本着为学生、为读者服务的精神策划了本套实用的课程设计案例精编丛书。

期待各位读者的意见和建议，希望各位不吝赐教，来信请至 xinyuanxuan@263.net。最后祝愿各位读者能通过本套丛书学习到更多更实用的知识，为将来的发展奠一块更好的基石。

丛书编委会

2005 年 10 月

前 言

Microsoft 在 2001 年推出的 .NET 框架是用于建立 Internet 时代应用程序的综合性平台, 它允许开发人员建立使用 Windows 平台的完全富于新特色的应用程序。 .NET 框架包含一个内容极其广泛的类框架, 为程序开发人员提供了方便的服务。此外, .NET 框架为安全性、跨平台通信、资源管理和应用扩展性提供了完整的服务。

C# 和 .NET 平台一起推出, 它是和 .NET 平台结合最紧密的语言, 也是最能完美地表现 .NET 平台架构的语言, 它继承了 C++ 和 Java 的很多优点, 摒弃了其中一些不安全或者应用比较麻烦的地方, 可以说, 它集众家之长, 而少众家之短, 是一种非常简洁高效的语言, 可以使用它来高效开发高性能的应用程序。

由于它与 .NET 平台的高度兼容性, C# 使得程序员可以快速地编写各种基于 Microsoft .NET 平台的应用程序, Microsoft .NET 提供了一系列工具和服务来最大限度地开发利用计算机与通信应用程序。C# 使得 C++ 程序员可以高效地开发程序, 而绝不会损失 C/C++ 原有的强大的功能。因为这种继承关系, C# 与 C/C++ 具有极大的相似性, 熟悉类似语言的开发者可以很快地转向 C#。

C# 程序员可以在 Microsoft .NET 平台上事半功倍地构建应用程序的扩展框架。C# 包含了内置的特性, 使任何组件可以轻松转化为 XML 网络服务, 并通过 Internet 被任何操作系统上运行的任何程序调用。

更突出的是, XML 网络服务框架可以使程序员像对待 C# 对象一样使用现有的 XML 网络服务。这样, 程序员就可以方便地使用他们已有的面向对象编程技巧来开发利用现有的 XML 网络服务。

另外还有一些非常优秀的特性, 使得 C# 成为一流的网络编程工具。例如, XML 正逐渐成为在网络上传输结构化数据的标准。这种数据集合往往非常小。为提高性能, C# 允许把 XML 数据直接映射到结构数据类型, 而不是类数据类型。这样对处理少量的数据非常有效。

本书利用实例由浅入深地介绍了使用 C# 开发应用程序的方法, 后面的一些案例稍加修改即可在实际工作中使用, 下面对本书的内容做一个简单的概述。

案例一: 简单介绍 C# 的语法, 并使用一个案例演示使用 C# 开发简单应用程序的操作方法。

案例二: 使用 C# 开发一个简单的文本编辑器, 介绍 RichTextBox 控件的使用方法。

案例三: 开发一个 Web 浏览器, 介绍使用 COM 组件 AxWebBrowser 的方法。

案例四: 使用 GDI+ 开发一个绘图板程序, 介绍使用 GDI+ 开发图形程序的一般方法, 并介绍工厂模式。

案例五: 使用 C# 开发一个脍炙人口的游戏——俄罗斯方块, 介绍开发简单的基于窗体

的游戏的一般方法。

案例六：使用 C# 开发一个翻译助手，可以翻译复制到剪贴板中的文字，介绍剪贴板的操作方法，并介绍自定义词典文件的方法。

案例七：使用 Socket 编程实现一个网络聊天室，介绍使用 C# 进行 Socket 编程和实现网络聊天室的一般方法。

案例八：使用 C# 开发一个相册管理器，介绍使用 C# 进行图形图像处理 and 文件操作的方法。

案例九：使用 C# 开发简单的 POP3 邮件管理系统，可以支持扩展 IMAP 邮件系统。本案例从底层做起，从邮件服务器获取邮件数据，然后解析存储，介绍 POP3 协议、Windows 服务程序的开发和 Socket 编程方法。

本书结合案例制作，以学生课程设计和应用系统开发为背景，广泛适用于学生使用 C# 进行课程设计。本书所有案例的完整源代码可从中国水利水电出版社网站 (<http://www.waterpub.com.cn/softdown/>) 下载。

除封面署名作者外，参与本书写作、整理、调试、校对的还有董丹丹、余孝标、吴浩东、王晓青、林丽、童剑、杨昭昭、张瑞丰、李强、王景华、李蓉、李明、张海滨、王鹏、刘剑锋等。由于作者水平有限，书中难免有不足和错漏之处，恳请广大读者批评指正。我们的联系方法是：xinyuanxuan@263.net。

作者

2005 年 10 月

目 录

丛书序

前言

案例一 Hello World	1
1.1 程序设计目的	1
1.2 程序功能介绍	1
1.3 关键技术介绍	1
1.3.1 .NET 框架简介	1
1.3.2 C#语言语法简介	2
1.3.3 本案例中使用的技术简介	4
1.4 程序设计	4
1.4.1 设计界面	4
1.4.2 代码编写	5
1.4.3 程序使用	8
1.5 小结	8
案例二 文本编辑器	9
2.1 程序设计目的	9
2.2 程序功能介绍	9
2.3 架构设计与关键技术介绍	9
2.3.1 需求分析和架构设计	9
2.3.2 程序关键技术介绍	10
2.4 程序设计	11
2.4.1 设计界面	11
2.4.2 代码编写	13
2.4.3 程序使用	20
2.5 小结	20
案例三 Web 浏览器	21
3.1 程序设计目的	21
3.2 程序功能介绍	21
3.3 架构设计与关键技术介绍	21
3.3.1 需求分析和架构设计	21
3.3.2 程序关键技术介绍	22
3.4 程序设计	25

3.4.1	设计界面	25
3.4.2	代码编写	27
3.4.3	程序使用	34
3.5	小结	34
案例四	绘图板	35
4.1	程序设计目的	35
4.2	程序功能介绍	35
4.3	架构设计与关键技术介绍	35
4.3.1	需求分析和架构设计	35
4.3.2	程序关键技术介绍	38
4.4	程序设计	40
4.4.1	设计界面	40
4.4.2	代码编写	43
4.4.3	程序使用	62
4.5	小结	63
案例五	俄罗斯方块	64
5.1	程序设计目的	64
5.2	程序功能介绍	64
5.3	架构设计与关键技术介绍	64
5.3.1	需求分析	64
5.3.2	架构设计	65
5.3.3	程序关键技术介绍	68
5.4	程序设计	68
5.4.1	设计界面	68
5.4.2	代码编写	70
5.4.3	程序使用	87
5.5	小结	89
案例六	翻译小助手	90
6.1	程序设计目的	90
6.2	程序功能介绍	90
6.3	架构设计与关键技术介绍	90
6.3.1	需求分析	90
6.3.2	架构设计	91
6.3.3	程序关键技术介绍	93
6.4	程序设计	94
6.4.1	设计界面	94
6.4.2	代码编写	96

6.4.3	程序使用	113
6.5	小结	114
案例七	网络聊天室	115
7.1	程序设计目的	115
7.2	程序功能介绍	115
7.3	架构设计与关键技术介绍	115
7.3.1	需求分析	115
7.3.2	架构设计	116
7.3.3	程序关键技术介绍	118
7.4	程序设计	119
7.4.1	设计界面	119
7.4.2	代码编写	121
7.4.3	程序使用	137
7.5	小结	138
案例八	相册管理器	140
8.1	程序设计目的	140
8.2	程序功能介绍	140
8.3	架构设计与关键技术介绍	140
8.3.1	需求分析	140
8.3.2	架构设计	141
8.3.3	程序关键技术介绍	144
8.4	程序设计	144
8.4.1	设计界面	144
8.4.2	代码编写	147
8.4.3	程序使用	162
8.5	小结	167
案例九	邮件管理系统	168
9.1	程序设计目的	168
9.2	程序功能介绍	168
9.3	架构设计与关键技术介绍	168
9.3.1	需求分析和架构设计	168
9.3.2	程序关键技术介绍	171
9.4	程序设计	176
9.4.1	设计界面	176
9.4.2	代码编写	179
9.4.3	程序使用	205
9.5	小结	208

案例一 Hello World

1.1 程序设计目的

本案例的目的在于使 C#初学者了解 C#程序的结构并掌握编写 C#程序的一般方法，因此程序本身并不复杂，只有一些简单的功能，但是程序的覆盖面比较广，几乎涵盖了所有的常用操作，读者可以通过本案例简单了解 C#编程，如果读者已经对使用 C#编程非常熟悉，则可以跳过本案例。

1.2 程序功能介绍

本案例程序将实现下列功能：

- 可以实现文档的存储和读取。
- 可以抛出自定义异常。

1.3 关键技术介绍

使用.NET 编程前首先需要了解.NET 框架。

1.3.1 .NET 框架简介

2001 年，Microsoft 推出了全新的.NET 战略，希望使互联网以其产品和服务为中心发展，这意味着它能让个人计算机、移动电话、编写设备、数码相机甚至家用电器直接彼此交换信息，或者与互联网交换信息，让计算机进入完全的网络时代。此战略的推出开启了一个应用程序开发的新时代。

.NET 框架是用于建立 Internet 时代的应用程序的综合性平台，它允许开发人员建立使用 Windows 平台的完全富于新特色的应用程序。.NET 框架包含一个内容极其广泛的类框架，为程序开发人员提供了方便的服务。此外，.NET 框架为安全性、跨平台通信、资源管理和应用扩展性提供了完整的服务。

.NET 框架具有非常强大的跨语言特性，使用它所许可的所有语言开发出来的项目可以非常完善地集成到一起，因为事实上用所有语言开发的项目最终都将被编译为中间代码，官方称之为 IL (Intermediate Language)，此中间代码在运行时由 CLR (Common Language Runtime，公共语言运行时) 转换为机器代码执行，因此，所有的.NET 框架开发的项目（这里指托管项目）在运行时都需要 CLR 的支持，也就是说，只有在安装了.NET Framework 的机器上才能够执行，就像 Java 的项目需要运行在安装了 Java Runtime Environment 的机器上一样。目前.NET 所支持的语言已经有数十种，包括 C++、C#、VB、J#、JScript、Alice、APL、COBOL、Component

Pascal、Fortran、Smalltalk、ML、Python、Perl、RPG 等。这种跨语言的特性给编程人员带来的效率增加是非常显著的，因为不同语言在不同领域中都具有不可替代的作用，比如说对于数学或者金融应用来说，在表达相同意图的情况下，采用 APL 语法要比采用 Perl 语法节省大量时间。

1.3.2 C#语言语法简介

C#是伴随着.NET 一起推出的，它能够完全完美地符合.NET 框架。它由 C++语言发展而来，综合了 C++和 Java 语言的优点，功能强大、简洁稳定，因而成为大部分.NET 应用程序开发人员的首选语言。下面简单介绍一下它的基本语法。

- 数据类型。表 1-1 中给出了 C#主要的基本数据类型，基本类型的长度与计算机或者操作系统无关。

表 1-1 C#中的基本数据类型

类型	说明
bool	true 或 false
byte	8 位无符号整数
short	16 位整数
int	32 位整数
long	64 位整数
float	32 位浮点数
double	64 位浮点数
char	16 位字符
string	多个 16 位字符

可以直接将宽度较窄的数据类型赋值给宽度较宽的数据类型，并且可以自动转换成新类型（如将一个 int 型数据赋值给 long 型数据），不需要强制转换，系统会自动转换；而如果是相反的情况，则必须通过强制类型转换来完成。但是由于数据的范围不同，在转换过程中可能会发生数据丢失。

需要注意的是，和 C++不同，在 C#中，不可以将数值型数据赋值给布尔型变量，也不可以在布尔类型和其他数据类型之间转换。

可以使用 Convert 方法将数值转换成字符串或者将字符串转换为数值。

- 变量定义。在 C#中，变量名可以是任意长度的，可以由数字和大小写字母的混合形式组成，需要注意，和 C++一样，变量名是区分大小写的。另外，C#程序中的任何变量在使用前，必须进行声明。与 C 类似，在 C#中，可以在一条语句中将多个变量初始化为一个值。
- 运算符。C#中的基本运算符和逻辑运算符与其他大多数现代语言基本类似，表 1-2 给出了 C#中的算术运算符，表 1-3 给出了 C#中的逻辑运算符。

表 1-2 C#中的算术运算符

运算符	说明
+	加
-	减
*	乘
/	除
%	求余（整除后的余数）

表 1-3 C#中的逻辑运算符

运算符	说明
&	按位与
	按位或
^	按位异或
~	取反
>>n	右移 n 位
<<n	左移 n 位

- 同 Java 和 C++ 一样，在 C# 中也可以使用 ++ 和 -- 运算符表示整型变量的加 1 和减 1 操作。
- 比较运算符和布尔运算符，表 1-4 中列出了 C# 中的比较运算符和布尔运算符。

表 1-4 C#中的比较运算符和布尔运算符

运算符	说明
>	大于
<	小于
==	等于
!=	不等于
>=	大于等于
<=	小于等于
&&	逻辑与
	逻辑或
~	逻辑非

需要注意，等于运算符是 ==，而赋值运算符是 =，它们很容易被误用。

- 条件判断和循环。在 C# 中使用条件判断和循环和在 C++ 或者 Java 中大致相同。
- 类和对象。与 Java 类似，所有的 C# 程序都是由类组成的，C# 没有全局数据块或者共享数据的概念，任何变量都必须是某个类的一部分。
- 数组。在 C# 中，数组的大小可以是动态分配的，并且可以在任何时候重新分配空间，另外在 System.Collections 命名空间中还提供了很多可变长数组对象，包括 ArrayList、

Hashtable、SortedList 等。下面将会较为详细的介绍这些对象的用法。

- 异常。C#使用异常来完成错误处理，可以使用 try...catch 模块包含代码来捕获异常。
- 文件处理。C#中的文件处理对象提供了一系列相当灵活的处理文件的方法。在 System.IO 命名空间中，有大量的关于文件操作的类可供使用。平常主要使用的是 File 对象（此类提供了很多静态方法，可以在不创建案例的情况下直接使用它们）、FileInfo 对象（使用此类可以灵活地获得文件的信息）、FileStream 对象（表示一个文件的流）、StreamReader 对象（表示一个流读取对象，对流进行读取）、StreamWriter 对象（表示一个写入流对象，对流进行写入）等。

1.3.3 本案例中使用的技术简介

- ArrayList 类。ArrayList 类是容量可以动态增加的数组，它实现 IList 接口，提供了很多方便易用的方法。和数组一样，它可以使用下标来访问其中的元素，我们可以像使用数组一样使用它，这完全没有问题，但是会损失其灵活性。我们通常使用 Add 方法向其中添加元素，这种元素可以是任意类型，因为所有的类型都会默认继承 object 类，而 ArrayList 类中存储的正是 object 元素。在提取 ArrayList 中的元素时，需要使用强制类型转换将提取出来的元素转换为原来的类型。
- Hashtable 类。Hashtable 类和 ArrayList 一样，也是容量可以动态增加的数组，也实现了 IList 接口，其中存储的也是 object 元素，但是它更加灵活。它不使用数组下标来访问其中的元素，而是使用键/值对来访问，也可以这样认为，Hashtable 中的键其实就是它的下标，不过不是数字罢了。使用 Hashtable 时，如果访问其中的元素，需要使用 IDictionaryEnumerator 接口来访问，调用此接口的 MoveNext 方法来获取 Hashtable 的下一个元素。
- FileInfo 类。FileInfo 类提供创建、复制、删除、移动和打开文件的案例方法，在操作文件上具有相当大的灵活性。可以使用它来获取一个文件的详细信息或者创建一个文件的详细信息，然后再通过别的类（比如 FileStream 等）来对此文件进行访问。
- FileStream 类。FileStream 类公开以文件为主的 Stream，既支持同步读写操作，也支持异步读写操作。此类是操作文件时最常使用的类，它对输入/输出进行缓冲，从而提高性能。
- StreamReader 类和 StreamWriter 类。这两个类提供了对字节流按照字符来读取或者写入的方法，一般用来操作文本类型的文件。它们实现了 TextReader 和 TextWriter，可以使用特定的编码对字节流进行字符的读取和写入。

1.4 程序设计

1.4.1 设计界面

本程序只包含了一个窗体，界面设计如下：在程序中新建窗体，命名为 HelloWorld，在窗体中添加一个 TextBox 控件和三个 Button 控件，分别命名并设置它们的属性。设置完毕后界面如图 1-1 所示。表 1-5 是窗体中的控件属性列表。

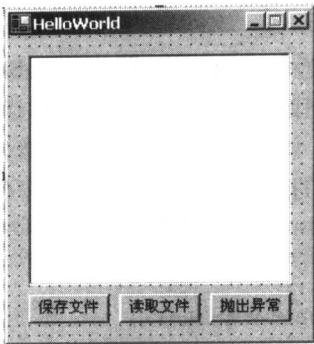


图 1-1 主窗体界面图

表 1-5 控件对象属性列表

对象	属性	设置值	控件功能
TextBox	Name	txtContent	容纳一个字符串
Button	Name Text	btnSaveFile 保存文件	保存文件
Button	Name Text	btnLoadFile 读取文件	读取文件
Button	Name Text	btnException 抛出异常	抛出一个异常

1.4.2 代码编写

1. 变量说明窗体加载方法

程序中使用到的变量如下：

```
private string FileName = "a.txt"; // 文件名
private string FileContent;        // 文件内容
```

窗体加载方法实现读取文件并设置文本框中的文本为文件内容，如果文件不存在则提示，然后创建文件，代码如下：

```
private void HelloWorld_Load(object sender, System.EventArgs e)
{
    FileInfo info = new FileInfo(FileName);
    if (!info.Exists)
    {
        MessageBox.Show("File Doesn't Exist! Will Create!");
    }
    FileStream fs = info.Open(FileMode.OpenOrCreate,
        FileAccess.ReadWrite, FileShare.ReadWrite);
    StreamReader sr = new StreamReader(fs);
    string strContent = sr.ReadLine();
    sr.Close();
    fs.Close();
    this.txtContent.Text = strContent;
    this.FileContent = strContent;
}
```

2. 鼠标响应方法

在鼠标响应中，演示了如何使用 `ArrayList` 和 `Hashtable`。在鼠标左键的响应中，使用了 `ArrayList` 来存储分割成字符串数组的文件内容，并使用对话框显示 `ArrayList` 中存储的所有元素的内容。在鼠标右键的响应中，使用了 `Hashtable` 来存储分割成字符串数组的文件内容，并使用对话框显示 `Hashtable` 中存储的所有元素的内容。程序代码如下：

```
private void HelloWorld_MouseDown(object sender,
    System.Windows.Forms.MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        ArrayList listContent = new ArrayList();
        // 将 FileContent 按照空格或者制表符分割成字符串数组
        string[] strContent = FileContent.Split(null);
        foreach (string str in strContent)
        {
            if (str != "")
            {
                listContent.Add(str);
            }
        }
        string strDisplay = "";
        foreach (object obj in listContent)
        {
            string str = obj as string;
            if (str != null)
            {
                strDisplay += str + System.Environment.NewLine;
            }
        }
        MessageBox.Show(strDisplay);
    }
    // 右键
    else
    {
        Hashtable ht = new Hashtable();
        // 将 FileContent 按照空格或者制表符分割成字符串数组
        string[] strContent = FileContent.Split(null);
        int number = 0;
        foreach (string str in strContent)
        {
            if (str != "")
            {
                ht.Add("Number"+number.ToString(), str);
            }
            number++;
        }
        string strDisplay = "";
        IDictionaryEnumerator enumerator = ht.GetEnumerator();
        while (enumerator.MoveNext()) {
```

```
        strDisplay += enumer.Key.ToString() +  
            ":" + enumer.Value.ToString() +  
            System.Environment.NewLine;  
    }  
    MessageBox.Show(strDisplay);  
}  
}
```

3. “保存文件”按钮响应方法

“保存文件”按钮实现将文本框中的内容保存到文件，如果文件不存在，则提示然后创建文件，代码如下：

```
private void btnSaveFile_Click(object sender, System.EventArgs e)  
{  
    FileInfo info = new FileInfo(FileName);  
    if (!info.Exists)  
    {  
        MessageBox.Show("File Doesn't Exist! Will Create!");  
    }  
    FileStream fs = info.Open(FileMode.OpenOrCreate,  
        FileAccess.ReadWrite, FileShare.ReadWrite);  
    StreamWriter sw = new StreamWriter(fs);  
    sw.WriteLine(FileContent);  
    sw.Close();  
    fs.Close();  
}
```

4. “读取文件”按钮响应方法

“读取文件”按钮实现将读取的文件内容存储到 FileContent 变量中，然后设置文本框的内容为文件的内容，如果文件不存在，则提示然后创建文件，代码如下：

```
private void btnLoadFile_Click(object sender, System.EventArgs e)  
{  
    FileInfo info = new FileInfo(FileName);  
    if (!info.Exists)  
    {  
        MessageBox.Show("File Doesn't Exist! Will Create!");  
    }  
    FileStream fs = info.Open(FileMode.OpenOrCreate,  
        FileAccess.ReadWrite, FileShare.ReadWrite);  
    StreamReader sr = new StreamReader(fs);  
    string strContent = sr.ReadLine();  
    sr.Close();  
    fs.Close();  
    MessageBox.Show("File Content is: " + strContent);  
}
```

5. “抛出异常”按钮响应方法

“抛出异常”按钮实现抛出一个自定义的异常，代码如下：

```
private void btnException_Click(object sender, System.EventArgs e)  
{  
    throw new Exception("I throw an Exception ,  
        but i don't know what's to do ,hehe!");  
}
```