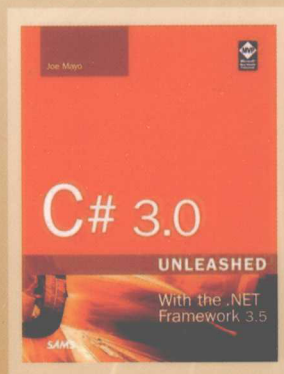


C# 3.0 Unleashed With the .NET Framework 3.5

C#揭秘

[美] Joe Mayo 著
张子阳 余昭辉 王波 译

- LINQ、WPF、WF、WCF，无所不包
- 全面、深入，适合各个层次需要
- C#与.NET平台开发必备



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

微软技术系列

C# 3.0 Unleashed With the .NET Framework 3.5

C#揭秘

[美] Joe Mayo 著
张子阳 余昭辉 王波 译

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

C#揭秘 / (美) 梅奥 (Mayo, J.) 著 ; 张子阳, 余昭辉, 王波译. — 北京 : 人民邮电出版社, 2010.4

(图灵程序设计丛书)

书名原文: C# 3.0 Unleashed: With the .NET Framework 3.5

ISBN 978-7-115-22221-3

I. ①C… II. ①梅… ②张… ③余… ④王… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第014974号

内 容 提 要

C# 提供了很多强大的特性, 特别是通过使用 lambda 表达式、扩展方法和 LINQ, 方便地引入了函数式编程, 使 C# 程序员如虎添翼。本书以 C# 语言的预备知识开篇, 最后以部署代码结束全书, 全面深入地讲解了 .NET 技术的各个知识点, 在广度和深度上恰到好处。本书不仅系统地阐述了 C# 开发相关的重要理论知识, 而且还提供了大量的实例和解决常见问题的最佳方案, 是 .NET 开发人员必备的参考书。

本书适合于各层次的 .NET 开发人员阅读。

图灵程序设计丛书

C# 揭秘

-
- ◆ 著 [美] Joe Mayo
 - 译 张子阳 余昭辉 王 波
 - 责任编辑 傅志红
 - 执行编辑 印星星
 - 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: 800×1000 1/16
 - ◆ 印张: 44.75
 - 字数: 1267千字 2010年4月第1版
 - 印数: 1-3 000册 2010年4月北京第1次印刷

著作权合同登记号 图字: 01-2008-4293号

ISBN 978-7-115-22221-3

定价: 99.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Authorized translation from the English language edition, entitled *C# 3.0 Unleashed: With the .NET Framework 3.5* by Joe Mayo, published by Pearson Education, Inc., publishing as Sams. Copyright © 2008 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Simplified Chinese-language edition copyright © 2010 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Pearson Education Inc.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

前言

欢迎阅读《C#揭秘》，一本C#编程语言的程序员指南和参考书。C#主要是一种面向对象的编程语言，由强调使用组件方法进行软件开发的微软公司创建。C#目前已经到第3版，而且仍在发展当中，本书会引领你进行一次学习的旅程，了解该语言的发展如何帮助你在软件开发过程中取得更多成就。

C#是.NET平台的多种语言之一，.NET平台包括一个称为CLR（Common Language Runtime，公共语言运行库）的运行时引擎和一个庞大的类库。运行时是一个管理代码的虚拟机，同时提供许多其他服务。类库包含差不多数以千计的可重用对象，并为桌面和Web应用程序开发支持多种用户界面技术。

C#作为一门编程语言正在发展着。它最初是一种面向对象的、基于组件的语言，但现在正发展成为一种人们常说的函数式编程语言。贯穿全书，你将会看到许多例子中对象和组件被用作应用程序构建基块（building block）。你也将会看到许多含有LINQ（Language Integrated Query，语言集成查询）的例子，LINQ是一种以声明方式查询数据源的方法，不管数据源是对象、关系数据库、XML或是其他任何形式。

正如C#（以及.NET平台）已经发展了，本书也是一样。本书上一版是一个以语言为中心的学习指南和参考，教你使用C#编程。读者背景很多样化，因为C#是一门新的语言，各种背景的开发者都会使用它编程。所有的应用程序都在命令行上进行编译，你只需要.NET框架SDK和一个编辑器，就能够完成一切了。

就本质而言，本书第一版遵循的理念也体现在这一版中。例如，在开始之前你并不需要知道.NET。如果你使用任何语言进行过编程，本书对你来说应该是很容易的。本书包含了几个命令行的例子，特别是在一开始的时候，因为我相信使用命令行仍然是一项必需且有用的技能。然而，我们很快会转移到Visual Studio 2008（VS2008）IDE（Integrated Development Environment，集成开发环境），本书剩余部分的绝大部分内容都将用到它。然而本书并不要求你必须使用VS2008；我很快便会演示不使用它如何创建应用程序，而附录A介绍了各种命令行选项，并伴有例子（和本书的第一版一样）。尽管如此，VS2008仍是提高工作效率的一个极好工具，不过贯穿本书我都会为通过以代码为中心的RAD（Rapid Application Development，快速应用程序开发）来实现算法提供一些小技巧。

除了涵盖VS2008以外，我也加了几个新的章介绍一些最新的技术，例如WPF（Windows Presentation Foundation，Windows演示基础）、WCF（Windows Communication Foundation，Windows通信基础）和AJAX。如果你喜欢前沿技术，还有一些ADO.NET实体框架（ADO.NET Entity Framework）和ADO.NET数据服务（ADO.NET Data Services）的章节。谈到数据，我已经在这本书中添加了一整部分好几章，专门讨论如何操作数据。

自从2000年7月，当我打开.NET的第一个公测前版本，我就着迷了，并选择C#作为我的语言。我生活得很好而且人们非常需要我的C#技能，即使是在不景气的经济环境下也是如此。最重要的是，作为一名正式的课程讲师和一名开发者，我已经在授课和使用各种工具为客户带去价值的过程中获得了

大量的经验。我希望自己立足实践总结出来并通过本书奉献给读者的所有经验、技巧，能像帮助我一样帮助你。

本书读者对象

如果你已经使用过别的计算机编程语言开发软件，那么理解本书的内容将没有任何问题。你已经知道了如何进行逻辑判断，并且构建迭代式代码。你也能够理解变量和基本的数字系统，例如十六进制。初学者可能想在一开始看些介绍性的内容，就像*Sams Teach Yourself C# in 21 Days*一样。实际上，只要有信心即使是初次接触编程的人也能够看懂本书。

这是一本为任何想要学习C#和.NET的程序员所写的书。它足够基础，让你尽可能看到C#的每一个方面，然而它也足够高级，能够帮你理解每天处理的企业级任务。

本书组织和目标

本书分为了8个部分。为了从头开始逐步学习，本书先讲一些较简单的材料和那些与C#语言本身紧密相关的内容。随后，本书转移到了与C#相关的其他领域，演示了如何使用数据、用户界面技术、Web服务以及其他有用的.NET技术。

第一部分是C#基础，涵盖了基本的C#语言语法等必要知识。第1章讨论.NET平台。这是很重要的一章，因为你需要知道你正在为什么样的环境构建应用程序。它涵盖了作为一个C#开发者应该知道的每一件事，并且应该是你不时返回复习的地方，以提醒自己要成为一个成功的C#开发者，需要掌握哪些必不可少的内容。在第2章，你将学习如何通过命令行和VS2008创建简单的C#应用程序。它只是即将全面介绍VS2008一个开始。第3章学习类型系统，它是成功使用C#开发.NET应用程序的另一个重要里程碑。第4章和第5章分别演示了如何操作字符串和数组。当学习到第7章时，你将拥有足够的技能来编写一个简单的应用程序并会遇到bug。因此，我希望你在继续学习，进入到第二部分更加复杂的面向对象编程之前，要先了解关于使用VS2008调试器的技巧。

第二部分讲述了C#中的对象和组件编程。在本书第一版中，我花费了整个一章讲述基本的面向对象编程概念。在这一版中做了一些改变，我将一部分概念放到了其他章中。通过这种方式，那些已经知道面向对象编程的开发者不需要跳过整章，那些不知道面向对象的开发者也不会完全被忽略。大多数时候，我们讨论的重点是C#如何实现面向对象编程，向现有的面向对象程序员解释那些有趣的细微差别，并向所有的C#开发者解释必需的知识。

第三部分会教你一些C#中更高级的功能。在第二部分理解对象的基础上，你将在这部分学习对象生命周期——对象在什么时候第一次初始化，以及在什么时候从内存中清除。在前面的章节之上构建了一个完整的知识体系，引领你掌握理解.NET内存管理、垃圾收集器所需的知识，作为一个开发者，它对你意味着什么，并且你可以做些什么来确保对象和与其协作的资源被恰当地管理。

第四部分用5章篇幅讲述关于数据的内容。本书第一版的反馈表明，读者想了解更多有关数据的内容。因此，你现在可以学习有关LINQ to Objects、LINQ to SQL、ADO.NET、LINQ to DataSet、XML、LINQ to XML、ADO.NET实体框架、LINQ to Entities、ADO.NET数据服务以及LINQ to Data服务的知识。实际上，这5章并非数据章节的全部，对于我为什么将数据部分放在书中较前面的位置，有一个很好的理由：贯穿本书剩余的部分，我都使用了LINQ。除了学习如何使用所有这些数据访问技术之外，你还将在全书中看到许多例子。

第五部分示范了如何使用各式各样的桌面用户界面技术。你有多种选择，包括在.NET 2.0中进行

了强化的控制台应用程序、Windows窗体，以及WPF。顺便一提，如果你对Silverlight感兴趣的话，可能想先阅读WPF那一章，因为这两种技术都使用了XAML，相同的布局和相同的控件集。这样不仅有助于我为你带去更多关于这些新技术的信息，而且你所学到的一项技术对于另一项技术也是有用的，扩展了你作为一名.NET开发者的技能。

第六部分教你如何创建Web用户界面。ASP.NET是如今.NET的主要Web UI技术，我提供了相当的篇幅来帮助你快速了解它。你要注意有关桌面和Web应用程序区别的讨论，因为它将会影响到你如何开发ASP.NET应用程序。在最近几年，AJAX（异步JavaScript和XML）已经成为了一个热门话题。我将会演示如何使用ASP.NET AJAX（它与VS2008一起发布），以使你的ASP.NET页面更快地响应用户操作。最新的Web UI技术是Silverlight，它能使你创建交互式的网站，而这曾经只可能出现在桌面UI技术中。Silverlight的一些新的功能是更简单地在Web上播放音频、视频，以及动画制作功能；这些新功能大大增强了用户的Web体验（类似于Adobe Flash）。

第七部分带你接触各式各样的通信技术。在一个相互连接的世界中，这些章教你如何使用必需的工具。你将学习如何使用TCP/IP、HTTP以及FTP，并使用.NET框架类库发送电子邮件。Remoting（远程处理）的章节依然保留，Web服务的章节也是一样。然而，新添的一章讲述了新的WCF Web服务。

第八部分涵盖了关于架构和设计的主题。许多程序员学习了C#和前面讨论的所有主题，然后使用他们学到的东西去寻找自己的方式来创建应用程序。如果他们找到了一个创建应用程序的有效途径，那么是值得肯定的。然而，人们通常想要知道将所有这些对象、组件和服务放在一起构建一个可用的应用程序，哪种方法最好。我并没有给出全部答案，因为架构和设计是一个庞大的话题。解决方案因情况而异，然而，我将快速浏览这一主题，演示一些我用过的技术。你将学习C#和.NET如何支持常见的设计模式，以及它们如何让你方便地应用这些模式。我演示了如何创建N层应用程序，并描述了多种方法，你可以采用我提出的这些方法，并以你自己的方式来使用它。我同样演示了如何使用几个.NET工具，包括类设计器，并介绍了WF（Windows Workflow，Windows 工作流），它拥有一个图形设计界面，用于可视化地创建应用程序。

第九部分是一些零散的技巧，它们对你的某些开发来说可能是很重要的。举个例子，大多数程序员偶尔会使用多线程。然而，随着多处理（multiprocessing）和多核CPU变得更加普遍，多线程将是一项大多数程序员都需要的技能，我在本书的这一版中添加了更多多处理/多线程信息。另外，根据你身处的国度，本地化和全球化可能是非常重要的，因此我解释了本地化的基本概念和附属程序集。此外，你做的项目可能仍要与许多遗留的代码进行通信。为了解决这个问题，关于Interop的章节涉及了P/Invoke（用于与Win32 DLL进行互操作）以及COM Interop（用于与COM协作）。本书还有一些运用COM+的信息。对于想要现成解决方案的人来说，我解释了如何使用.NET的跟踪工具，用于检测（instrumenting）和日志（logging）。还有一节关于如何使用现有的性能计数器，以及如何使用自定义性能计数器来检测代码，以通过Windows性能监视器（Windows Performance Monitor）进行诊断。

第十部分帮助你达成最终目标：部署代码。这是一系列的短篇章，教你创建安装程序并部署桌面或者Web应用程序。在此之前，我为你提供了关于程序集的一些更多信息，以及它们是由什么构成的。有关安全性的章节将会帮助你学习.NET CAS（Code Access Security，代码访问安全性）系统是如何工作的。我在讲述中适时给出了几个小技巧，以确保你在开发时能够更加顺畅。

这便是本书的全部内容了。我希望你在学习C#的过程中是幸运的，并希望你会发现本书是一个有帮助的学习工具和有用的参考书。

献辞

献给我美丽的妻子, Maytinee
你是照亮我前进道路的明灯
你的力量和支持让我坚定不移
我孩子们的母亲, 也是我最好的朋友
我爱你并真诚地感谢你

致谢

尽管只有我的名字出现在本书的封面上, 但如果没有很多人的宝贵奉献, 像这样大部头的书是绝对无法写成的。Sams出版社和微软公司的人士, 还有我的朋友和家人, 我永远感激你们。

对于本书此次新版本

我要感谢执行编辑Neil Rowe, 他给了我写作这一版本的机会, 并使得工作得以展开。感谢策划编辑Brook Farling, 感谢他引导我们完成各个步骤, 并给予了帮助。Andrew Beaster在协调作者审阅方面做得很出色, Mark Renfrow则帮助保持书稿内容组织良好, 并一直提供有价值的写作技巧。同样感谢Keith Cline, 他帮我编辑本书并非常漂亮地润色了语句。

我很高兴因为本书而与一些优秀的技术编辑一起工作。感谢Tony Gravagno、Todd Meister和J. Boyd Nolan帮我找出问题, 提出有价值的建议, 他们展现了很高的技术水准。

感谢在微软致力于C#、.NET框架和Visual Studio的人们。我的MVP主管Rafael Munoz经常帮助我找到合适的人来提供信息。

这里也要对Front Range上的几个用户组致以感谢, 他们允许我就.NET主题发表论述, 并且为我提供了实时的反馈: Boulder市的Visual Studio .NET用户组、科罗拉多斯普林斯市的.NET用户组、丹佛市的Visual Studio .NET用户组, 以及科林斯堡市的.NET特别兴趣组。感谢你们每一个人的问题、意见以及提问的勇气。

对于本书第一版

我要感谢执行编辑Shellery Kronzek, 她发现了我并提供了这次难得的机会。她的领导是鼓舞人心的。开发编辑Susan Hobbs非常让人敬畏, 她让我保持专注并组织有序。文字编辑Maryann Steinhart使我的写作看着很棒。我想感谢Sams出版社的其他人员, 如Katie Robinson、Leah Kirkpatrick、Elizabeth Finney、Pamalee Nelson和Laurie McGuire。同样感谢所有的编辑, 索引人员、打印人员、生产人员, 以及对这本书做出贡献的其他Sams出版社人员。

特别要感谢技术编辑Kevin Burton和Bill Craun。他们的技术专业素质和建议绝对是第一流的。他们提供了详细的指导, 他们的意见使本书大为改观。感谢Keith Olsen、Charles Tonkinson、Cedric和Christoph Wille审阅我早期的初稿。

感谢在微软组织作者研讨会和培训的所有人员。他们正在改变我们进行计算的方式, 并且在具有历史意义的运动中引领着产业——这是一个值得特别称赞的动机。特别感谢Eric Gunnerson在百忙中抽空审阅了部分章节。

感谢家人

Maytinee Mayo、Joseph A. Mayo Jr.、Jennifer A. Mayo、Kamonchon Ahantric、Lacee Mayo和June Mayo夫妇、Bob Mayo、Margina Mayo、Richard Mayo、Gary Mayo、Mike Mayo、Tony Gravagno、Tim Hoffman和Kirby Hoffman夫妇、Richard Bickerstaff和Barbara Bickerstaff夫妇、Bobbie Jo Burns、David Burns、

Mistie Lea Bickerstaff、Cecil Sloan爵士及夫人Margaret Sloan、小Cecil Sloan和夫人Jean Sloan、Lou Weiner和Rose Weiner夫妇、Mary Monette和 Ron Monette夫妇、Jack Freeman爵士和Bill Freeman。

感谢朋友和专业人士

Evelyn Black、Harry G. Hall、Arthur E. Richardson、Carl S. Markussen、Ruby Mitchell、Judson Meyer、Hoover McCoy、Bill Morris、Gary Meyer、Tim Leuers、Angela Dees- Prebula、Bob Jangraw、Jean-Paul Massart、Jeff Manners和 Stephanie Manners夫妇、Eddie Alicea、Gary Lefebvre和Gloria Lefebvre夫妇、Bob Turbyfill、Dick Van Bennekom、Barry Patterson、Otis Solomon、Brian Allen。

读者服务

访问我们的网站并且在www.informit.com/title/9780672329814进行注册，可以获得本书的更新、下载或者勘误。

目 录

第一部分 C#基础

第1章 .NET 平台简介.....2

- 1.1 .NET是什么.....2
- 1.2 CLR.....3
 - 1.2.1 为什么CLR如此重要.....3
 - 1.2.2 CLR特性.....4
 - 1.2.3 CLR执行过程.....4
- 1.3 .NET FCL.....6
- 1.4 C#和其他.NET语言.....7
- 1.5 CTS.....7
- 1.6 CLS.....7
- 1.7 小结.....8

第2章 C#和 Visual Studio 2008 概述.....9

- 2.1 编写一个简单的C#程序.....9
- 2.2 创建一个VS2008项目.....12
 - 2.2.1 运行新项目向导.....12
 - 2.2.2 理解解决方案和项目.....14
 - 2.2.3 使用VS2008编写代码.....15
 - 2.2.4 生成和运行程序.....16
 - 2.2.5 设置编译器选项.....18
- 2.3 注释代码.....19
 - 2.3.1 多行注释.....19
 - 2.3.2 单行注释.....19
 - 2.3.3 XML文档注释.....19
- 2.4 标识符和关键字.....21
 - 2.4.1 标识符.....21
 - 2.4.2 关键字.....22
- 2.5 约定和风格.....22

- 2.6 变量和类型.....23
 - 2.6.1 变量.....23
 - 2.6.2 简单类型.....24
- 2.7 明确赋值.....27
- 2.8 与程序交互.....28
 - 2.8.1 与控制台屏幕通信.....28
 - 2.8.2 与命令行通信.....28
 - 2.8.3 VS2008中的命令行选项.....28
 - 2.8.4 从程序返回值.....30
- 2.9 小结.....30

第3章 编写 C#表达式和语句.....31

- 3.1 C#运算符.....31
 - 3.1.1 一元运算符.....31
 - 3.1.2 二元运算符.....33
 - 3.1.3 关系运算符.....34
 - 3.1.4 逻辑运算符.....36
 - 3.1.5 赋值运算符.....38
 - 3.1.6 三元运算符.....38
 - 3.1.7 其他运算符.....39
- 3.2 语句.....40
- 3.3 块和作用域.....40
- 3.4 标签.....41
- 3.5 运算符优先级和结合性.....41
- 3.6 选择和循环语句.....42
 - 3.6.1 if语句.....42
 - 3.6.2 switch语句.....44
 - 3.6.3 C#循环.....45
 - 3.6.4 goto语句.....48
 - 3.6.5 break语句.....49
 - 3.6.6 continue语句.....50

3.6.7 return语句	50	5.1.10 分离和结合字符串	80
3.7 小结	51	5.1.11 操作字符串字符	81
第4章 理解引用类型和值类型	52	5.1.12 通过驻留池影响CLR的字符串处理	82
4.1 引用类型和值类型简介	52	5.2 StringBuilder类	83
4.2 统一类型系统	53	5.2.1 Append方法	83
4.2.1 统一类型系统的工作原理	53	5.2.2 AppendFormat方法	83
4.2.2 使用object进行通用编程	54	5.2.3 EnsureCapacity方法	84
4.2.3 装箱和拆箱对性能的潜在影响	54	5.2.4 ToString()方法	84
4.3 引用类型和值类型的内存分配	56	5.3 正则表达式	84
4.3.1 引用类型内存分配	57	5.3.1 基本的正则表达式操作	85
4.3.2 值类型内存分配	58	5.3.2 深入了解正则表达式	85
4.4 引用类型和值类型赋值	58	5.3.3 练习正则表达式的应用程序	86
4.4.1 引用类型赋值	58	5.4 小结	88
4.4.2 值类型赋值	61	第6章 数组和枚举	89
4.5 引用类型和值类型之间的更多区别	62	6.1 数组	89
4.5.1 引用类型和值类型继承的区别	62	6.1.1 一维数组	90
4.5.2 引用类型和值类型构建和终结的区别	62	6.1.2 多维数组	91
4.5.3 引用类型和值类型对象大小的考虑	63	6.1.3 交错数组	92
4.6 C#和.NET框架类型	63	6.2 System.Array类	93
4.6.1 C#别名和CTS	63	6.2.1 数组边界	93
4.6.2 使用System.Guid	64	6.2.2 搜索和排序	94
4.6.3 操作System.DateTime	65	6.3 使用枚举类型	95
4.7 可空类型	69	6.4 System.Enum结构	97
4.8 小结	70	6.4.1 枚举类型、整型和字符串之间的转换	97
第5章 操作字符串	71	6.4.2 迭代枚举类型成员	98
5.1 C#字符串类型	71	6.4.3 其他System.Enum成员	98
5.1.1 格式化字符串	72	6.5 小结	99
5.1.2 比较字符串	74	第7章 用Visual Studio 2008调试应用程序	100
5.1.3 检查字符串是否相等	75	7.1 单步调试代码	100
5.1.4 连接字符串	75	7.1.1 调试示例程序	100
5.1.5 复制字符串	76	7.1.2 设置断点	100
5.1.6 检查字符串内容	77	7.1.3 检查程序状态	101
5.1.7 提取字符串信息	77	7.1.4 单步调试代码	103
5.1.8 填充和移除字符串输出	78	7.1.5 其他必备的调试命令	104
5.1.9 更改字符串内容	79	7.1.6 使用调试器找出程序错误	104

7.1.7 附加到进程	107	9.3 多态	131
7.2 小结	109	9.3.1 仔细查看多态解决的问题	131
第二部分 使用 C#面向对象编程		9.3.2 使用多态来解决问题	133
第 8 章 设计对象	112	9.3.3 属性多态	135
8.1 对象成员	112	9.3.4 索引器多态	135
8.2 实例和静态成员	113	9.3.5 覆写System.Object类成员	136
8.3 字段	113	9.4 小结	138
8.3.1 常量字段	114	第 10 章 编写方法和自定义操作符	139
8.3.2 readonly字段	114	10.1 方法	139
8.4 方法	114	10.1.1 定义方法	139
8.5 属性	115	10.1.2 本地变量	140
8.5.1 声明属性	115	10.1.3 方法参数	141
8.5.2 使用属性	115	10.2 重载方法	146
8.5.3 自动实现属性	116	10.3 操作符重载	147
8.5.4 VS2008属性代码段	116	10.3.1 用于自定义类型的算数操作符重载	147
8.6 索引器	116	10.3.2 自定义类型上的逻辑操作符重载	149
8.7 回顾一下哪里适合使用分部类型	117	10.3.3 其他操作符重载技巧	150
8.8 静态类	118	10.4 转换和转换操作符重载	151
8.9 System.Object类	118	10.4.1 隐式转换与显式转换	151
8.9.1 检查对象类型	118	10.4.2 自定义值类型转换操作符	154
8.9.2 比较引用	119	10.4.3 自定义引用类型转换操作符	156
8.9.3 检查相等性	119	10.5 部分方法	157
8.9.4 获得散列值	119	10.6 扩展方法	158
8.9.5 克隆对象	120	10.7 小结	159
8.9.6 将对象作为字符串使用	120	第 11 章 错误处理和异常处理	161
8.10 小结	121	11.1 为什么需要异常处理	161
第 9 章 设计面向对象程序	122	11.2 异常处理的语法: try/catch块的基础	162
9.1 继承	122	11.3 使用finally块确保资源清理	163
9.1.1 基类	123	11.4 处理异常	164
9.1.2 调用基类成员	123	11.4.1 处理不同的异常类型	164
9.1.3 隐藏基类成员	124	11.4.2 处理并传递异常	165
9.1.4 版本控制	124	11.4.3 从异常中恢复	167
9.1.5 密封类	126	11.5 自定义异常	169
9.2 封装对象内部信息	127	11.6 checked和unchecked语句	171
9.2.1 数据隐藏	127	11.7 小结	173
9.2.2 使用修饰符支持封装	127		
9.2.3 用于对象的访问修饰符	130		
9.2.4 包含和继承	130		

第 12 章 使用委托和事件进行基于**事件的编程 174**

12.1 揭示委托 174

12.1.1 定义委托 175

12.1.2 创建委托的方法处理器 175

12.1.3 在委托上附加处理器 175

12.1.4 通过委托调用方法 176

12.1.5 使用委托实现多播 176

12.1.6 检查委托的相等性 178

12.2 实现委托推断 178

12.3 匿名方法 179

12.4 事件 180

12.4.1 定义事件处理程序 180

12.4.2 注册事件 181

12.4.3 实现事件 182

12.4.4 触发事件 184

12.4.5 修改事件的add/remove
方法 186

12.5 小结 190

第 13 章 使用命名空间命名和组织**类型 191**

13.1 为什么需要命名空间 191

13.1.1 组织代码 192

13.1.2 避免命名冲突 192

13.2 命名空间指令 192

13.2.1 using指令 192

13.2.2 别名指令 193

13.3 创建命名空间 194

13.4 命名空间成员 197

13.5 作用域与可见性 197

13.6 命名空间别名修饰符 198

13.7 外部命名空间别名 199

13.8 小结 200

第 14 章 抽象类和接口 201

14.1 抽象类 201

14.2 抽象类与接口的不同 203

14.3 实现接口 203

14.4 定义接口类型 204

14.4.1 方法 204

14.4.2 属性 204

14.4.3 索引 205

14.4.4 事件 205

14.5 隐式实现接口 205

14.5.1 单个类的接口实现 205

14.5.2 模拟多态行为 208

14.6 显式实现接口 213

14.7 接口映射 217

14.8 接口继承 219

14.9 小结 221

第三部分 应用 C#高级语言特性**第 15 章 管理对象生命周期 224**

15.1 对象初始化 224

15.1.1 实例构造函数 224

15.1.2 构造函数的重载 226

15.1.3 默认构造函数 227

15.1.4 私有构造函数 227

15.1.5 继承和实例化的顺序 228

15.1.6 静态构造函数 228

15.2 对象初始化器 229

15.3 对象终结器 229

15.4 自动内存管理 230

15.4.1 内存分配 230

15.4.2 垃圾收集器揭秘 231

15.4.3 GC优化 231

15.5 正确的资源清理方式 232

15.5.1 终结器带来的问题 233

15.5.2 Dispose模式 233

15.5.3 using语句 234

15.6 和垃圾收集器交互 235

15.7 小结 237

第 16 章 特性与反射 238

16.1 特性 238

16.1.1 使用特性 239

16.1.2 使用多个特性 239

16.2 特性参数 240

16.2.1 定位参数 240

16.2.2 命名参数	241	第四部分 学习 LINQ 和.NET 数据访问	
16.3 特性目标	241	第 19 章 使用 LINQ 访问数据	288
16.4 创建自定义特性	242	19.1 LINQ to Objects	289
16.5 反射	245	19.1.1 LINQ基本语法	289
16.6 在特性上反射	251	19.1.2 提取投影	290
16.6.1 动态创建代码	251	19.1.3 筛选数据	290
16.6.2 使用Reflection.Emit构建	253	19.1.4 给查询结果排序	290
16.7 小结	256	19.1.5 分组数据	291
第 17 章 泛型与迭代器	257	19.1.6 联结数据	291
17.1 非泛型集合	257	19.1.7 使用分组连接构建层次结构	291
17.2 理解泛型的优点	258	19.2 使用LINQ to SQL查询关系数据	292
17.2.1 泛型解决的问题	258	19.2.1 定义DataContext对象	292
17.2.2 泛型是面向对象的	260	19.2.2 通过DataContext查询	295
17.2.3 在数组、非泛型集合和泛型 集合之间选择	261	19.2.3 修改DataContext对象	295
17.3 构建泛型类型	262	19.2.4 调用存储过程	296
17.3.1 使用泛型实现单链表	262	19.2.5 使用SQL函数	297
17.3.2 超越集合应用泛型	268	19.2.6 使用存储过程修改数据库	297
17.3.3 使用泛型定义类型	270	19.2.7 使用分部方法扩展数据处理 逻辑	299
17.4 实现迭代器	273	19.3 标准查询运算符	301
17.4.1 GetEnumerator迭代器	273	19.3.1 排序运算符	301
17.4.2 方法迭代器	274	19.3.2 集合运算符	302
17.4.3 属性迭代器	275	19.3.3 过滤操作符	303
17.4.4 索引迭代器	276	19.3.4 量词操作符	304
17.4.5 运算符迭代器	276	19.3.5 投影操作符	304
17.4.6 迭代器作为一个值的序列	277	19.3.6 分割操作符	305
17.4.7 销毁迭代器	278	19.3.7 联结操作符	305
17.5 小结	279	19.3.8 分组操作符	306
第 18 章 lambda 表达式和表达式树	280	19.3.9 产生操作符	307
18.1 lambda表达式	280	19.3.10 相等操作符	307
18.1.1 lambda的语法	280	19.3.11 元素操作符	308
18.1.2 使用lambda	281	19.3.12 转换操作符	308
18.1.3 委托和lambda	282	19.3.13 连接操作符	309
18.2 表达式树	285	19.3.14 聚集操作符	309
18.2.1 将lambda转换为表达式树	285	19.4 小结	310
18.2.2 将表达式树转换为lambda	286	第 20 章 使用 ADO.NET 管理数据	311
18.3 小结	286	20.1 ADO.NET的架构	311
		20.1.1 ADO.NET组件	311

20.1.2 连接和无连接模式	312	22.2 使用VS2008创建实体数据模型	335
20.1.3 数据提供者	313	22.3 使用实体SQL查询实体	337
20.2 建立连接	314	22.3.1 访问实体	337
20.3 查看数据	315	22.3.2 查询实体	337
20.4 操作数据	317	22.4 创建自定义实体	338
20.4.1 插入数据	317	22.4.1 映射和架构	339
20.4.2 更新数据	318	22.4.2 向模型中添加自定义实体	339
20.4.3 删除数据	318	22.5 LINQ to Entities	342
20.5 调用存储过程	319	22.5.1 查询实体	342
20.6 使用无连接数据	319	22.5.2 修改实体数据	342
20.6.1 读取数据到DataSet	319	22.6 小结	344
20.6.2 将对DataSet的修改保存到 数据库	320	第 23 章 ADO.NET 数据服务在云计算 数据中的应用	345
20.7 LINQ to DataSet	323	23.1 在项目中添加ADO.NET数据服务	345
20.7.1 使用DataTable作为数据源	323	23.2 通过HTTP URL访问ADO.NET数据 服务	346
20.7.2 强类型字段的访问	323	23.2.1 查看实体集合	346
20.8 小结	324	23.2.2 选择实体	346
第 21 章 操作 XML 数据	325	23.2.3 筛选实体结果	348
21.1 XML数据流	325	23.2.4 给实体排序	349
21.2 写XML	326	23.2.5 跨越实体之间的关联	350
21.3 读取XML	327	23.3 使用数据服务客户端库编写代码	351
21.4 使用XML DOM	328	23.3.1 创建客户端项目	351
21.4.1 使用XPathDocument读取 XML	328	23.3.2 使用WebDataQuery查询 实体	351
21.4.2 使用XmlDocument操作XML	329	23.3.3 添加实体	352
21.5 使用LINQ to XML更容易地操作 XML	329	23.3.4 更新实体	352
21.5.1 LINQ to XML中的对象	330	23.3.5 删除实体	353
21.5.2 创建XML文档	330	23.3.6 使用LINQ to Data Services查 询实体	353
21.5.3 使用LINQ to XML操作命名 空间	331	23.3.7 使用WebDataGen.exe 生成类	354
21.5.4 读取XML文档	331	23.4 小结	354
21.5.5 查询XML文档	332	第五部分 创建桌面用户界面	
21.5.6 修改XML文档	332	第 24 章 挖掘控制台应用程序极限	358
21.6 小结	333	24.1 PasswordGenerator控制台应用程序 介绍	358
第 22 章 使用 ADO.NET 实体框架创建 数据抽象	334	24.2 与用户交互	359
22.1 实体概览	334		

24.3 处理命令行输入	360	26.2.4 StackPanel布局	389
24.4 设置控制台的颜色和位置	360	26.2.5 UniformGrid布局	389
24.5 小结	362	26.2.6 Grid布局	390
第 25 章 编写 Windows Forms 应用		26.2.7 DockPanel布局	392
程序	363	26.3 WPF控件	392
25.1 Windows Forms基础	363	26.3.1 Border	393
25.2 VS2008对Windows Forms的支持	366	26.3.2 Button	393
25.2.1 可视化设计环境	366	26.3.3 CheckBox控件	393
25.2.2 Windows Forms应用程序中		26.3.4 ComboBox控件	393
的文件	366	26.3.5 ContentControl控件	393
25.2.3 可视化设计器如何工作	367	26.3.6 DockPanel控件	394
25.3 使用Windows Forms控件	372	26.3.7 DocumentViewer控件	394
25.4 MenuStrip、StatusStrip和		26.3.8 Ellipse控件	394
ToolStrip控件	373	26.3.9 Expander控件	395
25.5 数据网格和数据绑定	375	26.3.10 Frame控件	395
25.5.1 建立数据绑定项目	375	26.3.11 Grid控件	395
25.5.2 绑定数据到ListBox控件	375	26.3.12 GridSplitter控件	395
25.5.3 绑定数据到DataGridView		26.3.13 GroupBox控件	396
控件	376	26.3.14 Image控件	396
25.6 GDI+要素	377	26.3.15 Label控件	396
25.6.1 Brush、Pen和Graphics		26.3.16 ListBox控件	396
对象	377	26.3.17 ListView控件	396
25.6.2 字体和绘制文本	378	26.3.18 MediaElement控件	396
25.7 附加窗体和对话框	379	26.3.19 Menu控件	397
25.7.1 模式和无模式对话框	379	26.3.20 PasswordBox控件	397
25.7.2 窗体通信	380	26.3.21 ProgressBar控件	397
25.7.3 通用对话框	381	26.3.22 RadioButton控件	397
25.8 小结	383	26.3.23 Rectangle控件	398
第 26 章 创建 WPF 应用程序	384	26.3.24 RichTextBox控件	398
26.1 XAML就够了	384	26.3.25 ScrollBar控件	398
26.1.1 介绍WPF应用程序	385	26.3.26 ScrollViewer控件	398
26.1.2 研究XAML	385	26.3.27 Separator控件	399
26.1.3 XAML中的控件	386	26.3.28 Slider控件	399
26.2 管理布局	387	26.3.29 StackPanel控件	399
26.2.1 控件对齐方式、尺寸和方框		26.3.30 StatusBar控件	399
模型	387	26.3.31 Tab控件	399
26.2.2 Canvas布局	388	26.3.32 TextBlock控件	400
26.2.3 WrapPanel布局	389	26.3.33 TextBox控件	400
		26.3.34 ToolBar控件	400
		26.3.35 ToolBarPanel控件	400

26.3.36	ToolBarTray控件	400	父控件	421
26.3.37	TreeView控件	401	27.6 导航	424
26.3.38	UniformGrid控件	401	27.6.1 用web.sitemap定义站点 布局	425
26.3.39	ViewBox控件	401	27.6.2 Menu控件导航	425
26.3.40	WindowsFormsHost控件	401	27.6.3 实现TreeView	426
26.3.41	WrapPanel控件	402	27.6.4 使用SiteMapPath添加面包 屑导航	428
26.4	事件处理	402	27.7 网站主题	428
26.5	数据绑定	403	27.7.1 设置主题	428
26.5.1	数据绑定概述	403	27.7.2 创建皮肤	428
26.5.2	显示数据列表	403	27.7.3 创建样式表	430
26.6	使用样式	406	27.8 网站安全	430
26.7	小结	407	27.9 数据绑定	431
第六部分 设计 Web 用户界面			27.9.1 设置业务对象	431
第 27 章 使用 ASP.NET 创建 Web 应用程序			27.9.2 简单数据绑定	432
27.1	Web应用程序模型	410	27.9.3 ObjectDataSource与数据 绑定	432
27.1.1	ASP.NET 页面请求概述	410	27.10 小结	434
27.1.2	C#代码驻留之地	411	第 28 章 用 ASP.NET AJAX 加强 Web 应用程序的交互性	
27.1.3	可伸缩性和状态管理从何 而来	411	28.1 何为AJAX	435
27.1.4	如何理解直观性能	412	28.2 设置ASP.NET AJAX站点	436
27.1.5	我为何使用ASP.NET	412	28.3 AJAX页面生命周期	437
27.2	使用VS2008创建ASP.NET项目	412	28.4 加载自定义脚本库	438
27.3	ASP.NET页入门	413	28.5 ASP.NET AJAX控件	439
27.3.1	Web窗体的构成	413	28.5.1 UpdatePanel控件	439
27.3.2	代码隐藏与页面生命周期	415	28.5.2 UpdateProgress控件	440
27.4	控件	417	28.5.3 Timer控件	441
27.4.1	服务器控件	417	28.6 通过JavaScript访问控件	441
27.4.2	HTML控件	418	28.6.1 使用JavaScript简化控件ID 访问	442
27.5	状态管理	419	28.6.2 访问已损坏的ASP.NET 控件ID	443
27.5.1	Application全局状态	419	28.7 通过ASP.NET AJAX调用Web服务	446
27.5.2	在缓存中保存可更新信息	419	28.7.1 使用Web服务和AJAX的原 因与取舍	446
27.5.3	保存单一请求状态	420	28.7.2 使用AJAX和Web服务	447
27.5.4	发布Cookie	420	28.8 小结	449
27.5.5	特定用户信息与Session 状态	421		
27.5.6	理解ViewState中的页面 状态	421		
27.5.7	页面重用与母版页和自定			