

PROGRAMMER TO PROGRAMMER™



# Fast Track C#

Written and tested for final release of .NET v1.0

## 高效掌握 C#

K.Scott Allen 等著  
Neil Avent  
康 博 译



清华大学出版社  
<http://www.tup.com.cn>  
<http://www.tup.tsinghua.edu.cn>



# 高效掌握 C#

K. Scott Allen 等著

Neil Avent

康 博 译

清华大学出版社

(京)新登字 158 号

北京市版权局著作权合同登记号: 01-2002-3204

## 内 容 简 介

C#是 Microsoft 专门为 .NET Framework 平台开发的一门新型的编程语言。虽然很多语言都能够编写 .NET 代码,但 C#是惟一针对 .NET Framework 设计的语言。因此在今后几年内,C#将会成为编写 .NET 应用程序的首选语言。

本书是您学习编写 C#应用程序的必备向导,它以循序渐进的教学方式,逐步讲解了 C#的基础知识和高级应用,并通过具体的示例为您展示了编写 C#应用程序的方法和技巧。读者可以从本书了解到 C#的入门知识、面向对象功能、ADO.NET、COM 和 COM+的互操作、ASP.NET 以及如何编写 Web 服务等相关内容。

本书是一本适合各层次读者的程序开发类图书,该书可以帮助初学者和高级开发人员全面学习 C#语言。特别适合于那些希望深入了解 .NET Framework 下的 C#语言,从而更好地探究其内部原理的读者。

K. Scott Allen Neil Avent:Fast Track C#

EISBN: 1-861007-11-6

Copyright©2002 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved.For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由英国乐思出版公司授权清华大学出版社出版。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,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

高效掌握 C#/(美)艾伦等著;康博译.——北京:清华大学出版社,2002

书名原文:Fast Track C#

ISBN 7-302-05885-7

I. 高... II. ①艾...②康... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 071699 号

出版者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.com.cn> <http://www.tup.tsinghua.edu.cn>

责任编辑:陈宗斌

印刷者:北京牛山世兴印刷厂

发行者:新华书店总店北京发行所

开本:787×1092 1/16 印张:24.75 字数:633 千字

版次:2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

书号:ISBN 7-302-05885-7/TP·3491

印数:0001~5000

定价:45.00 元

# 出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 [www.wrox.com](http://www.wrox.com) 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 [p2p.wrox.com](http://p2p.wrox.com) 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

# 前 言

C#是 Microsoft 的一种新语言，是专门为新平台 .NET Framework 而设计的语言。

.NET 基本的前提非常简单：即所有的 .NET 程序都被编译为一种中间语言(IL 或 MSIL 语言)，而不是可以被计算机的处理程序所理解的本机代码。当安装应用程序或运行应用程序时，这些 IL 代码就会被编译成本机代码。运行在 .NET Framework 上的代码由称为公共语言运行库(即 Common Language Runtime，简称 CLR)的运行环境管理。这个运行库提供了诸如自动无用存储单元收集以及类型安全检查之类的内存管理功能，它在某些方面与 Java 的虚拟机相似。

.NET Framework 还提供了大量的基类，所有使用 CLR 的语言都可以使用这些基类的功能，并且，它允许开发人员完成很多公共的编程任务。从长远来看(即如果将 .NET Framework 移植到非 Windows 平台)，这些基类可以通过避免调用特定操作系统的 API 来提供一定程度的平台无关性。

在过去的几年里，主要使用 Microsoft 平台的开发人员已经有两种基本语言供选择：他们可以选择功能强大的 Visual C++，也可选择易于使用的 Visual Basic。尽管 Visual Basic 确实为准备拓展其功能的人们提供了很多强大的功能，而 Visual C++又设计了大量 Appwizards 减轻 C++ 开发人员的开发难度，虽然通常用 Visual C++构建的应用程序要长一些，但是，在 Visual C++ 中开发的应用程序要比在 Visual Basic 中开发的应用程序的性能好一些。

.NET Framework 以两种方式改变了所有这一切。首先，Visual Basic .NET 比其先前的版本功能要强大得多。所有的 .NET 语言都将编译成中间语言，并且能够利用由 CLR 提供的功能。除此之外 Visual Basic .NET 中还添加了许多新的强大功能，比如真正的面向对象的特性以及使用 Try...Catch 语句处理异常的功能。然而，更重要的发展或许是一种全新语言的出现——C#，本书所要讨论的主题便是 C#。

C#是由 C++派生而来的新语言，但它是为支持快速应用程序开发(Rapid Application Development)而设计的，与 Visual Basic 的开发方式相似。C#允许我们使用设计器快速而简单地设计 Windows 应用程序和 Web 应用程序；Visual Studio.NET 负责处理大部分代码，比如实例化控件，以及编写事件处理程序等。这当然也包括由 Common Language Runtime 提供的功能，比如自动无用单元收集功能。不过，C#还为我们提供了很多 C++所没有的功能，C#甚至允许我们通过“不安全”代码块(绕过类型安全特性的代码)中的指针直接访问内存地址。

## 1. 本书读者对象

本书适用于想要学习 C#和 .NET 平台的所有读者。但是本书不适用于初学者——我们假定读者具有一定的编程经验，比如具有使用 C++进行 Windows 编程方面的经验。或者，读者可能具有使用 Java 语言的背景，并且想要了解 C#和 .NET 中的内幕。后面的一些章节需要您具备一些 Microsoft 技术的基本知识，比如 ADO 技术和 COM 技术。



## 2. 使用本书的要求

本书及其代码与 Microsoft Visual Studio .NET 的发布版本对应。因此使用本书的最低要求与 Microsoft Visual Studio .NET 产品的要求相同：PII-450 CPU、Microsoft Windows 2000 Profesional 或者更高的版本，至少 128 MB 的内存，大约 3GB 的可用硬盘空间。

## 3. 本书内容

本书旨在介绍 .NET 平台上的 C# 编程技术。正如我们已经提到的，C# 和 .NET 是紧密联系的。因为所有为 .NET Framework 编写的代码都可以广泛地应用 .NET 的基类，例如为输入/输出、数据操作、文件处理编写代码，所以，本书除了详细介绍 C# 语言本身的语法外，还将大量介绍上述内容。

详细讲解 C# 语言之前，我们首先对 .NET Framework 和 C# 做一个总体性的介绍。然后，介绍 .NET 编程的基础知识——.NET 类库，构建 Windows 应用程序，使用 ADO.NET 进行数据访问，.NET 单元部署——程序集。最后，介绍一些较高级的技术：将 C# 程序与 COM 和组件服务(Component Services)进行集成，构建 ASP .NET Web 应用程序以及使用 C# 构建 Web 服务。

现在让我们来看一下本书每一章将要介绍的内容：

### 第 1 章：.NET 和 CLR 概述

本章将介绍 .NET Framework 及其相关问题，还将介绍它如何解决这些问题。除此之外，我们还要讨论公共语言运行库(Common Language Runtime，即 CLR)，中间语言(Intermediate Language，即 IL)和通用类型系统(Common Type System，即 CTS)。

### 第 2 章：C# 简介

本章将介绍 C# 语言。在这一章中，我们首先讨论为什么还需要另一种编程语言，并将 C# 与当前开发人员正在使用的其他编程语言做一下比较。然后介绍可以在 .NET 环境中使用 C# 构建的应用程序类型。最后，介绍 C# 编译器。

### 第 3 章：C# 基础知识

从本章开始，我们将详细介绍 C# 语言。在这一章中，我们将对 C# 与 C++ 和 Java 的所有相似处和不同处进行比较。还要介绍 C# 程序的特性，包括数据类型、运算符和该语言中使用的语句。另外，还将介绍程序控制和循环，方法的调用和重载。另外，还将介绍程序错误以及如何处理这些错误。

### 第 4 章：C# 的面向对象功能

本章将介绍 C# 中处理对象的功能。在介绍结构和类之前，我们首先讨论一下枚举。我们还将讨论类的所有成员，包括字段、属性、方法、事件。除此之外，还要介绍诸如继承之类的面向对象概念，以及如何在 C# 中实现这些功能。本章中介绍的其他内容包括，接口和抽象类、运算符重载，以及索引器。

## 第 5 章: C#的高级特性

现在我们来了解一下 C#中的一些更高级的功能。首先,介绍处理类时需要用到的一些高级技术,比如可变参数列表。然后,介绍 C#的预处理器指令(我们可以使用这些指令来更改编译代码的过程)、使用 C#中的指针和不安全代码、委托和事件。最后,介绍 XML 文档。

## 第 6 章: C# .NET 编程

在介绍完 C#语言本身后,从这一章开始我们将讲解如何在 .NET 环境中使用这种语言。首先,我们要介绍一下反射,它允许我们在运行时检查类型和对象,并且查明是哪种类型或对象及其功能。然后,我们会介绍可用于代码中以提供额外的运行时信息的 .NET 的特性和标记。接下来,我们来看一下集合和数组方面的内容。最后,介绍 .NET 是如何支持用 C#编写多线程应用程序的。

## 第 7 章: 使用 .NET 基类库

本章将介绍一下最常用的基类。我们将编写范例程序以说明运行中的基类,并说明如何执行普通的 C#编程任务,例如字符串操纵、文件处理、读写文件、读写 XML 及 XML 串行化。

## 第 8 章: 构建 Windows 应用程序

本章将介绍使用 Windows 窗体进行快速应用程序开发(Rapid Application Development, 即 RAD)。简要介绍如何使用 Visual Studio 创建应用程序后,我们来看一下利用 Visual Studio .NET 集成开发环境(Integrated Development Environment, IDE)将控件添加到 Windows 窗体中。在这里,我们用一个范例应用程序来演示创建菜单、处理事件、构建 Windows 应用程序的其他内容,比如标准对话框等。接下来,介绍如何从 Visual Studio .NET 内部部署应用程序。

## 第 9 章: 程序集和 ILDASM

在这一章中,我们将介绍程序集和清单的概念。程序集是包含代码和用来描述组件的所有元数据(即组件的清单)的 .NET 组件。ILDASM 工具允许我们查看程序集的内容。.NET Framework 使用共享名称来定位程序集,而不用考虑它们在文件系统中的实际存储位置,因此我们要介绍一下如何为程序集创建共享名。这样一来程序集可以在多个应用程序中使用。另外,我们还要看一下代码访问的安全性。

## 第 10 章: 使用 ADO.NET 进行数据访问

在这一章中,我们要介绍在 .NET 平台上使用 ADO.NET 进行数据访问,这是 Microsoft 最新的数据访问技术。简要介绍数据付费用户和提供程序类后,我们将介绍只可向前的只读数据阅读器,及如何使用存储过程和 ADO.NET 中的事务处理。其次,将介绍如何使用 DataSet 访问和更新数据库,如何将 DataSet 绑定到 DataGrid 上,并介绍一些 Visual Studio .NET 用来进行数据访问的内置功能。最后,我们来分析一些 ADO.NET 处理 XML 的功能。

## 第 11 章: COM 与 COM+互操作

接下来,我们介绍如何将现有的代码与 C#应用程序和组件进行集成。首先,看一下如何从



C#应用程序中调用现有的 COM 组件，在这里我们使用了早期绑定和后期绑定。随后，通过创建一个封装 COM 组件及导出一个类型库，讨论如何调用 C#组件，在这里我们将这些 C#组件看作是 COM 组件。

在对组件服务(Component Services)做了综述之后，我们要介绍如何利用 .NET 组件使用这些功能。我们将在 .NET 环境下创建事务处理组件以及使用 COM+服务对它们进行注册等内容，然后介绍如何从 C#组件中利用 COM+的一些其他主要功能，如消息队列。

最后，我们将介绍使用 PInvoke(Platform Invocation Services, 平台调用服务)来调用非托管函数。

## 第 12 章: ASP.NET

这是本书的倒数第 2 章，在这一章中，我们将介绍如何使用 ASP.NET(ASP 最新的版本)创建 Web 应用程序。在对这些供出售的新功能作了综述之后，我们将展示一个创建简单的 ASP.NET Web 应用程序的例子。我们将在保持状态和验证用户输入方面介绍服务器端控件的用途。最后，介绍配置 Web 应用程序，以及一些 ASP.NET 提供的监控工具。

## 第 13 章: Web 服务

最后一章，我们来介绍另一种新的 Web 应用程序—Web 服务。服务是一种通过 Internet 在服务器上提供应用程序逻辑的方法。它允许客户机使用服务器上的数据，并不需要依赖特定的、预先构建好的静态或动态 Web 页。介绍过 Web 服务及其使用的协议后，我们来演示一下如何构建一个 Web 服务以及客户程序如何使用这项服务，以及如何使用异步方法来处理由网络混乱引起的、可能存在的服务中断。

### 4. 技术支持

如果您希望直接向专家咨询书中的问题，可以发送电子邮件到 [support@wrox.com](mailto:support@wrox.com)，要求在邮件的主题栏中输入本书的书名和 ISBN(国际标准图书编号)的后 4 位数字。一个典型的电子邮件应包括以下内容：

- 在主题栏中必须有本书的书名、ISBN 的后 4 位数字和问题的页数。
- 信息的正文应包括读者的名字、联系信息和问题。

我们将不回复您的无用邮件，因为，我们仅仅需要有用的详细资料，以便节约您和我们的时间。当您发送一个电子邮件信息时，它将经过下面一系列的支持：

- 用户支持：首先，您的信息将被递送到我们的用户支持人员手中，并由他们阅读。它们将归档一些被频繁问到的问题，并将立即回答有关本书或者 Web 站点的任何常见问题。

- 编辑支持：接着，一些有深度的问题将被送到对本书负责的技术编辑手中，他们对程序设计语言或者特定的产品有着丰富的经验，能够回答相关主题的详细技术问题。一旦问题得到了解决，编辑就会将勘误表发到 Web 站点上。

- 作者支持：最后，如果编辑不能回答您的问题(这种情况很少发生)，他们将请求本书的作者。我们将尽量保护作者免受干扰，以便不影响其写作。然而，我们也非常高兴转寄给他们一些特殊的问题。所有 Wrox 公司的作者都为他们的书提供技术支持。作为回应，他们将发送电子邮件给用户和编辑，进而使所有的读者受益。

Wrox 公司的支持过程仅仅对那些与我们出版的书目内容直接相关的问题提供支持, 对于超出常规书目支持的问题, 您可以从 <http://p2p.wrox.com/> 论坛中的公共列表中获得支持信息。

### 源代码

书中所有可用的源代码在 Wrox.com 站点上都可找到。在您访问 Wrox 公司站点(地址为 <http://www.wrox.com/>)时, 通过 Search 工具或书名列表, 可以方便地定位需要的书目。然后, 单击 Code 列中的 Download 超链接, 或者单击本书的详细页面中的 Download Code 超链接, 就可以下载相应的范例代码。

从我们的站点上下载的文件都是使用 WinZip 压缩过的文档。保存附件到本地磁盘上的文件夹中后, 需要使用一个解压缩程序(例如 WinZip 或 PKUnzip)来解压缩文件。在解压缩文件时, 通常将代码解压缩到每一章所在的文件夹中。在解压缩的过程中, 应确保解压缩程序(WinZip、PKUnzip, 其他)被设置为使用原有文件夹名。

### 勘误表

我们已经尽最大努力确保本书中的文本和代码没有错误, 但是错误仍然在所难免。如果您发现本书存在错误, 例如拼写错误或不正确的代码段, 请给我们发来反馈信息, 我们将不胜感激。勘误表的发送可以节约其他读者学习本书的时间, 而且能够帮助我们提供更高质量的信息。请将您的反馈信息发送电子邮件到 [support@wrox.com](mailto:support@wrox.com)。我们会检查您的反馈信息, 如果正确, 将其粘贴到本书的勘误页面上, 或者在本书的后续版本中使用。

要在我们的站点上找到勘误表, 请访问 <http://www.wrox.com/>, 并通过 Search 工具来定位本书页面。然后, 在本书的详细页面上单击 Book Errata 超连接即可。在该页面上, 您可以查看已被提交和验证过的所有勘误。您可以单击 Submit Errata 链接, 以便将您所发现的勘误告知我们。

### p2p.wrox.com

为了便于作者和其他人讨论, 特将讨论内容加入到 P2P 站点的邮件列表中, 而且我们唯一的系统将 programmer to programmer™(由程序员为程序员而著)的编程理念与邮件列表、论坛、新闻组以及所有其他服务内容(一对一的邮件支持系统除外)相联系。如果您向 P2P 发送一个问题, 相信它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家所检查到。无论您是在阅读本书, 还是在开发自己的应用程序, 都可以在 p2p.wrox.com 站点中找到许多对自己有所帮助的邮件列表。

按照下面的步骤可以预订一个邮件列表:

1. 登录 <http://p2p.wrox.com/> 站点。
2. 从左边的主菜单栏选择一个适当的类别。
3. 单击希望加入的邮件列表。
4. 按照说明订阅并填写自己的邮件地址和密码。
5. 回复您收到的确认邮件。
6. 使用预定管理程序加入更多的邮件列表并设置自己的邮件选项。

# 目 录

|                           |    |
|---------------------------|----|
| 第 1 章 .NET 和 CLR 概述       | 1  |
| 1.1 .NET 简介               | 1  |
| 1.1.1 .NET 与 COM 的现有问题    | 2  |
| 1.1.2 COM 和 COM+          | 3  |
| 1.2 编译并执行一个 C# 应用程序       | 3  |
| 1.2.1 编译代码                | 4  |
| 1.2.2 执行                  | 4  |
| 1.3 Microsoft 中间语言        | 5  |
| 1.3.1 Microsoft 中间语言的编程方法 | 6  |
| 1.3.2 Microsoft 中间语言数据处理  | 6  |
| 1.4 通用类型系统(CTS)           | 7  |
| 1.5 公共语言规范(CTS)           | 9  |
| 1.6 语言的互操作性               | 10 |
| 1.7 .NET 基类库(BCL)         | 10 |
| 1.8 程序集                   | 11 |
| 1.8.1 元数据和清单              | 11 |
| 1.8.2 共享程序集               | 12 |
| 1.8.3 私有程序集               | 12 |
| 1.8.4 命名空间                | 12 |
| 1.9 应用程序域                 | 13 |
| 1.10 实时编译                 | 14 |
| 1.11 无用单元回收器              | 15 |
| 1.11.1 异常                 | 15 |
| 1.11.2 安全性                | 16 |
| 1.11.3 属性                 | 16 |
| 1.12 本章小结                 | 16 |
| 第 2 章 C# 简介               | 17 |
| 2.1 需要 C# 的原因             | 17 |
| 2.2 现有语言中存在的问题            | 18 |
| 2.2.1 C 与 C++             | 18 |
| 2.2.2 Visual Basic        | 19 |
| 2.2.3 Java                | 19 |



|        |                             |    |
|--------|-----------------------------|----|
| 2.3    | 进入 C# .....                 | 20 |
| 2.4    | C#和 Java 的比较 .....          | 23 |
| 2.5    | C#与其他.NET 语言的比较 .....       | 24 |
| 2.5.1  | 托管 C++ .....                | 24 |
| 2.5.2  | Visual Basic .NET .....     | 24 |
| 2.5.3  | J# .....                    | 24 |
| 2.5.4  | JScript.NET .....           | 24 |
| 2.6    | C#应用程序 .....                | 25 |
| 2.6.1  | ASP.NET .....               | 25 |
| 2.6.2  | Web 窗体 .....                | 26 |
| 2.6.3  | 移动的 ASP.NET .....           | 26 |
| 2.6.4  | Web 控件 .....                | 26 |
| 2.6.5  | Web 服务 .....                | 26 |
| 2.6.6  | Windows 应用程序 .....          | 27 |
| 2.6.7  | Windows 控件 .....            | 27 |
| 2.6.8  | Windows 服务 .....            | 27 |
| 2.6.9  | 类库 .....                    | 27 |
| 2.6.10 | 控制台应用程序 .....               | 28 |
| 2.7    | C#编译器 .....                 | 28 |
| 2.8    | 本章小结 .....                  | 29 |
| 第 3 章  | C#基础知识 .....                | 31 |
| 3.1    | C#入门 .....                  | 31 |
| 3.1.1  | C#中的可执行文件 .....             | 32 |
| 3.1.2  | 使用 Visual Studio .NET ..... | 32 |
| 3.1.3  | 程序结构 .....                  | 33 |
| 3.2    | 基本语法 .....                  | 34 |
| 3.2.1  | 注释 .....                    | 34 |
| 3.2.2  | 数据类型 .....                  | 35 |
| 3.2.3  | 语句 .....                    | 50 |
| 3.2.4  | 定义和调用方法 .....               | 55 |
| 3.2.5  | 错误处理 .....                  | 57 |
| 3.2.6  | 输入/输出控制 .....               | 61 |
| 3.3    | 本章小结 .....                  | 64 |
| 第 4 章  | C#的面向对象功能 .....             | 65 |
| 4.1    | 枚举 .....                    | 65 |
| 4.2    | 结构 .....                    | 66 |
| 4.2.1  | 定义结构的方法 .....               | 68 |

|       |                          |     |
|-------|--------------------------|-----|
| 4.2.2 | 结构构造函数                   | 69  |
| 4.2.3 | 声明的顺序                    | 70  |
| 4.3   | 类                        | 70  |
| 4.3.1 | 构造函数和析构函数                | 73  |
| 4.3.2 | 常量和只读成员                  | 76  |
| 4.3.3 | this 引用                  | 76  |
| 4.3.4 | 属性                       | 77  |
| 4.3.5 | 静态成员                     | 79  |
| 4.4   | 继承                       | 80  |
| 4.4.1 | 使用基类引用                   | 81  |
| 4.4.2 | 受保护的访问                   | 81  |
| 4.4.3 | 调用基类构造函数                 | 82  |
| 4.5   | 虚方法                      | 82  |
| 4.6   | 隐藏基类方法                   | 83  |
| 4.7   | 抽象类和方法                   | 84  |
| 4.8   | 防止派生                     | 85  |
| 4.9   | 类型强制转换                   | 85  |
| 4.10  | 对象类                      | 86  |
| 4.11  | 运算符重载                    | 90  |
| 4.12  | 接口                       | 91  |
| 4.13  | 索引器                      | 95  |
| 4.14  | 本章小结                     | 97  |
| 第 5 章 | C#的高级特性                  | 99  |
| 5.1   | 深入了解类                    | 99  |
| 5.1.1 | 可变参数列表                   | 99  |
| 5.1.2 | 用户定义转换                   | 100 |
| 5.2   | 预处理程序                    | 102 |
| 5.2.1 | #define, #undef 和 #if 语句 | 102 |
| 5.2.2 | #warning 和 #error 语句     | 104 |
| 5.2.3 | #line 语句                 | 104 |
| 5.2.4 | #region 和 #endregion 语句  | 105 |
| 5.3   | 委托                       | 105 |
| 5.4   | 事件                       | 109 |
| 5.5   | 不安全代码                    | 113 |
| 5.5.1 | 指针简介                     | 114 |
| 5.5.2 | fixed 关键字                | 115 |
| 5.5.3 | stackalloc 关键字           | 116 |



|                                     |            |
|-------------------------------------|------------|
| 5.6 XML 文档 .....                    | 116        |
| 5.6.1 生成文档 .....                    | 117        |
| 5.6.2 一个 XML 文档实例 .....             | 119        |
| 5.7 本章小结 .....                      | 122        |
| <b>第 6 章 C# .NET 编程 .....</b>       | <b>123</b> |
| 6.1 命名空间和 using 关键字 .....           | 123        |
| 6.1.1 命名空间 .....                    | 123        |
| 6.1.2 在程序中使用命名空间 .....              | 125        |
| 6.2 映射和类型库 .....                    | 128        |
| 6.3 属性 .....                        | 136        |
| 6.3.1 使用属性 .....                    | 137        |
| 6.3.2 编写定制属性 .....                  | 138        |
| 6.3.3 使用属性 .....                    | 142        |
| 6.4 集合与数组 .....                     | 144        |
| 6.4.1 数组 .....                      | 144        |
| 6.4.2 集合 .....                      | 149        |
| 6.5 线程 .....                        | 153        |
| 6.5.1 应用程序域 .....                   | 153        |
| 6.5.2 System.Threading 命名空间 .....   | 154        |
| 6.6 本章小结 .....                      | 159        |
| <b>第 7 章 使用 .NET 基类库 .....</b>      | <b>160</b> |
| 7.1 .NET 类库命名空间 .....               | 160        |
| 7.2 C# 中的字符串处理 .....                | 161        |
| 7.2.1 System.String 类 .....         | 161        |
| 7.2.2 处理字符串 .....                   | 163        |
| 7.2.3 StringBuilder 类: 易变的字符串 ..... | 164        |
| 7.3 C# 中的文件系统 .....                 | 165        |
| 7.3.1 文件和文件夹 .....                  | 166        |
| 7.3.2 读取、写入和流 .....                 | 172        |
| 7.4 在 C# 中使用 XML .....              | 177        |
| 7.4.1 向 XML 文件中写入数据 .....           | 178        |
| 7.4.2 读取 XML 文件 .....               | 180        |
| 7.4.3 XML 串行化 .....                 | 182        |
| 7.5 HTML 内容描述符 .....                | 185        |
| 7.6 本章小结 .....                      | 189        |

|                              |     |
|------------------------------|-----|
| 第 8 章 构建 Windows 应用程序        | 190 |
| 8.1 Windows 窗体               | 190 |
| 8.1.1 Windows 控件             | 192 |
| 8.1.2 事件句柄                   | 194 |
| 8.1.3 自定义窗体                  | 195 |
| 8.1.4 简单的计算程序                | 196 |
| 8.2 部署应用程序                   | 206 |
| 8.3 本章小结                     | 212 |
| 第 9 章 程序集和 ILDASM 工具         | 213 |
| 9.1 什么是程序集                   | 213 |
| 9.1.1 使用程序集的好处: DLL Hell 的终结 | 214 |
| 9.1.2 使用应用程序的好处: 并行处理        | 214 |
| 9.1.3 程序集包含的内容               | 214 |
| 9.2 程序集的结构                   | 218 |
| 9.2.1 单文件程序集                 | 218 |
| 9.2.2 多文件程序集                 | 218 |
| 9.2.3 附属程序集                  | 219 |
| 9.3 程序集示例: Shapes 类库         | 220 |
| 9.4 程序集属性                    | 222 |
| 9.4.1 程序集的信息属性               | 223 |
| 9.4.2 文化属性                   | 223 |
| 9.4.3 版本属性                   | 224 |
| 9.4.4 强名属性                   | 225 |
| 9.5 查看程序集的内容: ILDASM         | 226 |
| 9.5.1 查看清单的内容                | 228 |
| 9.5.2 查看一个方法的内容              | 230 |
| 9.5.3 将程序集内容作为中间语言输出         | 230 |
| 9.6 私有程序集                    | 231 |
| 9.6.1 单客户程序                  | 231 |
| 9.6.2 私有程序集绑定和搜索             | 232 |
| 9.7 共享程序集                    | 235 |
| 9.7.1 构建一个共享程序集              | 235 |
| 9.7.2 将共享程序集安装到全局程序集缓存中      | 236 |
| 9.7.3 在全局程序集缓存中同时安装不同的版本     | 239 |
| 9.7.4 共享程序集搜索                | 239 |
| 9.7.5 共享程序集配置文件              | 240 |



|        |                               |     |
|--------|-------------------------------|-----|
| 9.8    | 数字签名和签名代码 .....               | 242 |
| 9.9    | 代码访问的安全性 .....                | 244 |
| 9.9.1  | 证据和安全策略 .....                 | 244 |
| 9.9.2  | 代码许可 .....                    | 244 |
| 9.10   | 程序集的高级内容 .....                | 245 |
| 9.10.1 | 反射和程序集的种类 .....               | 245 |
| 9.10.2 | 显式加载和隐式加载 .....               | 246 |
| 9.10.3 | 动态程序集 .....                   | 247 |
| 9.10.4 | 交叉语言支持 .....                  | 247 |
| 9.10.5 | COM 互操作程序集 .....              | 248 |
| 9.11   | 本章小结 .....                    | 248 |
| 第 10 章 | 使用 ADO.NET 进行数据访问 .....       | 250 |
| 10.1   | ADO.NET .....                 | 250 |
| 10.2   | .NET 数据提供程序 .....             | 251 |
| 10.2.1 | SQL Server .NET 提供程序 .....    | 251 |
| 10.2.2 | OLE DB .NET 提供程序 .....        | 251 |
| 10.2.3 | ODBC .NET 提供程序 .....          | 251 |
| 10.3   | ADO.NET 类结构 .....             | 252 |
| 10.3.1 | 数据使用类 .....                   | 252 |
| 10.3.2 | .NET 提供程序类 .....              | 254 |
| 10.3.3 | 其他的 .NET Framework 类 .....    | 255 |
| 10.4   | 使用 ADO.NET .....              | 255 |
| 10.4.1 | 示例数据库: MSDE 和 Northwind ..... | 256 |
| 10.4.2 | .NET 数据提供程序示例 .....           | 256 |
| 10.4.3 | DataSet 示例 .....              | 269 |
| 10.4.4 | 使用 DataGrid 显示数据 .....        | 277 |
| 10.4.5 | 强类型的 DataSet .....            | 280 |
| 10.4.6 | 在 ADO.NET 中使用 XML .....       | 282 |
| 10.5   | 本章小结 .....                    | 287 |
| 第 11 章 | COM 与 COM+ 互操作 .....          | 288 |
| 11.1   | COM 互操作 .....                 | 288 |
| 11.1.1 | 生成元数据 .....                   | 289 |
| 11.1.2 | 创建和调用 .....                   | 291 |
| 11.1.3 | 使用反射 .....                    | 293 |
| 11.1.4 | 带有 ActiveX 控件的互操作 .....       | 295 |

|        |                                |     |
|--------|--------------------------------|-----|
| 11.2   | COM 客户端的互操作 .....              | 295 |
| 11.2.1 | 私有部署 .....                     | 296 |
| 11.2.2 | 共享部署 .....                     | 301 |
| 11.3   | 利用 COM+ 服务 .....               | 301 |
| 11.4   | 平台调用服务 .....                   | 312 |
| 11.4.1 | 识别 DLL 函数 .....                | 312 |
| 11.4.2 | 使用 DllImportAttribute 属性 ..... | 314 |
| 11.4.3 | 原型调用排序 .....                   | 315 |
| 11.5   | 本章小结 .....                     | 317 |
| 第 12 章 | ASP.NET .....                  | 318 |
| 12.1   | ASP.NET 的由来 .....              | 318 |
| 12.2   | 创建一个 Web Form 项目 .....         | 319 |
| 12.2.1 | 添加控件 .....                     | 320 |
| 12.2.2 | 后台编码文件 .....                   | 322 |
| 12.2.3 | 用户控件 .....                     | 328 |
| 12.2.4 | 其他服务器控件 .....                  | 330 |
| 12.2.5 | 使用 Web 控件进行数据绑定 .....          | 332 |
| 12.3   | ASP.NET 的配置 .....              | 337 |
| 12.4   | ASP.NET 中的诊断——跟踪 .....         | 341 |
| 12.4.1 | 页面级跟踪 .....                    | 341 |
| 12.4.2 | 应用程序级跟踪 .....                  | 343 |
| 12.5   | 本章小结 .....                     | 344 |
| 第 13 章 | Web 服务 .....                   | 345 |
| 13.1   | Web 服务涉及的领域 .....              | 345 |
| 13.1.1 | 平台的互操作性 .....                  | 345 |
| 13.1.2 | 分布式计算 .....                    | 345 |
| 13.2   | 编写一个 Web 服务程序 .....            | 346 |
| 13.3   | 测试与调试 Web 服务 .....             | 349 |
| 13.3.1 | 使用 HTTP POST .....             | 350 |
| 13.3.2 | 使用 SOAP 协议 .....               | 351 |
| 13.3.3 | Web 服务描述语言 .....               | 352 |
| 13.4   | 创建 Web 服务客户程序 .....            | 354 |
| 13.5   | XML 串型化 .....                  | 356 |
| 13.6   | XML 规范 .....                   | 360 |
| 13.7   | Windows Form 客户程序 .....        | 361 |



---

|                           |     |
|---------------------------|-----|
| 13.8 异步 Web 服务调用 .....    | 363 |
| 13.9 WebMethod 特性详解 ..... | 366 |
| 13.9.1 Web 服务事务处理 .....   | 366 |
| 13.9.2 Web 服务高速缓存 .....   | 367 |
| 13.9.3 状态管理 .....         | 368 |
| 13.9.4 状态管理的缺点 .....      | 370 |
| 13.10 SOAP 头 .....        | 370 |
| 13.11 本章小结 .....          | 373 |