

微软 Visual Studio.NET 程序开发系列丛书

C# 技术内幕

[美] Tom Archer 著
侯晓霞 柴洪辉 译

清华大学出版社

(京)新登字 158 号

内 容 简 介

C#由 C 和 C++语言发展而来，是一种简单、现代、面向对象、类型安全的语言。本书分为基础篇、C#类基础、编写代码和高级 C# 4 个部分，分别描述了关于 C#和.NET 开发环境的一个特定方面的内容，其中包括 .NET 框架介绍、C#编程原则、类型系统、类、特性、接口、多线程编程和使用组合体等。

本书简明易懂，思路清晰，结合恰当的示例代码，特别适于希望学习 C#和.NET 开发环境的读者，以及已经具有 C 家族语言（C、C++或 Java）背景的读者阅读。

Inside C#

Microsoft Press

Copyright © 2001 by Microsoft Corporation

Original English language edition published by Microsoft Press, a Division of Microsoft Corporation

All rights reserved.

No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher. For sale in the People's Republic of China only.

本书中文简体版由 Microsoft Press 授权清华大学出版社出版发行，未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分内容。

北京市版权局著作权合同登记号：图字 01-2001-0120 号

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。

书 名：C#技术内幕

作 者：[美] Tom Archer

译 者：侯晓霞 柴洪辉

责任编辑：孙 岩

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.tsinghua.edu.cn>

印 刷 者：

发 行 者：新华书店总店北京发行所

开 本：787×960 1/16 印张：23.25 字数：512 千字

版 次：2001 年 11 月第 1 版 2001 年 11 月第 1 次印刷

书 号：ISBN 7-900637-59-1

印 数：0001~5000

定 价：49.00 元（含 1 张光盘）

《微软 Visual Studio .NET 程序开发系列丛书》序

删除的内容: 系列丛书
带格式的

删除的内容:
带格式的

.NET 技术简介

删除的内容:

.NET 技术是 2000 年微软公司推出的全新概念, 对于它很难做出一个明确的定义, 它代表了一个集合、一个环境、一个编程的基础结构。其目的是将互联网本身作为构建新一代操作系统的基础, 对互联网和操作系统的设计思想进行延伸。具体地说, .NET 技术就是要在不同的网站之间建立起协定, 促进网站之间的协同合作, 实现信息的自动交流, 从而帮助用户最大限度地获取信息、并对他们的数据进行简单、高效的管理。

删除的内容: 促使

对于最终用户来说, .NET 技术的实现将使计算机的功能得到大幅度的提升, 而计算机的操作却变得更加简单。他们将完全摆脱人为的硬件束缚, 利用任何设备、通过任何系统、在任何时间、任何地点访问互联网的多维时空, 并对其进行跨应用程序的集成。用户对个人信息任何修改——无论是通过个人电脑、便携设备还是灵通卡——将即时和自动地通知到所有需要这些信息的地方。

删除的内容: 将

对于应用程序的开发人员来说, .NET 的意义也许更为重大。通过 .NET 技术, 程序开发人员不必再像过去那样通过集成本地系统服务来构建应用程序, 花费大量的精力来考虑如何构建基本的结构, 而无法专注于如何实现软件所特有的商业价值。现在, 程序开发人员可以访问 Microsoft .NET 所提供的内容丰富的类库, 直接使用各种各样来自于 Internet 的模块来构建自己的应用程序, 而无需为不同的工作环境重新编写程序, 快速、高效地开发应用程序。

删除的内容: 不用

对于网络开发人员来说, 借助 .NET 技术所提供的、基于 XML 的松散耦合技术, 他们终于可以真正地应用程序逻辑分发到网络上, 而不必顾及服务端与客户端是否属于同一种结构、执行机制是否相同、接口是否对应。Microsoft .NET 将把 ASP 和程序集成带到一个新的水平上, 将集成程序和其他的集成或非集成程序整合; 把这些程序风格化; 形成针对这些应用程序编写新程序的能力; 并可以在离线状态下运行它们。

删除的内容: 把

删除的内容: 了

删除的内容:

Visual Studio.NET 简介

Visual Studio.NET 是 Microsoft 为实现其 .NET 技术而开发的一整套工具组件。它把开发功能强大、性能可靠的企业网络解决方案进行了简化。通过提供终端到终端的网络开发

删除的内容:

删除的内容:

删除的内容: 加以

II C#技术内幕

能力以及可伸缩、可复用的服务器端组件，Visual Studio.NET 将会大大地提高生产率，并促使商务活动更加有效地关注快速多变而又充满竞争的市场需求。

Visual Studio.NET 具有许多令人激动人心的特性，其中的某些特性是对早期 Visual Studio®版本的增强，而另外一些特性则是全新的。这些新特性包括：一个单一的集成化开发环境、Visual Basic.NET 当中面向对象的新特性、以及大量在产品开发生命周期中使用的工具。

带格式的

删除的内容: 的

Visual Studio.NET 的主要作用在于：

- **提供加速开发过程的高效工具**

带格式的: 项目符号和编号

Visual Studio.NET 提供了一个统一的、紧密集成的可视化编程环境，以帮助您简化开发网络应用程序的过程、缩短学习使用的时间。通过共享的 HTML、XML 和样式单编辑器，您可以轻轻松松地借助包括 C#在内的任何一种 Visual Studio 语言来开发网络应用程序。

删除的内容: 来

- **提供对各种各样的网络应用程序的快速设计能力**

借助于 Web Forms，程序员们可以使用他们在开发基于窗体的桌面应用程序时所使用的技巧来创建跨平台、跨浏览器的网络应用程序。Web Forms 被用于网络的服务器端，它能高速地运行，并生成符合 HTML 3.2 规范、能够在各种浏览器上阅读的文档。

- **利用 XML 和 Web Services 来简化分布式计算**

Web Services 借助标准的 Internet 协议在网络上调用商务逻辑。HTTP 被作为 Web Services 传输的基础协议，该协议使得对功能的请求能够穿越各种团体所使用的防火墙。XML 被用来对上述功能请求的参数进行格式统一，从而使得这些请求能够适用于所有的软件和硬件。这样使得对 Web Services 的访问可以通过任何一种语言、使用任何一种组件模型、在任何一种操作系统上实现。

删除的内容: 用

- **快速构建中间层商务组件**

Visual Studio 的一个核心目标就是要为基于服务器的应用程序提供应用程序快速部署(Rapid Application Development, RAP)工具。Visual Studio.NET 在创建中层商务逻辑时所表现出的效率丝毫不逊于 Visual Basic®开发窗体程序时的效率。利用 Visual Studio.NET 创建的组件将为您商务运作提供足够的功能和伸缩性。

删除的内容: 色

带格式的

- **构建可靠的可伸缩解决方案**

利用 Visual Studio.NET，您可以非常轻松地创建具有自动伸缩能力的可靠的应用程序和组件。例如，如果使用 Visual C++来编写网络应用程序，新推出的 ALT Server 将会极大地提高性能。

Visual Studio.NET 的其他新性能包括：

● 一种新的语言：C#

带格式的：项目符号和编号

C#语言是一种优雅、简单、类型安全、面向对象的编程语言。它在向 C 和 C++ 的用户提供 RAD 的同时，保持了 C/C++ 所特有的强大功能和控制能力。C# 的关键特性包括：

- 具有与 C++ 十分相似的模型和语法。99% 的语句、表达式和运算符都与 C++ 相同
- 具有与 COM+ 服务完全的互换性
- 提供对 COM 和平台的完全支持能力，使您能够轻松地迁移现有的代码
- 提供自动的无用单元收集能力
- 提供可扩展的分类元数据，使您可以声明新类型和各种元数据
- 对基于网络的组件交互提供 XML 支持

删除的内容：的

Visual Studio.NET 的新特性使之成为基于微软 .NET 框架(Microsoft .NET Framework，微软下一代的网络应用程序开发平台)的应用程序提供了一个完整的开发环境。Visual Studio.NET 提供的关键性基础技术简化了安全、可伸缩、高度可用的网络应用程序和 Web Services 的创建、部署和更新的方法，并为程序员们施展他们的技巧提供了广阔的空间。

除此以外，Microsoft .NET Framework 还帮助网络开发人员使用自己所喜爱的语言，按照使用本地对象的方式来使用 Web Services。程序开发人员可以把注意力投向如何增强产品所特有的功能，以为公司赢得竞争优势。这样的结果是更快的产品开发速度、更高的开发效率和更高的产品质量。

删除的内容：

本套丛书的内容与安排

本套丛书取材于由微软出版社提供、独家授权的大量第一手资料，本套丛书目前包括 5 册，分别为：

删除的内容：

删除的内容：五册

删除的内容：六

带格式的：项目符号和编号

● 《C#语言参考手册》

深入详细地介绍 C# 语言的语法规则和例外情况，为学习者提供从 C/C++ 到 C# 的快速入门指南。

● 《C#编程技术》

用大量的程序代码来比较 C# 与 C/C++ 的异同，并具体地说明如何利用 C# 访问 Microsoft .NET 类库，编写功能强大的应用程序。

● 《C# 技术内幕》

带格式的：项目符号和编号

精心组织成几个逻辑上有序的部分，分别描述了 C# 和 .NET 开发环境的一个特定内容：基础篇、C# 类基础、编写代码、高级 C#。

● 《Visual Studio .NET 开发环境使用指南》

带格式的：项目符号和编号

删除的内容：应用程序

详细地介绍新一代的应用程序开发环境的使用方法和技巧，为您对 Visual Studio.NET 开发环境的亲身体验提供全面详细的使用指南。

● 《XML 3.0 技术内幕》

详细深入地介绍最新版本的 XML (Extensible Markup Language, 可扩展标记语言) 的语言规范和使用方法，帮助读者快速深入地了解这[项](#)当今网络的核心技术。

随着技术的发展，并根据读者的需要，我们将陆续增加新的[书目](#)。

本套书在编写的过程中面临着如何迅速地把最新的 IT 技术传播给国内的广大读者的巨大压力。迫于时限，在章节篇幅的安排上秉承“形式服从于内容”的宗旨，因而造成某些章节篇幅远远超出其他章节的情况。另外，由于本书所讨论技术的新颖性，使得可资借鉴的中文资料相当稀少，加上编者们的水平有限，故而某些术语的译称可能[并](#)不尽如人意，甚至可能还会有不少错误，对于这些问题，恳请广大读者批评指正。

删除的内容: <#> 《Visual Basic.NET 程序设计》
在对 Visual Basic 6.0 和 Visual Basic.NET 比较的基础上，详细地介绍后者的新特性，以及这些特性如何使得大型复杂的应用程序的开发变得轻松容易。

删除的内容: 门

删除的内容: 内容

删除的内容: 能

编 者

2001 年 8 月

删除的内容: 4

前言

写作缘由

做了 20 年程序员之后——每思及此，我就觉得韶光不再！——我基本上已经达到这样一种境界：编程对于我来说，开始显得有些陈腐了。不要错解我的意思：如果我是一个亿万富翁并且不需要工作的话，我仍然可能会继续书写代码，因为这是我真正乐于从事的事情。但现实却是我常常会产生一种“穷途末路”的感觉。终于，Microsoft .NET 和 C# 出现在我的面前，一个全新的世界打开了！我曾与其他几个朋友交谈过，他们跟我一样，都被 .NET 的美好前景“重新唤醒”。我们终于拥有了这项激动人心的新技术，可以用来解决已经与我们纠缠多年的问题（比如：多语言开发环境、大型复杂系统的分发配置和版本问题等等）。我之所以写这本书，是因为书写代码再一次能够让我激动不已。每天清晨醒来，我都忍不住会想一想今天将要学到很棒的新东西。我希望在您学习这门语言的时候，也能够分享我这份狂热的心情。

目前任何写作有关 C# 的人都不得不边写边学。如果你在学习 SDK 或者目前使用的编程语言期间曾经编写过应用程序的话——谁又没有这样做过呢？——你就会知道这事有点难。现在试着想象一下：如果你写了一点东西，而有千千万万的人将要阅读它的时候，你又会怎样呢？最大的问题在于做了一半的时候——一旦你醒悟你在做什么——你就会渴望重新设计整个过程并且重新来过！但是很显然，由于最后期限来临，这样的想法并不实际。我相信本书是一个学习 C# 语言的好工具。因为我是边写边学，书中肯定会有一些不妥之处，也有一些地方我本来可以做得更好。但是，如果有机会发行第二版的话，我保证我们都会从我书写本书时我的个人学习过程中受益匪浅，因为，我所遇到的问题也就是您将要面临的问题。

最后，我想说的是我欢迎任何有关本书的反馈意见。我不是那种自认“我写书，所以我很牛气”的人，我只是一个很普通的人，幸运地得到了创作本书的机会而已。我总是乐于向他人学习并且乐此不疲。您可以通过 <http://www.TheCodeChannel.com> 与我联系。

本书的读者对象

本书适合于希望了解 C# 和 .NET 开发环境的读者阅读。我已经说过，.NET 是一个激动人心的平台，同时也是 Microsoft Windows 开发环境未来的潮流。本书假定读者已经具有 C 系列语言之一（C、C++ 或 Java）的背景知识。如果说还有什么其他的条件，那就是在编写应

删除的内容: 引子

删除的内容:

带格式的

删除的内容:

删除的内容: 二十

删除的内容: 会

删除的内容: Microsoft.NET

删除的内容: 新的、

删除的内容: 等等

删除的内容: 的新的很酷的

删除的内容: 的

删除的内容: 一

删除的内容:

删除的内容: 在

删除的内容: 著作

删除的内容: 书写

删除的内容: 重新设计

删除的内容: 不

删除的内容: (这一段中读者 ... [1]

删除的内容:

删除的内容: 书写

删除的内容:

带格式的

删除的内容: 谁应该阅读

删除的内容:

删除的内容: 学习

删除的内容: 家族

删除的内容: 背景

删除的内容: :

删除的内容: 我想

删除的内容: 另外的 ... [2]

删除的内容: 书写

II C#技术内幕

用程序时学习和探索新思路的那种渴望。因为您正手握此书，您可能已经心存这种渴望！

本书的组织结构

本书精心组织成逻辑上有序的几个部分：每一部分包含了若干章，描述了 C#或.NET 开发的一大类特定内容。

本书以第 1 部分“基础篇”开始，这部分面向的读者是入门级别的 C#程序员和刚刚接触 .NET 开发环境的读者。这部分中的章节提供了对 .NET 的介绍，并且说明如何创建和测试您的第一个 C#应用程序。

在第 2 部分“C#类基础”中，介绍了在 C#中定义和使用类的一些基本要素。这部分中的章节着重于为您打下一个坚实的基础，以便了解在 C#中支持哪些成员(枚举、属性、数组等等)，以及在 C#应用程序中如何定义和使用它们。

尽管在学习定义类成员的几个章节中您已经写过代码了，但是直到第 3 部分“编写代码”，才开始介绍各种任务的不同侧面，诸如控制程序流、处理错误(包括异常)、以及使用代表来书写事件处理器等等。

本书以第 4 部分“高级 C#”结束。作为一个典型的老程序员，我最喜欢书写这部分内容。其中包括了多线程编程、反射、使用非托管代码(包括 COM 互用性)和版本管理等。

关于本书配套光盘

本书包括一张配套光盘。如果您在 Windows 中启用了 AutoRun 功能，当您把这张光盘插入到光驱中时就会出现一个屏幕，为您提供安装选项。要想手动打开这个安装屏幕，可以从该光盘的根目录下运行 StartCD 程序。StartCD 程序为您提供到 MSDN 的连接，利用该连接您可以下载最新版本的 Microsoft .NET Framework SDK，在编译和运行示例程序时将用到它。

本书的示例程序位于光盘上的 Code 文件夹中。您可以直接从光盘上查看示例程序，也可以从 StartCD 上使用安装程序将它们安装到您的硬盘上。

注意：如果您不能浏览 Samples 文件夹中的文件，有可能是因为您的光盘驱动程序太老，以至于不支持长文件名。如果是这样，要想浏览这些文件，您就必须使用安装程序将这些示例文件安装到您的硬盘上。

系统要求

您如果想从本书中最大程度地获益，我强烈建议在阅读每一章时都一定要演练一遍示例程序。这样的话，您将需要安装最新的 .NET Framework SDK。在写作本书期间，它包括 .NET 运行环境和 C#编译器。此外，我特意避免使用 Visual Studio.NET 产品，以便集中精力在编程语言和运行环境之上，而不对您特定的开发环境做任何限制。因此，本书中所

删除的内容: 的...吧 [3]

删除的内容:

带格式的

删除的内容:

删除的内容: 几个

删除的内容: 的

删除的内容: 一组... [4]

删除的内容:

删除的内容: 一部分

删除的内容:

删除的内容: 第二

删除的内容: 特性

删除的内容:

删除的内容: 书...第... [5]

删除的内容: 书...开... [6]

删除的内容: 句...柄... [7]

删除的内容:

删除的内容: 第四... [8]

删除的内容:

带格式的

删除的内容: 随书发行的

删除的内容: ... [9]

删除的内容: 盘... [10]

删除的内容: 光盘上... [11]

删除的内容: 您

删除的内容:

删除的内容: 文件夹... [12]

删除的内容:

带格式的

删除的内容: 文件夹

删除的内容: 了

删除的内容:

带格式的

删除的内容: 系统需求

删除的内容:

删除的内容: 推荐... [13]

删除的内容: 过... [14]

有的示例程序都是从命令行编译和运行的。

下面我们再来看一下组成 Microsoft .NET 的关键组件和一些相关的技术。

• 新的编程语言 C#

C#是 C 和 C++语言家族中第一种面向组件的编程语言。它从 C 和 C++派生而来，是一种使用简便、面向对象、类型安全的现代编程语言。C#兼具 Visual Basic 的高效性和 C++的强大功能。

• 通用运行语言

高性能的通用运行语言(Common Language Runtime, CLR)包含执行引擎、垃圾收集器、即时编译、安全系统和丰富的类框架(即 Microsoft .NET 框架)。CLR 从底层设计，能够支持多种语言。

• 通用语言规范

通用语言规范(Common Language Specification, CLS)描述了各种语言所共有的功能。CLS 这种相对很低的限制有助于创建许多与 CLS 兼容的语言。这些语言可获得的好处包括：对 Microsoft .NET 框架功能的完全访问和与其他兼容语言之间充分的互用性。比如，一个 Visual Basic 的类可以从 C#类中继承而来并且覆盖掉其虚拟方法。

• 以 CLR 为目标的大量语言

由 Microsoft 提供的以 CLR 为目标的编程语言包括：Visual Basic、带有受控扩展的 Visual C++、Visual C#和 JScript。至于由第三方所提供的编程语言则数不胜数。

• Web 服务

目前的全球互联网主要是由单独的站点所组成。如果用户要访问多个站点以便货比三家，比如为一组人员做旅行计划，那么他所访问的这些 Web 站点之间通常并没有合作。下一代的互联网则是基于 Web 站点之间的合作的网络。这样做的理由很简单：合作的 Web 站点能够更好地满足用户的需求。Microsoft 的 Web 服务技术鼓励 Web 站点之间的合作，因此允许站点之间通过基于 XML 的标准协议进行通信，即使这些 Web 站点在编程语言和应用平台上都相互独立，也不对此造成影响。还有很多重要的 Web 服务将基于 C#和 Windows 上运行的 CLR，但这个框架结构却的确是开放的。

• Visual Studio.NET

Visual Studio.NET 将上述所有的部分联系在一起，以便在各种各样的编程语言中更简便地创建各种各样的组件、应用程序和服务程序。

既然我们已经涉及到了一些与 C#相关的重要技术，那么就对 C#本身进行一些详细的探讨吧。开发人员已经为他们所选择的编程语言投入了大量的心血，因此新语言的责任就是通过保留简单、持续改进和谨慎创新的组合来证明自己的价值。

删除的内容:

删除的内容: 致谢

首先我要感谢我的编辑——Devon Musgrave 的帮助。我从来没想过要成为一名“作者”。我只是一名程序员，碰巧愿意帮助别人，而写书正是帮助别人的一条途径。如果没有 Devon 帮助我将所说的话整理成我想说的话，恐怕这本书没人会读得懂。谢谢你，Devon！

... [15]

删除的内容: 部分

删除的内容: <#>

带格式的

带格式的: 项目符号和编号

删除的内容: Microsoft

删除的内容: 公共

带格式的

带格式的: 项目符号和编号

删除的内容: 语言

带格式的

删除的内容: 环境

带格式的

删除的内容: 公共. ... [16]

删除的内容: 公共

带格式的: 项目符号和编号

删除的内容: 公共. ... [17]

带格式的: 项目符号和编号

删除的内容: 丰富. ... [18]

删除的内容: 管理. ... [19]

带格式的: 项目符号和编号

删除的内容: 的基于 ... [20]

带格式的: 项目符号和编号

删除的内容: 更大范围

删除的内容: ... [21]

简单保留

希波克拉底说过：“要养成这样的习惯：要么有所帮助，要么至少没有伤害”。在我们设计 C# 的过程中，“没有伤害”这个原则起到了非常重要的作用。如果一个 C 或 C++ 的功能将问题描述得很清楚，我们会全盘保留。更重要的是，C# 在很多关键的领域都借用了 C 和 C++ 中的要素，如表达式、语句和整个的语法。由于在很多典型程序中都会包含这些要素，因此 C 和 C++ 开发人员定会 C# 一见倾心。

删除的内容:

带格式的

持续不断的改进

持续改进的地方简直太多了，以至于在这个简短的前言中无法一一陈述。但以下这几个改变的地方值得引起关注，因为它们消除了一些在 C 和 C++ 中很常见并且很耗时的错误：

- 变量在使用之前必须初始化，因此消除了由于变量未初始化而导致的故障。
- 类似 if 和 while 的语句要求使用布尔值。开发人员如果意外地使用了赋值操作符(=)代替相等操作符(==)的话，在编译时就能发现错误。
- 在 switch 语句中不允许出现 贯行(fall-through)。开发人员如果意外地省略了 break 语句的话，在编译时就会发现错误。

删除的内容:

带格式的

带格式的

带格式的

带格式的：项目符号和编号

深思熟虑的创新

在 C# 的类型系统中可以发现一些深层次的创新，包括以下这些部分：

- C# 类型系统采用了自动内存管理，因此将开发人员从耗费时间和容易出错的手动内存管理中解放出来。与大多数类型系统不同，C# 类型系统还允许对指针类型和对象地址进行直接操作(这些手动内存管理技术只允许在某些特定的安全环境下使用)。
- C# 类型系统是统一的——所有的类型都是对象。通过对“装箱”和“卸箱”这些概念的创新使用，C# 在值类型和引用类型之间架起了一道桥梁，允许将任何数据都当作对象看待。
- 特性、方法和事件是基本要素。很多语言都省略了对特性和事件的支持，因此造成在编程语言和相关框架之间不必要的 mismatch。比如，如果框架支持特性而编程语言并不支持特性，那么给特性做增量就会很麻烦(如 `o.SetValue(o.GetValue()+1)`)。如果该语言也支持特性，操作就简单多了(`o.Value++`)。
- C# 支持属性，允许对组件的描述性信息进行定义和使用。能够定义新种类的描述性信息对于语言设计人员来说总是一个强大的工具。现在所有的 C# 开发人员都拥有这个能力了。

删除的内容: 落空

带格式的

删除的内容: 谨慎

带格式的

带格式的：项目符号和编号

删除的内容: 的

删除的内容: 显得相当笨拙

亲爱的读者，衷心地希望你们能愉快地写下第一个 C# 程序，并且在以后的日子里写出更多更好的 C# 程序。

删除的内容:

删除的内容:

作者: Tom Archer

页 I: [1] 删除的内容	suny	2001-6-5 9:13:00
(这一段中译者认为用“你”比用“您”要好，请斟酌。)		
页 I: [2] 删除的内容	zhukang	2001-10-18 9:46:00
另外的一个预备要求		
页 II: [3] 删除的内容	zhukang	2001-10-26 15:27:00
的		
页 II: [3] 删除的内容	zhukang	2001-10-18 9:46:00
吧		
页 II: [3] 删除的内容	zhukang	2001-10-18 9:46:00
存在		
页 II: [4] 删除的内容	zhukang	2001-10-26 15:31:00
一组		
页 II: [4] 删除的内容	zhukang	2001-10-26 15:31:00
节		
页 II: [4] 删除的内容	zhukang	2001-10-26 15:31:00
和		
页 II: [4] 删除的内容	zhukang	2001-10-26 15:31:00
环境		
页 II: [4] 删除的内容	zhukang	2001-10-26 15:31:00
一个		
页 II: [5] 删除的内容	zhukang	2001-10-26 15:31:00
书		
页 II: [5] 删除的内容	zhukang	2001-10-18 9:47:00
第三		
页 II: [6] 删除的内容	suny	2001-6-6 14:33:00
书		
页 II: [6] 删除的内容	suny	2001-6-5 9:21:00
开始		

页 II: [6] 删除的内容 您	suny	2001-6-5 9:21:00
页 II: [6] 删除的内容 会见识	suny	2001-6-5 9:21:00
页 II: [7] 删除的内容 句	zhukang	2001-10-18 9:47:00
页 II: [7] 删除的内容 柄	zhukang	2001-10-18 9:47:00
页 II: [8] 删除的内容 第四	zhukang	2001-10-18 9:47:00
页 II: [8] 删除的内容 映射	zhukang	2001-10-18 9:47:00
页 II: [8] 删除的内容 未管理	zhukang	2001-10-18 9:47:00
页 II: [8] 删除的内容 化	zhukang	2001-10-26 15:31:00
页 II: [9] 删除的内容	suny	2001-8-7 9:07:00
页 II: [9] 删除的内容 随书发行的	suny	2001-6-5 9:22:00
页 II: [10] 删除的内容 盘	zhukang	2001-10-18 9:47:00
页 II: [10] 删除的内容 动器	zhukang	2001-10-18 9:48:00
页 II: [11] 删除的内容 光盘上所包含的 eBook 的连接，这是一个本书示例文件的安装程序；同时也提供到	suny	2001-8-7 17:39:00
页 II: [11] 删除的内容 Microsoft.NET	suny	2001-8-7 10:04:00

页 II: [12] 删除的内容 文件夹	zhukang	2001-10-18 9:48:00
页 II: [12] 删除的内容 之	zhukang	2001-10-18 9:48:00
页 II: [13] 删除的内容 推荐	zhukang	2001-10-18 9:49:00
页 II: [13] 删除的内容 别忘了	zhukang	2001-10-18 9:49:00
页 II: [14] 删除的内容 过	suny	2001-6-5 9:25:00
页 II: [14] 删除的内容 运行时	suny	2001-6-6 22:23:00
页 II: [14] 删除的内容 运行时	suny	2001-6-6 22:23:00
页 III: [15] 删除的内容 致谢	suny	2001-8-9 14:07:00

首先我要感谢我的编辑——Devon Musgrave 的帮助。我从来没想过要成为一名“作者”。我只是一名程序员，碰巧愿意帮助别人，而写书正是帮助别人的一条途径。如果没有 Devon 帮助我将所说的话整理成我想说的话，恐怕这本书没人会读得懂。谢谢你，Devon！

我还要感谢本书的技术编辑——Brian Johnson。在将书中的文字与示例程序一一相配的过程中，Brian 功不可没。在我们准备出版本书时，也幸亏有他的帮助才能克服掉几个编译器后来所出现的改变。我写作本书时依据的是 C# 的一个测试版，到了 C# 编译器向一般大众发布的时候，本书中或许会有一两个示例程序无法正常运行。但是，所有这一切都不是 Brian 的过错，他在测试每个示例程序时已经极度一丝不苟了。谢谢你，Brian！

Microsoft 出版社的另外两位同志同样值得感谢：Anne Hamilton 和 Danielle Bird。他们两位负责启动这个项目并且给了我写作本书的机会。他们对我的信任使我度过了写作本书的难关，我非常感谢他们的支持。谢谢！

我还要感谢以下这些 Microsoft 同事，他们在我尝试学习 C# 语言和 .NET BCL 的过程中对我的请教不厌其烦：Joe Nalewabau(他解答了一些关于 C# 的早期问题)、Brian Harry(他提供了有关最终定稿的信息)和 Steven Pratschner(他提供了有关全局汇编缓存和汇编版本化的帮助)。同样，非常感谢 Scott Wiltamuth，他写的前言非常棒！

最后，我想要感谢 Aravind Corera，他在书写未管理代码一章中贡献非凡。是他在 COM 部分的精心工作使得该章表现不凡。干得好，Aravind！我希望我们能再次合作！

页 III: [16] 删除的内容 公共	zhukang	2001-10-26 15:33:00
页 III: [16] 删除的内容 语言	zhukang	2001-10-26 15:33:00
页 III: [16] 删除的内容 环境	zhukang	2001-10-26 15:33:00
页 III: [16] 删除的内容 及	zhukang	2001-10-18 9:49:00
页 III: [17] 删除的内容 公共	zhukang	2001-10-26 15:33:00
页 III: [17] 删除的内容 公共级别上的	zhukang	2001-10-18 9:49:00
页 III: [17] 删除的内容 小	zhukang	2001-10-18 9:50:00
页 III: [17] 删除的内容 入门障碍	zhukang	2001-10-18 9:50:00
页 III: [17] 删除的内容 一个	zhukang	2001-10-18 9:50:00
页 III: [17] 删除的内容 俱乐部	zhukang	2001-10-18 9:50:00
页 III: [17] 删除的内容 这个俱乐部的每个成员都享有双重权利	zhukang	2001-10-18 9:50:00
页 III: [17] 删除的内容 丰富	zhukang	2001-10-18 9:50:00
页 III: [17] 删除的内容 它	zhukang	2001-10-18 9:51:00

页 III: [17] 删除的内容 的	zhukang	2001-10-23 9:37:00
页 III: [18] 删除的内容 丰富	zhukang	2001-10-18 9:51:00
页 III: [18] 删除的内容 集	zhukang	2001-10-18 9:51:00
页 III: [19] 删除的内容 管理	zhukang	2001-10-18 9:51:00
页 III: [19] 删除的内容 s	zhukang	2001-10-18 9:51:00
页 III: [20] 删除的内容 的基于 XML 的	zhukang	2001-10-18 9:51:00
页 III: [20] 删除的内容 讯	zhukang	2001-10-18 9:51:00
页 III: [20] 删除的内容 尽管	zhukang	2001-10-18 9:51:00
页 III: [20] 删除的内容 是独立	zhukang	2001-10-18 9:52:00
页 III: [20] 删除的内容 的	zhukang	2001-10-18 9:52:00
页 III: [21] 删除的内容	zhukang	2001-8-16 11:55:00
页 III: [21] 删除的内容 仔	zhukang	2001-10-18 9:52:00
页 III: [21] 删除的内容 花	zhukang	2001-10-18 9:53:00
页 III: [21] 删除的内容 了不少钱	zhukang	2001-10-26 15:33:00

页 III: [21] 删除的内容
简单

zhukang

2001-10-18 9:53:00

目 录

第 部分 基础篇	1	3.1.1 选择编辑器	27
第 1 章 面向对象编程的基础知识	1	3.1.2 “Hello, World”	29
1.1 一切都是“对象”	2	3.1.3 使用命令行编译器	30
1.1.1 对象和类	5	3.1.4 运行应用程序	31
1.1.2 实例化	6	3.2 代码详解	32
1.2 面向对象编程语言的三大原则	7	3.2.1 “直达式”编程	32
1.2.1 封装	8	3.2.2 类和成员	32
1.2.2 继承	10	3.2.3 Main 方法	33
1.2.3 多态性	13	3.2.4 System.Console.WriteLine 方法	33
1.3 本章小结	16	3.2.5 名称空间和指示符 using	33
第 2 章 Microsoft .NET 简介	17	3.2.6 骨架代码	35
2.1 Microsoft .NET 平台	17	3.3 出错情况	36
2.2 .NET 框架	18	3.4 ILDASM 研究	37
2.2.1 Windows DNA 和 .NET	18	3.5 C#编程原则	41
2.2.2 公共语言运行环境	18	3.5.1 何时定义您自己的 名称空间	41
2.2.3 .NET 框架类库	19	3.5.2 命名原则	41
2.2.4 Microsoft 中间语言 和 JITters	21	3.5.3 命名规范标准	42
2.2.5 统一的类型系统	22	3.6 本章小结	44
2.2.6 元数据和反射	23	第 部分 C#类基础	47
2.2.7 安全问题	24	第 4 章 类型系统	47
2.2.8 部署	24	4.1 一切都是对象	47
2.2.9 与非控代码的互用性	25	4.2 数值类型和引用类型	48
2.3 本章小结	26	4.2.1 数值类型	48
第 3 章 C#应用程序简介	27	4.2.2 引用类型	48
3.1 编写第一个 C#应用程序	27	4.3 装箱与开箱	49