

C# 3.0 实例精通

Mastering C# 3.0 Step by Step

王院峰 陈静 等编著

本书特色

- ◎ 涵盖基础知识、核心技术、典型实例等内容
- ◎ 按照“技术要点→实现代码→源程序解读”的模式讲解，容易上手
- ◎ 提供**170**余个典型实例进行讲解

超值光盘内容

- ◎ 本书源代码
- ◎ **1200**余页编程技术文档（免费赠送）
- ◎ **45**个编程专题视频讲座（免费赠送）



编程实例大讲堂

C# 3.0 实例精通

Mastering C# 3.0 Step by Step

王院峰 陈静 等编著

图

0.240

(

ISBN

1-1

中国

机械

王院峰 陈静 等编著

北京航空航天大学出版社

2009年3月第1次印刷

184mm x 260mm · 37.52印张

ISBN 978-7-111-25278-1

(ISBN 978-7-89483-240-2 光盘)

定价: 69.80元(附光盘)

机械工业出版社 China Machine Press

本书介绍 .NET 的新版本 .NET Framework 3.5 以及新的 IDE Visual Studio 2008。

本书中所有范例创建，都是基于 Microsoft 刚刚发布的 Visual Studio 2008 所进行的，范例的内容不但涉及了 C# 3.0 的各个方面，而且还对 .NET Framework 3.5 中所增加和增强的一些功能进行了详细的实例代码演示和讲解，如 LINQ、Windows Presentation Foundation、Windows Communication Foundation 和 Windows Workflow Foundation 等。

对于本书，初学者可通过所有实例的学习，掌握从 Visual C# 2005 到 C# 3.0 中的所有知识；对于学习过 C# 语言的读者，可选择本书中涉及新知识点的实例进行学习。本书适合 C# 3.0 初中级用户阅读，通过学习，能将本书中的实例应用于实际项目开发中。本书也可作为各大专院校的实训手册。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

C# 3.0 实例精通/王院峰，陈静等编著. —北京：机械工业出版社，2009.3
(编程实例大讲堂)

ISBN 978-7-111-25978-7

I. C… II. ①王… ②陈… III. C#语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 212434 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑：李华君

北京瑞德印刷有限公司印刷

2009 年 3 月第 1 版第 1 次印刷

184mm × 260mm · 37.75 印张

标准书号：ISBN 978-7-111-25978-7

ISBN 978-7-89482-940-5(光盘)

定价：69.80 元(附光盘)

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：(010)68326294

丛书序

不积跬步，无以至千里。

——荀子

初学编程的人很苦恼的一件事是不知道如何上手。其实有两种思路都可以很好地上手。一种是按部就班，像大学里的 C 程序设计课程一样，从基本语法，到各种具体应用程序逐渐深入；还有一种方法是，把基本语法和一些函数等的用法用比较典型的实例贯穿起来，通过学习这些实例来掌握编程知识，这也是一种很好的方法。因为学习编程需要大量的实践才能学好，而这种方法正好符合这种学习特点，所以也有比较好的效果。按照这个思路我们策划了这套书，可以让初学编程的人员快速入门与提高。

丛书特色

作为一套以实例贯穿始终的图书，本丛书在编写上着重体现以下特色。

1. 以实例引导学习，可快速入门

本丛书以全新的实例模式编写，每本书都是以实例贯穿始终，读者可以在实例引导下一步一步地学习编程，增强了编程的亲身体验，可以快速入门，达到良好的学习效果。

2. 编写模式科学，讲解细致

本丛书中贯穿的实例大都是按照“技术要点→实现代码→源程序解读”的模式编写，非常科学，讲解也很细致，容易掌握。

3. 实例数量丰富，实践性强

本丛书每本书都是以数以百计的实例指导读者学习，这些例子实用强，可为读者以后程序开发奠定坚实的基础。

4. 代码规范，注释丰富

为了增强代码的易读性，丛书编写时对代码进行了丰富的注释，非常易于读者阅读和理解，增强学习效果。

5. 光盘内容实用、超值

配书光盘提供了书中所涉及的源代码，以方便读者使用。除此之外，还特别免费提供了大量的编程入门视频和技术文档，以方便相关人员学习和教学使用。

6. 提供技术支持

本丛书提供了论坛：<http://www.rzchina.net>，读者可以在上面提问交流。另外，论坛上还有一些小的教程、视频动画和各种技术文章，可帮助读者提高开发水平。

丛书包含的书目

- 《Java 实例精通》
- 《Java Web 整合开发实例精通——Struts + Hibernate + Spring》
- 《Visual C++ 实例精通》
- 《Visual Basic 实例精通》
- 《C# 3.0 实例精通》
- 《ASP.NET 3.5 实例精通》
- 《PHP 实例精通》
- 《JavaScript 实例精通》

本丛书读者定位

- 初学编程的人员；
- 已经入门，需要通过实例提高编程水平的人员；
- 大中专院校的学生；
- 社会培训学员；
- 相关程序员。

阅读本丛书的几点建议

- 没有基础的读者建议按顺序阅读，不要跳跃，不要跳步。
- 有基础的读者可以跳过一些特别基础的章节学习。
- 如果感觉学习本书有困难，建议先阅读机械工业出版社“编程红宝书”丛书中的对应入门图书。
- 多动手，亲自完成书中的实例，加深理解。
- 遇到问题，除了本书的技术支持论坛，还可利用网络资源解决。例如，利用 Google 和 Baidu 搜索相关资料，或者在相关论坛上发帖提问，会有热心人给你答复。
- 要重点阅读源代码及其注释，可以有效提高代码理解能力。

正所谓“宝剑锋从磨砺出，梅花香自苦寒来”。编写这样的一套书也实属不易，是一个需要克服很多困难、花费大量心血才能完成的“浩大工程”。同样，在学习编程的道路上也不会一帆风顺，肯定有许多磨难等着你。我们伟大的思想家荀子早都说过，“不积跬步，无以至千里”。做任何事都得脚踏实地，才能走得远，希望以此与各位读者共勉。看到你们能以此套书提升编程水平，便是我们最开心的事了！

丛书策划编辑

前 言

软件业在飞速地发展，转眼间微软已经发布了 .NET Framework 的新版本 3.5，Visual Studio 2008 则对以前版本的许多功能进行了改进，增加了许多令人兴奋的新功能，使得应用程序的创建更为简单和规范。

作为一个程序开发人员，及时了解和掌握最新技术，将有助于提高自身的价值和竞争力。对于一个初学者来说，实例分析能直观地对知识点进行剖析，更能使读者明白该知识点的使用；对于一个有过编程和项目开发经验的程序员来说，接触最新的技术知识也许是比较渴望的，而且参考实例讲解进行学习，可更快速地掌握相关知识点；对于一个正在进行项目开发的程序员来说，某些实例可直接应用到项目中，免去了先查找资料后编写代码实现某个功能的麻烦。

目前，市面上介绍 C# 3.0 的书籍较少，鉴于此，编者结合自己的项目开发经验，并查阅 .NET Framework 3.5 的官方技术文档，编写了这本实例讲解的书籍。

本书特色

相比同类图书，本书具有以下明显特色。

1. 实例写作，学习直观

本书采用实例写作的模式，将技术的实现过程直接展现在读者面前，使读者的学习更为直观。对于所有实例，都采用代码、图示相结合的方式，做到了明确直观。

2. 知识最新，兼容性强

本书中，各实例都采用 Visual Studio 2008 进行创建，覆盖了 .NET 3.5 的各个方面，包含了几乎所有 C# 3.0 的知识，帮助读者全面地学习新技术。

3. 实例丰富，强调实践

在进行实例设计时注重实际应用，不会让读者读完了还不知道该怎么去使用。本书中的很多例子都来源于作者的实际开发，通过对这些例子的学习，可以增强读者的动手能力。

4. 代码规范，注释丰富

本书所涉及的程序源代码层次清楚，语句简洁，注释丰富，体现了代码优美的原则，这样从一开始便给读者树立了良好的榜样，有利于读者养成良好的代码编写习惯。

5. 讲解循序渐进，章节跨度不大

读者可以根据自己的实际基础，选择适合自己的章节开始学习，而不会觉得跨度太大。

本书内容

本书采用实例讲解的写作模式，共分 4 部分进行介绍。

第一部分(第1章~第3章)讲述了C# 3.0基础语言知识,从基本的变量、语句、表达式、数据类型到结构化编程,对于每个知识点都进行了范例演示。本部分还讲述了开发环境的安装和配置,并对C#的面向对象编程进行了实例演示,为读者掌握后面的知识点打下基础。

第二部分(第4章~第5章)讲述了C# 3.0高级语言知识,包括程序的异常处理、值类型和引用类型的理解、数组和集合、属性、委托与事件、泛型、垃圾回收等。学习并理解本章范例内容后,可为编写复杂应用程序打下坚实的基础。

第三部分(第6章~第9章)介绍了Windows与Web应用程序的创建。第6、7章演示了Windows应用程序中控件和组件的使用,以及在程序中进行常用的数据处理的知识;第8、9章演示了Web应用程序的构建方法,包括常用控件的使用,以及数据控件和ADO.NET的数据处理技巧。

第四部分(第10章~第16章)介绍了Visual C# 2008的平台、组件与框架,包括ASP.NET AJAX站点的构建、语言集成查询(LINQ)的详细范例、应用程序中水晶报表的创建和使用。第14、15章和16章这3章,使用丰富而又具有代表性的实例,来演示了.NET Framework 3.5中所增加和增强的3个功能,分别为Windows Presentation Foundation、Windows Communication Foundation和Windows Workflow Foundation。

本书适合的读者

- 从未接触过C#编程的自学人员;
- 了解一些Visual C# 2005知识,但还需要进一步学习新知识的读者;
- .NET平台下的开发人员或者正在进行相关项目开发的程序员;
- 各大中专院校的在校学生和相关授课老师;
- 其他编程爱好者。

本书的作者

本书主要由王院峰、陈静编著。其中第1~5章主要由陈静编写,第6~16章主要由王院峰编写。其他参与编著和资料整理的人员有冯华君、刘博、刘燕、叶青、张军、张立娟、张艺、彭涛、徐磊、戎伟、朱毅、李佳、李玉涵、杨利润、杨春娇、武鹏、潘中强、王丹、王宁、王西莉、石淑珍、程彩红、邵毅、郑丹丹、郑海平、顾旭光。

编 者

目 录

丛书序

前言

第 1 章 C# 3.0 入门	1
1.1 Visual Studio 2008 的安装与部署	1
1.1.1 Visual Studio 2008 IDE 的安装	1
首要的工作就是将 Visual Studio 2008 安装到计算机中。本节介绍 Visual Studio 2008 Beta2 版本的安装过程。	
1.1.2 安装 MSDN	2
MSDN Library 的使用是免费的，微软提供了在线查阅的 MSDN Library，为了方便日常使用，可将其安装到硬盘中。	
1.2 初识 Visual Studio 2008	3
1.2.1 Visual Studio 2008 的新特性	3
作为微软推出的新一代集成开发工具 Visual Studio 2008，不但弥补了较早版本的一些不足，功能上也进行了相关的新增和增强。	
1.2.2 Visual Studio 2008 开发环境	7
Visual Studio 2008 由以下部分组成：菜单工具栏、标准工具栏、停靠或自动隐藏在左侧、右侧、底部以及编辑器空间的各种工具窗口。本节介绍 Visual Studio 2008 开发环境。	
1.2.3 Visual Studio 2008 IDE 基本功能设置	12
Visual Studio 2008 IDE 中，提供了许多选项设置和个性化的设置功能，本节将进行介绍。	
1.3 使用 Visual Studio 2008 开发第一个 C# 程序	15
1.3.1 第一个 C# 控制台应用程序	15
介绍了那么多的新特性，本节将创建一个 C# 控制台应用程序。	
1.3.2 第一个 Windows 窗体应用程序	17
本节将创建第一个 Windows 窗体应用程序。	
第 2 章 C# 语言基础	19
2.1 使用变量与常量	19
2.1.1 理解基本语句	19
语句是组成程序的基本单位，输入输出语句、注释语句等。	
2.1.2 常量的使用	21
任何程序的编写都离不开常量和变量的使用，学习一门程序语言，首先得了解它的	

	常量和变量。本节将演示在 C# 中常量的使用。	
2.1.3	变量的使用 任何程序语言都有它自己的常量和变量命名规则。本节将演示在 C# 语言中如何使用变量。	23
2.1.4	使用基本的数据类型 C# 是强类型语言；因此每个变量和对象都必须具有声明类型。本节将演示基本的数据类型的使用。	25
2.1.5	转换数据类型 就算把同一系列的数据从一种类型的变量移动到另一种类型的变量中，结果也会有差别，与所期望的大不相同，这就需要对数据进行类型转换。	28
2.2	使用运算符与表达式	30
2.2.1	使用赋值与算术运算符 运算符的范围非常广泛，有简单的，也有非常复杂的，其中有一些只能在数值应用程序中使用，这些运算符被称为算术运算符。	30
2.2.2	使用关系与逻辑运算符 用关系运算符把两个表达式连接起来的式子，称为关系表达式。关系表达式的返回值是布尔型，即“真”或“假”。	32
2.3	使用选择结构	35
2.3.1	使用 if 结构 在大多数要进行条件判断等场合，将会使用 if 语句，if 语句还可进行嵌套，用于比较复杂的多重条件判断等场合，本节将对 if 结构进行演示。	35
2.3.2	使用 switch 结构 switch 语句是多分支选择语句，用于将一个表达式的值同许多其他值比较，并按比较结果选择下面该执行哪些语句。	40
2.4	使用循环结构	42
2.4.1	使用 while 循环 while 语句是循环语句，同时也属于条件判断语句，它可根据某个条件，来对 while 语句内的代码进行循环执行。	42
2.4.2	使用 do...while 循环 do...while 循环可实现与 while 循环几乎相同的功能，本节将对此进行演示。	45
2.4.3	使用 for 循环 for 循环是 C# 中使用频率比较高的循环语句，但同时也是循环类语句中最复杂的，本节将演示如何使用 for 循环。	47
2.4.4	综合实例——输出一个菱形 将菱形分解成为上下两部分，上边的是一个正立的正三角形，下边是一个倒立的正三角形，下边的正三角形边长比上边的三角形边长小 1。	49
2.5	C# 基础知识综合实例——实现一个简易计算器 主要将本章所介绍的 C# 基础知识点组合起来，实现一个简易的计算器。	53
第 3 章	C# 面向对象编程	58
3.1	使用 C# 创建类与对象	58
3.1.1	创建并使用类 简单地说，类是一种抽象的数据类型，但是其抽象的程度可能不同。而对象就是一个类的实例。	58

3.1.2	使用类的属性	60
	属性是对现实中实体特征的抽象，它提供了对类或对象性质的访问。本节将演示如何定义和使用类的属性。	
3.1.3	使用类的字段	62
	字段是类或结构中的对象或值，类和结构使用字段来进行数据的封装。	
3.1.4	使用类的方法	65
	任何事物必须封装在类中，或者作为类的成员。本节将演示如何使用类的方法。	
3.1.5	使用静态类	68
	静态类和类成员用于创建无需创建类的实例就能够访问的数据和函数。本节将演示如何创建和使用建静态类。	
3.1.6	使用部分类	71
	当所要创建的类包含一种类型的多个成员时，代码文件就会很长，稍不仔细就很容易混淆，这就可以给代码分组，即通过创建部分类来实现。	
3.2	结构与枚举	75
3.2.1	使用结构构造数据类型	75
	当需要将不同类型的简单数据组合成一个有机的整体时，使用结构体无疑是一个聪明的选择。	
3.2.2	使用枚举	77
	枚举是一个指定的常数，其基础类型可以是除 char 外的任何整型。本节将对这种常用而又简单的变量类型——枚举进行演示。	
3.3	使用继承	79
3.3.1	类的继承	79
	继承是可以使用以前所创建的类的方法和属性，通过简单的程序代码来建造功能强大的类。	
3.3.2	类的多态性	83
	多态性是一个非常重要的概念。它允许客户对一个对象进行操作，由对象来完成一系列的动作，具体实现哪个动作、如何实现则由系统解释。	
3.3.3	使用构造函数	86
	构造函数是在创建给定类型的对象时执行的类方法。构造函数具有与类相同的名称，它通常对新对象的数据成员进行初始化。	
3.3.4	使用析构函数	90
	在类的实例超出范围时，要确保其所占的资源能被回收，于是就出现了析构函数。	
3.3.5	使用虚方法	92
	当类中的方法声明前加上了 virtual 修饰符，该方法就被称为虚方法，虚方法可以被重载。	
3.3.6	理解并使用抽象类	94
	抽象类可提供多个派生类可共享的基类的公共定义，但抽象类却不能实例化。	
3.4	创建接口	97
3.4.1	创建并使用接口	97
	接口只是作为类或结构的管理，主要是对类或结构的构成进行限制，而具体的实现，还是由类或结构来做的。	
3.4.2	使用多个接口	101
	在实际需要的时候，类可以实现多个接口，来满足需要，本节将对此进行演示。	

第 4 章 C#高级语言知识(一)	104
4.1 处理错误和异常	104
4.1.1 简单的异常处理	104
异常处理是捕获预期的和突发事件的技术,其目的是在错误发生之前,能够事先预料错误,使得程序更可靠。	
4.1.2 使用多个 catch 语句处理程序	107
上一节谈到了在 C#中处理错误和异常的方法,并说明了不同的异常类用来捕捉不同的错误和异常。可是对于在 try 块中如果可能引发多种类型的异常这种写法就很难满足要求。	
4.1.3 使用 checked 和 unchecked	110
在 C#语言中,C#语句既可以在已检查的上下文中执行,也可以在未检查的上下文中执行。在已检查的上下文中,算法的溢出将引发异常。	
4.2 理解值和引用	113
4.2.1 认识 System.Object 类	113
System.Object 类是 .NET Framework 中所有类的最终基类;它是类型层次结构中的根。它支持 .NET Framework 类层次结构中的所有类,并为派生类提供低级别的服务。	
4.2.2 复制类	116
在实际应用中,可能会碰到需要创建一个对象的副本进行其他使用,这就用到复制类的相关知识。	
4.2.3 创建 ref 和 out 关键字	117
使用 ref 关键字,可以使参数按引用传递。当控制权传递回调用方法时,在方法中对参数的任何更改都将反映在该变量中。	
4.2.4 使用委托	119
所有委托都是继承自 System.Delegate 类,并且有一个调用列表,调用委托时所执行的方法都被存放在这样一个链接列表中。	
4.2.5 理解装箱和拆箱	121
装箱和拆箱使得在 C#类型系统中,任何值类型、引用类型和对对象类之间进行转换。或者说,有了装箱和拆箱,对任何类型的值来说,最终都可以将其看作是对象类型。	
4.3 使用数组和集合	123
4.3.1 创建和使用数组	123
数组是一种数据结构,它是由类型声明的,包含了若干相同类型的变量。	
4.3.2 将数组作为参数传递	127
因为数组是引用类型,所以数组可作为参数传递给方法。本节将对此进行演示。	
4.3.3 数组的复制	128
由于数组是引用类型,在创建其副本的过程中,很容易就会将复制变成了创建新的引用,本节将演示几种如何对数组进行复制的方法。	
4.3.4 定义使用交错数组	130
交错数组是元素为数组的数组,它的元素的维度和大小可以不同。	
4.3.5 创建并使用隐式类型的数组	132
C#语言中,可以创建隐式类型的数组,在这样的数组中,数组实例的类型是从数组初始值设定项中指定的元素推断而来的。	
4.3.6 使用 ref 和 out 传递数组	134
可以使用 ref 和 out 向传递其他参数一样来将数组作为参数进行传递,本节范例将对	

	此进行演示。	
4.3.7	使用 ArrayList 类 System. Collections 命名空间提供了一个名为 ArrayList 的数组类, 它实现了 IList 接口, 使得其使用大小可按需动态增加。	137
4.3.8	使用 Queue 类 System. Collections 命名空间提供了一个名为 Queue 的集合类, 随着向 Queue 中添加元素, 它的容量通过重新分配按需自动增加。	139
4.3.9	使用 Stack 类 System. Collections 命名空间提供了一个名为 Stack 的集合类, 随着向 Stack 中添加元素, 它的容量通过重新分配按需自动增加。	141
4.3.10	使用 Hashtable Hashtable 通常被称为哈希表, 它表示键(key)/值(value)对的集合。	142
4.4	使用属性与索引器 146	
4.4.1	使用属性 属性使类能够以一种公开的方法获取和设置值, 同时隐藏了实现这种功能的具体代码。	146
4.4.2	实现接口属性 可以在接口上声明属性, 本节将对此进行演示。	149
4.4.3	使用索引器 索引器类似于属性, 不同之处在于它们的访问器采用参数。	151
4.4.4	实现接口中的索引器 索引器还可在接口上进行声明, 索引器中的访问器的用途是指示该索引器是读写、只读还是只写。	154
第5章	C#高级语言知识(二) 158	
5.1	委托与事件 158	
5.1.1	使用委托 介绍一个引用类型——委托(delegate), 实际上相当于函数指针原形。	158
5.1.2	委托的合并 委托的合并也被称为组合多路广播委托, 本节将对此进行演示。	160
5.1.3	使用匿名方法 要将代码块传递为委托参数, 创建匿名方法则是唯一的方法。	162
5.1.4	事件的订阅和发布 类或对象可以通过事件向其他类或对象通知所发生的相关事情。	164
5.2	泛型 167	
5.2.1	理解泛型 泛型类和泛型方法同时具备可重用性、类型安全和效率, 这是非泛型类和非泛型方法无法具备的。	167
5.2.2	泛型和数组 在 C# 3.0 中, 下限为零的一维数组回自动实现 IList<T> 接口。这使得可以创建能够使用相同代码循环访问数组和其他集合类型的泛型方法。	171
5.2.3	泛型接口的使用 为泛型集合类或表示集合中项的泛型类定义接口, 通常会为其他相关工作带来方便。	173



对于泛型类来说，使用泛型接口将十分有用。

5.3 迭代器	179
5.3.1 为整数列表创建迭代器	179
可以为类创建一个迭代器，该迭代器将按相反顺序返回元素，或在迭代器返回元素之前对每个元素执行操作。	
5.3.2 为泛型列表创建迭代器	181
可以为泛型列表创建一个迭代器，该迭代器将按相反顺序返回元素，或在迭代器返回元素之前对每个元素执行操作，在创建时不必实现整个 IEnumerator 接口。	
5.4 使用垃圾回收	184
5.4.1 使用 GC 类	184
GC 类提供着一种自动回收未使用内存的服务。它控制着系统垃圾回收器。	
5.4.2 实现 Dispose() 方法	187
对象类型的 Dispose() 方法，可以释放该对象所拥有的所有资源。它还通过调用其父类型的 Dispose() 方法，释放其基类型拥有的所有资源。	
第 6 章 创建 Windows 应用程序(一)	191
6.1 使用 C# 3.0 公共控件	191
6.1.1 实现系统托盘程序	191
使用 Windows 窗体中的 NotifyIcon 组件，可以将后台运行且没有用户界面的进程显示图标，在任务栏的状态通知区域中进行显示。	
6.1.2 实现简单的计算器	197
使用 Windows 窗体 Button 控件和 TextBox 控件可实现一个简单的计算器，本节将对此进行演示。	
6.1.3 ComboBox 实现列表项的自动选择	201
Windows 窗体的 ComboBox 控件可在下拉组合框中显示数据。并在用户进行输入时，根据用户当前的输入内容，自动匹配列表中的有关项。	
6.1.4 使用 RadioButton 控件和 CheckBox 控件	203
CheckBox 控件指示某特定条件是打开的还是关闭的。而 RadioButton 控件为用户提供由两个或多个互斥选项组成的选项集，在同一时间只能选择一个。	
6.1.5 HScrollBar 控件实现颜色调节器	206
HScrollBar 控件用于在应用程序中实现一个水平滚动的滚动条，以方便在较长的项列表或大量信息中转移。	
6.1.6 实现一个简单的写字板	209
窗体中的 RichTextBox 控件可用于显示、输入和操作格式化的文本。	
6.1.7 TreeView 实现磁盘目录列表	217
使用 TreeView 控件，可以显示出节点的层次结构，就像 Windows 资源管理器功能的左窗格中显示的文件夹一样。	
6.1.8 ListView 实现简单资源浏览器	221
ListView 控件可以显示带图标的项的列表，使用 ListView 控件可以创建类似于 Windows 资源管理器右边窗格的用户界面。	
6.1.9 使用 WebBrowser 制作简易浏览器	227
WebBrowser 控件将 WebBrowser ActiveX 控件进行了托管包装，使得现在直接使用该控件可以很方便的在应用程序中进行网页浏览的功能。	
6.1.10 使用掩码进行输入验证	236

掩码文本框控件是一个增强型的文本框控件，它支持用于接受或拒绝用户输入的声明性语法，这种语法被称做掩码。	
6.1.11 使用 TrackBar 控件	240
TrackBar 控件用于在大量信息中进行浏览，或者以可视的形式来进行数字设置的调整。	
6.2 使用组件	242
6.2.1 使用 DirectoryEntry 组件创建虚拟目录	242
DirectoryEntry 组件可封装 Active Directory 域服务层次结构中的节点或对象，这些包括创建、删除、重命名、移动子节点和枚举子级。	
6.2.2 使用 ErrorProvider 组件进行输入验证	246
使用 ErrorProvider 组件，可以对窗体或控件上的用户输入内容进行验证。本节将对此进行演示。	
6.2.3 使用 EventLog 组件操作 Windows 事件日志	249
EventLog 组件提供了与 Windows 事件日志进行交互的方法和属性，本节范例将使用 EventLog 组件操作 Windows 事件日志和自定义日志。	
6.2.4 使用 FileSystemWatcher 组件创建文件监视器	254
FileSystemWatcher 组件可以侦听文件系统更改通知，并在目录或目录中的文件发生更改时引发事件。	
6.2.5 使用 ServiceController 组件管理 IIS 服务	257
ServiceController 组件可进行 Windows 服务的管理，并允许连接到正在运行或者已停止的服务、对其进行操作或进行有关该服务信息的获取。	
6.2.6 使用 Process 组件获取进程信息	260
Process 组件提供了对本地和远程进程的访问功能，并能够对本地系统进程进行启动和停止。	
第 7 章 创建 Windows 应用程序(二)	263
7.1 连接 Access 数据库	263
可以利用 ADO.NET 所提供的内置对象，只需要编写少量代码，就可以与 Access 数据库建立数据连接并进行数据的获取。	
7.2 连接 SQL Server 数据库	267
使用 SqlConnection 对象可以建立一个与 SQL Server 数据库的连接，可通过它的 State 属性得到当前连接的状态。	
7.3 数据的插入	270
可以利用 ADO.NET 所提供的内置对象，只需要编写少量代码，就可以建立与 SQL Server 数据库的数据连接，并进行数据的插入操作。	
7.4 数据的修改	274
通过使用 ADO.NET 所提供的内置对象，只需要编写少量代码，就可以建立与 SQL Server 数据库的数据连接，并进行数据的修改操作。	
7.5 数据的删除	279
本节程序中，将演示在只定义一个 SqlDataAdapter 对象的情况下，更换查询语句，来对同一 DataSet 进行填充。	
7.6 数据库的分离和附加	283
可通过调用数据库系统中的存储过程，来实现常规的数据库维护工作。	

第 8 章 构建 Web 应用程序(一)	289
8.1 使用 ASP.NET 基本控件	289
8.1.1 实现多功能日历	289
网页上面要是多一个交互式的日历,那么将会是一个吸引用户的亮点。使用 Calendar 控件,可以轻松地创建自己的多功能日历。	
8.1.2 使用容器控件分组管理其他控件	292
使用容器控件 Panel,可以对放置于其中的一组控件,进行统一管理,如显示、隐藏等。本节将对此进行演示。	
8.1.3 实现页面项目的列表	294
使用 BulletedList 服务器控件,可以创建有序或无序(编号的)的项列表,本节将对此进行演示。	
8.1.4 构建多视图界面	297
在 Web 页面中使用多视图界面技术,在需要的时候显示当前所需要的内容,浏览器更能感觉到网站的人性化。	
8.1.5 Wizard 控件创建分布式界面	300
借助于 Wizard 控件,可以很方便地实现分布式的界面,将所有条目分布显示,也会给用户一种友好的感觉。	
8.1.6 Substitution 控件向缓存页中插入动态内容	304
当一个页面的所有内容都被缓存时,如果需要页面中某些内容还是动态更新的话,那么可使用 Substitution 控件。	
8.2 使用检验控件	306
8.2.1 使用 RequiredFieldValidator 控件进行必填项验证	306
可以使用必填字段验证控件 RequiredFieldValidator,来检查该表单项目是否有输入值。	
8.2.2 使用 RangeValidator 验证数据的范围	308
RangeValidator 控件可以检查一个表单项的值是否位于最大值和最小值之间。这个最大值和最小值可以是日期、数值、货币或字符串。	
8.2.3 CustomValidator 控件实现自定义验证	310
通过使用 CustomValidator 控件,可以让用户自己编写检验函数,执行自定义的检验。	
8.2.4 使用正则表达式执行验证	313
实际的项目开发过程中,有的表单项是特定的,所填内容必须符号某种规则。使用 RegularExpressionValidator 来进行检验,该控件可以读取表单项中输入的值,并判断是否和设置的正则表达式想匹配。	
8.3 使用 Web 部件	315
8.3.1 实现可自由拖动的 Web 部件	315
在常见的个人博客中,页面上的多个模块可以随用户的喜好自由拖动放置,可以更加直观的自由定制,更能吸引用户。本节将演示在 ASP.NET 中,如何实现可自由拖动的 Web 模块。	
8.3.2 实现可自由添加的 Web 部件	318
比较常见的 Web 系统如博客等,当用户成功注册后,需要先对页面进行设定,这时候,会有一系列的模块需要选择添加,在 ASP.NET 中,可借助于 CatalogZone 控件来实现该功能。	
8.3.3 Web 部件的个性设置	321
光是可以自由拖动应用模块的位置和对模块进行添加,还不能达到完全定制的目的,	

本节将演示如何对添加到页面中的模块，进行外观、布局、行为和其他属性的编辑。	
8.3.4 Web 部件的导入导出	325
本节的范例代码中，首先创建了一个 XML 树，并保存到 XML 文件中，然后进行查询操作。	
第 9 章 构建 Web 应用程序(二)	329
9.1 使用数据源控件获取数据	329
9.1.1 AccessDataSource 控件获取 Access 数据	329
在 Web 页面中，可以通过一个 AccessDataSource 数据源控件，来完成 Access 数据的获取工作。	
9.1.2 XmlDataSource 控件获取 XML 数据	332
使用 XmlDataSource 控件，可以将 XML 数据绑定到数据绑定控件。本节实例将展示这种实现方法。	
9.1.3 SqlDataSource 控件获取 SQL Server 数据	334
在 Web 页面中，可以通过一个 SqlDataSource 数据源控件，来完成 SQL Server 数据的获取工作。	
9.1.4 ObjectDataSource 控件获取类对象数据	337
ObjectDataSource 控件则提供一种方法，可以将页面上的相关数据控件，与中间层业务对象进行绑定。	
9.1.5 获取网站地图文件数据	342
使用 SiteMapDataSource 控件，可以访问站点地图中的导航信息。	
9.1.6 获取来自 LINQ to SQL 数据模型的数据	344
通过 LinqDataSource 数据源控件，Web 开发者可以很方便地将数据展示控件绑定到 LINQ to SQL 数据模型上。	
9.2 使用数据绑定控件显示数据	348
9.2.1 使用 GridView 控件分页展示数据	348
使用 GridView 控件，可以以网格的形式将数据展示在页面中，它还内置了分页功能。	
9.2.2 FormView 实现数据的新增、删除、修改	352
使用 FormView 控件，可以操作数据源中的单个记录。	
9.2.3 使用 DataList 定制数据显示界面	355
DataList 控件在数据的显示格式上，有很大的灵活性。它允许开发人员自定义数据显示模板。	
9.2.4 使用 ListView 实现数据的个性化显示	359
在 ASP.NET 3.5 中，新增了一个数据展示控件 ListView，比起以前的 DataList 控件，该控件具有更高的可定制性，并内置了分页功能。	
9.2.5 使用 ListView 实现数据的编辑、插入和删除	363
使用 ListView 控件，配合数据源控件，在不编写其他代码的情况下，可实现数据的新增、修改和删除等操作。	
9.3 ADO.NET 数据处理	366
9.3.1 手动创建 DataTable	366
作为内存中数据的一个表，DataTable 也可被手动进行创建，本节将对此进行演示。	
9.3.2 合并两个 DataSet 数据表	369
使用了 DataTable 的 Merge() 方法来进行数据表的合并操作。	
9.3.3 DataSet 的复制	372

	DataSet 对象提供了一个 Copy() 方法, 来实现 DataSet 的复制操作, 本节将对此进行演示。	
9.3.4	DataSet 转换为 XML 文件	375
	使用 DataSet 对象的 WriteXml() 方法, 可以将 DataSet 中的数据写入 xml 文件中。	
9.3.5	应用数据库缓存依赖	380
	可以使用 SqlCacheDependency 类创建依赖于数据库中表或行的缓存项。当表中或特定行中发生更改时, 带有依赖项的项便会失效, 并不会从缓存中移除。	
第 10 章	构建 ASP.NET AJAX 站点	383
10.1	ScriptManager 控件实现客户端脚本全球化日期	383
	ASP.NET 中的 Ajax 功能基于 ScriptManager 控件中的设置, 来提供客户端全球化支持。	
10.2	使用 UpdatePanel 控件	386
10.2.1	一个简单的 Ajax 无刷新程序	386
	通过直接使用 UpdatePanel 控件, 指定页面的部分刷新范围, 就可实现页面的局部刷新。	
10.2.2	多个 UpdatePanel 独立更新	388
	实际应用中, 经常会碰到要在一个页面中创建好多独立刷新的区域, 这就需要对同一页面中所存在的多个 UpdatePanel 进行独立刷新。	
10.2.3	母版页中使用 UpdatePanel 控件	391
	在应用了母版页的页面中, 可以使用 Ajax 技术。也可以直接将 Ajax 技术应用于母版页中。	
10.2.4	外部事件刷新 UpdatePanel	393
	在某些特殊要求下, 必须使用外部控件来对 UpdatePanel 内部的内容进行刷新。	
10.2.5	UpdatePanel 中使用验证控件	396
	当把验证控件和输入控件置于 UpdatePanel 中后, 也减少不必要的系统开销。	
10.2.6	UpdatePanel 的错误处理	399
	当 UpdatePanel 控件中的部分页发生错误时, 程序会显示带有错误消息的浏览器消息框, 但这种效果并不理想。本节将演示如何以自定义的方式, 向客户端显示自定义错误信息。	
10.3	使用 UpdateProgress 控件	404
10.3.1	实现页面更新提示	404
	页面内容在更新过程中, 有时候等待是比较漫长的, 并不是每位用户都有耐心来等待, 要是更新的过程中, 能在页面上显示一些类似于进度条或文字性信息之类的提示, 那么用户会更有耐心来等待, 因为用户知道当前网页到底在“做什么”。	
10.3.2	使用客户端脚本取消异步回发	406
	用户在浏览页面时, 对于异步回发, 如果时间过久的话, 用户可随时取消这次控件的交互, 本节将演示如何在异步回发时取消这一操作。	
10.3.3	使用客户端脚本显示更新进度	410
	可通过编写 JavaScript 代码, 来对具有客户端行为的 UpdateProgress 控件进行扩展。	
10.4	使用 Timer 控件	414
10.4.1	Ajax 定时刷新	414
	Timer 控件将按定义的时间间隔执行回发。本节将对此进行演示。	
10.4.2	多个 UpdatePanel 使用 Timer 控件	416
	页面中如果同时存在多个 UpdatePanel 区域时, 可在只使用一个 Timer 控件的情况	