

C#与.NET 4高级程序设计（第5版）



[C#与.NET 4高级程序设计（第5版）_下载链接1](#)

著者:[美] Andrew Troelsen

出版者:人民邮电出版社

出版时间:2011-4

装帧:平装

isbn:9787115250322

C#语言作为.NET平台上的第一语言，已经成为目前功能最强大的通用语言之一。

本书是被誉为“C#圣经”的经典巨著，因语言生动流畅、剖析深入、涵盖全面而广受推崇，畅销不衰，曾经获得Referenceware编程图书大奖，并入选Jolt大奖提名。书中

探讨了C#语言和.NET平台的各种特性，包括重载运算符、指针、泛型等高级功能和CIL、远程处理、Windows Forms、ASP.NET、ADO.NET等技术，不少概念都通过IL代码透视其背后的本质，使你知其然，更知其所以然。新版更透彻阐述了C# 2010和.NET 4新功能，包括动态语言运行时（DLR）、任务并行库（TPL，包括PLINQ）、ADO.NET 实体框架（包括LINQ to EF）、扩展的WPF API，以及改进的COM互操作。

与同类图书不同，全书由世界级C#专家Andrew Troelsen以一人之力完成，因此写作思路和布局谋篇都独具匠心。中文版由国内多位微软MVP联手译出，强大的译者阵容有力地保证了这部权威著作原汁原味的重现。

作者介绍:

Andrew Troelsen 世界级C#专家，微软Visual C# MVP。他是著名的微软技术咨询企业Intertech的合伙人和副总裁，该公司的客户包括微软、霍尼韦尔、美国宇航局等。他曾为MSDN网站和MacTech网站撰写了有关各种操作系统平台上.NET技术的文章，并经常在业界主要技术会议上发表演讲和开设技术讲座。除本书外，他还撰写了COM and .NET Interoperability和Visual Basic .NET and the .NET Platform: An Advanced Guide等十多部.NET技术方面的著作。

目录: 第一部分 C#与.NET平台

第1章 .NET之道 2

1.1 .NET之前的世界 2

1.2 .NET解决方案 5

1.3 .NET平台构造块（CLR、CTS和CLS）

简介 5

1.4 其他支持.NET的编程语言 7

1.5 .NET程序集概览 9

1.6 CTS 14

1.7 CLS 17

1.8 CLR 18

1.9 程序集/命名空间/类型的区别 19

1.10 使用ildasm.exe探索程序集 24

1.11 使用Reflector来查看程序集 26

1.12 部署.NET运行库 27

1.13 .NET的平台无关性 27

1.14 小结 28

第2章 构建C#应用程序 30

2.1 .NET Framework 4 SDK的作用 30

2.2 用csc.exe构建C#应用程序 31

2.3 使用Notepad++构建.NET应用程序 36

2.4 使用SharpDevelop构建.NET应用程序 36

2.5 使用Visual C# 2010 Express构建.NET

应用程序 38

2.6 使用Visual Studio 2010构建.NET应用程序 40

2.7 小结 50

第二部分 C#核心编程结构

第3章 C#核心编程结构 I 52

3.1 一个简单的C#程序 52

3.2 有趣的题外话：System.Environment类的其他成员 57

- 3.3 System.Console类 58
- 3.4 系统数据类型和C#简化符号 62
- 3.5 使用字符串数据 70
- 3.6 窄化和宽化数据类型转换 76
- 3.7 隐式类型本地变量 81
- 3.8 C#迭代结构 84
- 3.9 条件结构和关系/相等操作符 86
- 3.10 小结 89
- 第4章 C#核心编程结构 II 90
- 4.1 方法和参数修饰符 90
- 4.2 C#中的数组操作 99
- 4.3 枚举类型 105
- 4.4 结构类型 109
- 4.5 值类型和引用类型 112
- 4.6 C#可空类型 118
- 4.7 小结 121
- 第5章 定义封装的类类型 122
- 5.1 C#类类型 122
- 5.2 构造函数 125
- 5.3 this关键字的作用 128
- 5.4 static关键字 133
- 5.5 定义OOP的支柱 139
- 5.6 C#访问修饰符 142
- 5.7 第一个支柱：C#的封装服务 144
- 5.8 自动属性 154
- 5.9 对象初始化器语法 157
- 5.10 常量数据 160
- 5.11 分部类型 162
- 5.12 小结 163
- 第6章 继承和多态 164
- 6.1 继承的基本机制 164
- 6.2 回顾Visual Studio类关系图 168
- 6.3 OOP的第二个支柱：继承 169
- 6.4 包含/委托编程 174
- 6.5 OOP的第三个支柱：C#的多态支持 176
- 6.6 基类/派生类的转换规则 187
- 6.7 超级父类：System.Object 189
- 6.8 小结 195
- 第7章 结构化异常处理 196
- 7.1 错误、bug与异常 196
- 7.2 .NET异常处理的作用 197
- 7.3 最简单的例子 199
- 7.4 配置异常的状态 202
- 7.5 系统级异常（System.System-Exception） 206
- 7.6 应用程序级异常（System.Applica-tionException） 207
- 7.7 处理多个异常 210
- 7.8 谁在引发什么异常 215
- 7.9 未处理异常的后果 215
- 7.10 使用Visual Studio调试未处理的异常 216
- 7.11 损坏状态异常简介 217
- 7.12 小结 218
- 第8章 对象的生命周期 219
- 8.1 类、对象和引用 219
- 8.2 对象生命周期的基础 220

- 8.3 应用程序根的作用 223
- 8.4 对象的代 224
- 8.5 .NET 1.0 至.NET 3.5的并发垃圾回收 225
- 8.6 .NET 4下的后台垃圾回收 225
- 8.7 System.GC类型 225
- 8.8 构建可终结对象 229
- 8.9 构建可处置对象 231
- 8.10 构建可终结类型和可处置类型 234
- 8.11 延迟对象实例化 237
- 8.12 小结 240
- 第三部分 高级C#编程结构
- 第9章 接口 242
 - 9.1 接口类型 242
 - 9.2 定义自定义接口 245
 - 9.3 实现接口 247
 - 9.4 在对象级别调用接口成员 248
 - 9.5 接口作为参数 250
 - 9.6 接口作为返回值 252
 - 9.7 接口类型数组 253
 - 9.8 使用Visual Studio 2010实现接口 253
 - 9.9 通过显式接口实现解决命名冲突 254
 - 9.10 设计接口层次结构 257
 - 9.11 构建可枚举类型 (IEnumerable和IEnumerator) 260
 - 9.12 构建可克隆的对象 (ICloneable) 265
 - 9.13 构建可比较的对象 (Comparable) 269
 - 9.14 小结 273
- 第10章 泛型 274
 - 10.1 非泛型集合 274
 - 10.2 泛型类型参数的作用 281
 - 10.3 System.Collections.Generic命名空间 285
 - 10.4 创建自定义泛型方法 292
 - 10.5 创建自定义泛型结构和类 294
 - 10.6 类型参数的约束 297
 - 10.7 小结 300
- 第11章 委托、事件和Lambda 301
 - 11.1 .NET委托类型 301
 - 11.2 在C#中定义委托类型 302
 - 11.3 System.MulticastDelegate与System.Delegate基类 304
 - 11.4 最简单的委托示例 305
 - 11.5 使用委托发送对象状态通知 307
 - 11.6 方法组转换语法 313
 - 11.7 委托协变 314
 - 11.8 泛型委托 316
 - 11.9 C#事件 318
 - 11.10 C#匿名方法 325
 - 11.11 Lambda表达式 328
 - 11.12 小结 333
- 第12章 高级C#语言特性 335
 - 12.1 索引器方法 335
 - 12.2 操作符重载 339
 - 12.3 自定义类型转换 347
 - 12.4 扩展方法 352
 - 12.5 分部方法 360
 - 12.6 匿名类型 362

- 12.7 指针类型 367
- 12.8 小结 373
- 第13章 LINQ to Object 374
 - 13.1 LINQ特有的编程结构 374
 - 13.2 LINQ的作用 377
 - 13.3 将LINQ查询应用于原始数组 379
 - 13.4 返回LINQ查询的结果 385
 - 13.5 将LINQ查询应用到集合对象 387
 - 13.6 C#LINQ查询操作符 389
 - 13.7 LINQ查询语句的内部表示 396
 - 13.8 小结 400
- 第四部分 用.NET程序集编程
- 第14章 .NET程序集入门 402
 - 14.1 定义自定义命名空间 402
 - 14.2 .NET程序集的作用 407
 - 14.3 .NET程序集的格式 409
 - 14.4 构建和使用单文件程序集 412
 - 14.5 构建和使用多文件程序集 421
 - 14.6 私有程序集 423
 - 14.7 共享程序集 427
 - 14.8 使用共享程序集 433
 - 14.9 配置共享程序集 435
 - 14.10 发行者策略程序集 439
 - 14.11 <codeBase>元素 440
 - 14.12 System.Configuration命名空间 441
 - 14.13 小结 442
- 第15章 类型反射、晚期绑定和基于特性的编程 443
 - 15.1 类型元数据的必要性 443
 - 15.2 反射 447
 - 15.3 构建自定义的元数据查看器 450
 - 15.4 动态加载程序集 454
 - 15.5 反射共享程序集 457
 - 15.6 晚期绑定 458
 - 15.7 .NET特性的作用 461
 - 15.8 构建自定义特性 465
 - 15.9 程序集级别（和模块级别）特性 467
 - 15.10 使用早期绑定反射特性 469
 - 15.11 使用晚期绑定反射特性 470
 - 15.12 反射、晚期绑定和自定义特性的使用背景 471
 - 15.13 构建可扩展的应用程序 472
 - 15.14 小结 477
- 第16章 进程、应用程序域和对象上下文 478
 - 16.1 Windows进程的作用 478
 - 16.2 .NET平台下与进程进行交互 480
 - 16.3 .NET应用程序域 488
 - 16.4 与默认应用程序域进行交互 489
 - 16.5 创建新的应用程序域 492
 - 16.6 对象上下文边界 495
 - 16.7 进程、应用程序域和上下文小结 498
 - 16.8 小结 499
- 第17章 CIL和动态程序集的作用 500
 - 17.1 学习CIL语法的原因 500
 - 17.2 CIL指令、特性和操作码 501
 - 17.3 入栈和出栈：CIL基于栈的本质 502

- 17.4 正反向工程 504
- 17.5 CIL指令和特性 510
- 17.6 .NET基础类库、C#和CIL数据类型的映射 515
- 17.7 在CIL中定义类型成员 516
- 17.8 剖析CIL操作码 518
- 17.9 使用CIL构建.NET程序集 522
- 17.10 动态程序集 526
- 17.11 小结 534
- 第18章 动态类型和动态语言运行时 535
- 18.1 dynamic关键字的作用 535
- 18.2 DLR的作用 540
- 18.3 使用动态类型简化后期绑定调用 542
- 18.4 使用动态数据简化COM互操作 544
- 18.5 使用C# 2010的特性进行COM互操作 548
- 18.6 小结 552
- 第五部分 .NET基础类库
- 第19章 构建多线程应用程序 554
- 19.1 进程、应用程序域、上下文及线程之间的关系 554
- 19.2 .NET委托的简短回顾 556
- 19.3 委托的异步性 557
- 19.4 异步调用方法 559
- 19.5 System.Threading命名空间 564
- 19.6 System.Threading.Thread类 564
- 19.7 以编程方式创建次线程 567
- 19.8 并发问题 572
- 19.9 使用Timer Callback编程 578
- 19.10 CLR线程池 579
- 19.11 .NET平台下的并行编程 581
- 19.12 并行LINQ查询 (PLINQ) 588
- 19.13 小结 590
- 第20章 文件输入输出和对象序列化 591
- 20.1 研究System.IO命名空间 591
- 20.2 DirectoryInfo和FileInfo类型 592
- 20.3 使用DirectoryInfo类型 593
- 20.4 使用Directory类型 596
- 20.5 使用DriveInfo类类型 597
- 20.6 使用FileInfo类 598
- 20.7 使用File类型 601
- 20.8 Stream抽象类 603
- 20.9 使用StreamWriter和StreamReader类型 605
- 20.10 使用StringWriter和StringReader类型 608
- 20.11 使用BinaryWriter和Binary-Reader 609
- 20.12 以编程方式“观察”文件 610
- 20.13 对象序列化 612
- 20.14 为序列化配置对象 614
- 20.15 选择序列化格式化程序 616
- 20.16 使用BinaryFormatter序列化对象 618
- 20.17 使用SoapFormatter序列化对象 620
- 20.18 使用XmlSerializer序列化对象 621
- 20.19 序列化对象集合 623
- 20.20 自定义Soap/Binary序列化过程 624
- 20.21 小结 628
- 第21章 ADO.NET之一：连接层 630
- 21.1 ADO.NET的宏观定义 630

21.2 ADO.NET数据提供程序	632
21.3 其他的ADO.NET命名空间	634
21.4 System.Data命名空间的类型	635
21.5 使用接口的抽象数据提供程序	639
21.6 创建AutoLot数据库	641
21.7 ADO.NET数据提供程序工厂模型	645
21.8 ADO.NET的连接层	650
21.9 使用数据读取器	654
21.10 构建可重用的数据访问库	656
21.11 创建控制台UI前端	664
21.12 数据库事务	669
21.13 小结	673
第22章 ADO.NET之二：断开连接层	674
22.1 ADO.NET断开连接层	674
22.2 DataSet的作用	675
22.3 使用DataColumn	677
22.4 使用DataRow	679
22.5 使用DataTable	682
22.6 将DataTable对象绑定到用户界面	687
22.7 使用数据适配器	695
22.8 向AutoLotDAL.dll添加断开连接功能	698
22.9 多表DataSet对象和数据关系	701
22.10 Windows Forms数据库设计器工具	706
22.11 将强类型的数据库代码隔离到类库中	713
22.12 LINQ to DataSet	717
22.13 小结	722
第23章 ADO.NET之三：Entity Framework	723
23.1 Entity Framework的作用	723
23.2 创建和分析EDM	730
23.3 对概念模型进行编程	739
23.4 AutoLotDAL 4.0版，加入实体	744
23.5 将数据实体绑定到Windows Forms GUI	748
23.6 小结	752
第24章 LINQ to XML简介	753
24.1 两个XML API的故事	753
24.2 System.Xml.Linq命名空间的成员	756
24.3 使用XElement和XDocument	759
24.4 在内存中操作XML文档	762
24.5 小结	766
第25章 WCF	767
25.1 各种分布式计算API	767
25.2 WCF的作用	772
25.3 WCF核心程序集	774
25.4 Visual Studio WCF项目模板	775
25.5 WCF应用程序的基本构成	776
25.6 WCF的ABC	777
25.7 构建WCF服务	781
25.8 承载WCF服务	784
25.9 构建WCF客户端应用程序	791
25.10 用WCF 4.0简化配置设置	795
25.11 使用WCF服务库项目模板	800
25.12 以Windows服务承载WCF服务	803
25.13 从客户端异步调用服务	807
25.14 定义WCF数据契约	809

- 25.15 小结 813
- 第26章 Windows Workflow Foundation
 - 4.0简介 814
 - 26.1 定义业务流程 814
 - 26.2 构建简单的工作流 815
 - 26.3 WF 4.0运行时 818
 - 26.4 检查WF 4.0中的活动 823
 - 26.5 构建流程图工作流 825
 - 26.6 用专门的库隔离工作流 833
 - 26.7 使用工作流库 840
 - 26.8 小结 842
- 第六部分 使用WPF构建桌面用户界面
- 第27章 WPF和XAML 844
 - 27.1 WPF背后的动机 844
 - 27.2 各种形式的WPF应用程序 846
 - 27.3 WPF程序集 850
 - 27.4 创建不使用XAML的WPF应用程序 856
 - 27.5 仅使用XAML构建WPF应用程序 863
 - 27.6 将标记转换为.NET程序集 867
 - 27.7 WPF XAML语法 871
 - 27.8 使用代码隐藏文件构建WPF应用程序 879
 - 27.9 使用Visual Studio 2010构建WPF应用程序 881
 - 27.10 小结 889
- 第28章 使用WPF控件编程 890
 - 28.1 WPF核心控件概述 890
 - 28.2 使用面板控制内容布局 894
 - 28.3 使用嵌套面板构建窗口框架 903
 - 28.4 WPF控件命令 907
 - 28.5 使用Expression Blend构建WPF用户界面 912
 - 28.6 构建Ink API选项卡 919
 - 28.7 Documents API 928
 - 28.8 构建Documents选项卡 929
 - 28.9 WPF数据绑定模型 934
 - 28.10 小结 941
- 第29章 WPF图形呈现服务 942
 - 29.1 理解WPF的图形呈现服务 942
 - 29.2 使用形状呈现图形数据 943
 - 29.3 WPF画刷和画笔 952
 - 29.4 图形变换 955
 - 29.5 使用Expression Blend处理形状 958
 - 29.6 使用绘图和几何图形呈现图形数据 962
 - 29.7 使用Expression Design生成复杂的向量图形 965
 - 29.8 使用可视化层呈现图形数据 967
 - 29.9 小结 972
- 第30章 WPF资源、动画和样式 973
 - 30.1 理解WPF资源系统 973
 - 30.2 使用对象（逻辑）资源 978
 - 30.3 理解WPF动画服务 988
 - 30.4 用XAML创建动画 992
 - 30.5 WPF样式的作用 994
 - 30.6 使用Expression Blend生成样式 1001
 - 30.7 小结 1003
- 第31章 WPF控件模板和用户控件 1004
 - 31.1 依赖属性的作用 1004

- 31.2 构建自定义依赖属性 1008
- 31.3 路由事件 1013
- 31.4 逻辑树、可视树和默认模板 1016
- 31.5 在Visual Studio 2010中构建自定义控件模板 1022
- 31.6 使用Blend构建自定义UserControl 1028
- 31.7 创建WPF应用程序Jackpot Deluxe 1035
- 31.8 小结 1043
- 第七部分 使用ASP.NET构建Web应用程序
- 第32章 构建ASP.NET网页 1046
 - 32.1 HTTP的作用 1046
 - 32.2 Web应用程序和Web服务 1047
 - 32.3 HTML的作用 1048
 - 32.4 客户端脚本的作用 1053
 - 32.5 回发到Web服务器 1054
 - 32.6 ASP.NET API的特性 1055
 - 32.7 构建单个文件的ASP.NET网页 1058
 - 32.8 使用代码文件构建ASP.NET Web页面 1066
 - 32.9 ASP.NET Web Site和ASP.NET Web Application 1070
 - 32.10 ASP.NET网站目录结构 1072
 - 32.11 页面类型的继承链 1073
 - 32.12 与传入的HTTP请求交互 1074
 - 32.13 与输出HTTP响应交互 1077
 - 32.14 ASP.NET网页的生命周期 1078
 - 32.15 Web.config文件的作用 1081
 - 32.16 小结 1082
- 第33章 ASP.NET Web控件、母版页和主题 1083
 - 33.1 Web控件的本质 1083
 - 33.2 Control和WebControl基类 1085
 - 33.3 ASP.NET Web控件的类别 1089
 - 33.4 构建ASP.NET汽车网站 1091
 - 33.5 验证控件的作用 1104
 - 33.6 使用主题 1110
 - 33.7 小结 1114
- 第34章 ASP.NET状态管理技术 1115
 - 34.1 状态问题 1115
 - 34.2 ASP.NET状态管理技术 1117
 - 34.3 ASP.NET视图状态的作用 1117
 - 34.4 Global.asax文件的作用 1120
 - 34.5 应用程序状态与会话状态的差别 1123
 - 34.6 使用应用程序缓存 1127
 - 34.7 维护会话数据 1131
 - 34.8 cookie 1134
 - 34.9 <sessionState>元素的作用 1136
 - 34.10 ASP.NET用户配置API 1137
 - 34.11 小结 1143
- 第八部分 附录
- 附录A Windows Forms编程 1146
 - A.1 Windows Forms命名空间 1146
 - A.2 创建一个简单的Windows Forms程序 1147
 - A.3 Visual Studio Windows Forms项目模板 1152
 - A.4 剖析Form 1157
 - A.5 响应鼠标活动 1163
 - A.6 设计对话框 1166
 - A.7 通过GDI+呈现图形数据 1171

A.8 创建一个完整的Windows Forms应用程序 1175
A.9 小结 1182
附录B 使用Mono进行平台无关的.NET开发 1183
B.1 .NET的平台无关性 1183
B.2 获取和安装Mono 1186
B.3 Mono开发语言 1188
B.4 兼容Microsoft的Mono开发工具 1189
B.5 使用Mono创建.NET 应用程序 1190
B.6 谁在使用Mono 1196
B.7 推荐学习 1196
B.8 小结 1197
• • • • • [\(收起\)](#)

[C#与.NET 4高级程序设计（第5版）_下载链接1_](#)

标签

C

#.NET

编程

计算机

程序设计

编程语言

计算机科学

软件-编程-工程

评论

1、已经是4.5了，有几个新的语法糖。2、如果你不是搞计算机语言学的，还是不要学P

神.....每天看那么多书....

前11章大概读过

.....

C#读物。。。很厚，但是基本都能点到。不针对语言来说，此书不错!update:03xx//

很让我惊艳的一本书，从浅入深讲的明明白白的。
另外，这书真贵！！！不过2星是给翻译的。。。不得不说。。。对照着英文看了一章。。。我就看不下去翻译的了。。。

不愧是C#圣经，详尽，通俗，适度深入地介绍C#的各种语法特性，入门必读。。。

内容比较经典，就是有些例子有点罗嗦，没用字段太多。

看着看着就会了

不给5分说不过去啊。

大致过了一遍语法方面，和java很像；说明讲解很详细

看完图文教程的后话了

讲解清晰明了,暂告一段落

中级进阶书

复习用，一千来页分20多天逐页读完，不熟悉的部分偶尔敲敲代码，以后当字典翻。感觉对于大多数人来说有些部分很少会用到，AT的C#教材，只要翻译差不多水准相当可以，大而全。

C#圣经

更深入…

大而全，买来就翻了一遍，基本可以当个目录来看

经典，比看微软的帮助省事多了。

适合.NET平台初学者学习。

当初学习 C# 时最爱的一本书。

[C#与.NET 4高级程序设计（第5版）_下载链接1_](#)

书评

★ C# and the .NET Platform (June 20, 2001) ★ C# and the .NET Platform, 2nd Edition (June 5, 2003) ★ Pro C# 2005 and the .NET 2.0 Platform, 3rd Edition (September 14, 2005) ★ Pro C# with .NET 3.0, Special Edition (February 5, 2007) -- C#与.NET 3.0高级...

详略得当 .NET 3.0-4.0带来的全新组件几乎全部都有讲解 不愧为 “.net圣经”
推荐新手可以从后往前读 WPF WCF WF ADO.NET和EF的内容 这样可以增进阅读趣味
开发人员可以精读有关GC/并发/LINQ/IL方面的内容，查缺补漏

个人感觉C#4.0的新功能都介绍到了，甚至包括C#3.5,C#3.0中一些被遗忘的，平时不太用的功能都有写。
希望把项目从VS2008升级到VS2010的同学必读此书，唯一的遗憾就是书太厚，有很多内容第4版已经介绍过了。

很不好意思的说，我只是一名大一的学生，刚看完一本WPF的书，是C#里的纯正菜鸟级别，所以有什么失误之处还请各位大侠原谅。我觉得这本书是很好很好的，很全面，但似乎过于全面了，各种语言一起上，搞得我等没什么编程语言的小辈无言以对了。但我还是由衷的感谢作者，给了我很多...

这本书很好啊，通读一遍对整个.net平台会有一个整体的了解。相当不错。目前在看中文版的，等看完再看看英文的。

很不错的一本书，对初学者来很有意义，虽说部分内容有点不好理解，但是不理解可以先跳过去啦，慢慢看，收获一定不会少的~

在介绍语法的时候，能从字节码级解释，很透彻，不错。
如果你对Java的字节码有所了解，那么看起来相当容易上手。

这本书是本人学习C#与.NET编程开发的必备参考书。通过它可以迅速提高。这本书较厚，C# 2005 和 .NET 3.0 的内容全部覆盖并都有实例讲解。给本人印象较深的是IL方面的知识内容，这是.NET系开发工程师通往高级软工所必须掌握的技能之一！在这里，本人推荐款很好...

不用说了，我正在学习这本书。对于我来说简直如获至宝。其经典不亚于《C#入门经典》，但是翻译还是有些错误。别的不说，质量说了算。建议欲买此书的人先去书店看看，毕竟这本书还是价钱不菲。也可在网上免费试读，下面是试读网址：<http://book.csdn.net/bookfiles/650/>

对于稍微有点.NET基础的读者，通读全书基本不会发现有什么难度，没有任何难理解的地方。

对于.NET介绍的很全面，但是每一点都说的比较肤浅，刚刚说到稍微深入的地方时戛然而止，读起来总是感觉差一口气，意犹未尽的感觉。

适合初级学习，对.NET有一个总体把握和理解。也可以当...

Purchase this book and you'll get the free, fully searchable eBook--a \$30 value! (Details are printed inside the book.)

C# 2005 has changed the .NET 2.0 landscape. To name just a few improvements, C# now supports generics, partial types and nullab...

[illegible]

看第二天了，发现C#很多地方跟Java差不多。不过多了好多比较古怪的写法，像方法重载，为啥就非得整个virtual和override，感觉像脱了裤子放P。

C#与.NET 4高级程序设计 (第5版) 下载链接1