

<<C#捷径教程>>

图书基本信息

书名：<<C#捷径教程>>

13位ISBN编号：9787115192585

10位ISBN编号：7115192588

出版时间：2009-2

出版单位：人民邮电出版社

作者：纳什

页数：444

译者：刘新军

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<C#捷径教程>>

前言

因为在学校微软技术俱乐部活动的缘故，C#Beta版一经问世我就从微软高校关系部拿到了试用版，并且在微软的支持下俱乐部的一帮人一起做了几个小项目。

2003年10月，微软全球副总裁JawadKhaki来北邮做讲座，之后有人去索取签名，一个哥们居然拿着一本《Java编程思想》上去。

结果Jawad大笔一挥在扉页上写下了几个大字“c# is a better Java”。

这句话正是我当时对c#的第一印象，可能也正反映了微软对C#的期望和当时业界对C#的评价。

当然微软的雄心远不止于此，.NET才是它的战略发展方向，C#只是推动这个发展的一个独门武器。

几年来，微软投入了大量的资源，把这款武器打造得越来越锋利。

随着泛型、匿名方法、迭代器、分部类型 / 分部方法、LINQ、匿名类型、对象 / 集合初始化器、扩展方法、Lambda表达式等一系列特性的逐步引入，C#俨然成为了特性最丰富的编程语言。

丰富的特性带来了强大的功能和高效的开发效率，但与之相伴的副作用之一就是加陡了学习曲线。

一件再好的武器，如果你不能透彻掌握其奥妙玄机，后果将轻则浪费宝物，重则误伤自己。

因此，除了会使手中的利器，找到一位好师父、觅得一本好宝典同样非常重要。

目前市面上有很多关于C#的图书，也有很多诸如《24小时学会XXX》、《轻松掌握XXX》的图书。

刚开始，我对出版社定下的本书中文书名——《C#捷径教程》颇有微词。

不过，仔细一想，“捷径”确实最贴切原书名中Accelerated的本意。

学习原本没有捷径可走，但要在有限的时间内取得最大的成效，还是有策略和方法的。

本书的“捷径”体现在以下两个方面。

1. 以相对短的篇幅浓缩了C#的几乎所有的精华，高屋建瓴，取精用弘。

本书没有纠缠于繁琐的语法细节（有任何C / C++ / Java语言背景的人都可以触类旁通），也没有涉及华丽的IDE操作与界面设计内容（可以通过Tutorial或Help更方便地获得），而是直接切入C#语言本身的精髓，让你最直接地感受并掌握最激动人心的特性。

2. 以简洁的语言与丰富的示例来讲解各个特性，一针见血，酣畅淋漓。

作者是位资深的程序员，因此本书的行文风格都是基于程序员的角度和思维方式，使你在阅读的过程中有一种心领神会的默契之感。

当然，翻译的过程没有捷径可走。

本书的面世是许多人帮助与支持的结果。

感谢编辑陈兴璐小姐。

这是第二次与她合作。

她一如既往地友善耐心、认真负责，保证了本书的按时按质完成。

感谢杨军、崔晓川、刘光强、李辛鹤、徐进在翻译过程中给予我的帮助，他们的专业水平和中肯建议，使翻译润色不少。

与他们的讨论交流，也让我受益匪浅。

感谢小顾同学，忍受了我长达半年深居简出的宅男生活：)。

由于时间仓促，水平有限，不足之处望各位同仁不吝赐教。

希望你在《C#捷径教程》中找到学习C#的捷径！

<<C#捷径教程>>

内容概要

《Accelerated C#2008 C#捷径教程》通过许多精彩的示例介绍了每个特性，深入浅出地讲解了C#语言的核心概念，以及如何聪明地应用C#的习惯用法和面向对象的设计模式来挖掘C#和CLR的能力。

C# 3.0 提供了很多强大的特性，通过使用lambda 表达式、扩展方法和语言集成查询(LINQ)，方便地引入了函数式编程，使C# 程序员如虎添翼。

《Accelerated C#2008 C#捷径教程》适合有一定编程经验的程序员阅读。

<<C#捷径教程>>

作者简介

TreyNash, 知名.NET技术专家。
现于领先的无线设备提供商CSR公司担任主任工程师。
负责蓝牙解决方案的开发。
在转向.NET平台之前。
他有丰富的COM, DCOM / ATL编程经验。

<<C#捷径教程>>

书籍目录

第1章 C#预览1.1 C#和C++的区别1.1.1 C#1.1.2 C++1.1.3 CLR垃圾回收1.2 C#程序示例1.3 C# 2.0扩展特性概述1.4 C# 3.0新特性概览1.5 小结第2章 C#和CLR2.1 CLR中的JIT编译器2.2 程序集及程序集加载器2.2.1 最小化程序的工作集2.2.2 给程序集命名2.2.3 加载程序集2.3 元数据2.4 交叉语言的兼容性2.5 小结第3章 C#语法概述3.1 C#是一门强类型的语言3.2 表达式3.3 语句和表达式3.4 类型和变量3.4.1 值类型3.4.2 引用类型3.4.3 默认变量初始化3.4.4 隐式类型化局部变量3.4.5 类型转换3.4.6 as和is操作符3.4.7 泛型3.5 命名空间3.5.1 定义命名空间3.5.2 使用命名空间3.6 控制流3.6.1 if-else、while、do-while和for3.6.2 switch3.6.3 foreach3.6.4 break、continue、goto、return和throw3.7 小结第4章 类、结构和对象4.1 类定义4.1.1 字段4.1.2 构造函数4.1.3 方法4.1.4 属性4.1.5 封装4.1.6 可访问性4.1.7 接口4.1.8 继承4.1.9 密封类4.1.10 抽象类4.1.11 嵌套类4.1.12 索引器4.1.13 分部类4.1.14 分部方法4.1.15 静态类4.1.16 保留的成员名字4.2 值类型定义4.2.1 构造函数4.2.2 this的含义4.2.3 终结器4.2.4 接口4.3 匿名类型4.4 对象初始化器4.5 装箱和拆箱4.5.1 什么时候发生装箱4.5.2 效率和混淆4.6 System.Object4.6.1 等同性及其意义4.6.2 IComparable接口4.7 创建对象4.7.1 new关键字4.7.2 字段初始化4.7.3 静态(类)构造函数4.7.4 实例构造函数和创建顺序4.8 销毁对象4.8.1 终结器4.8.2 确定性的析构4.8.3 异常处理4.9 可清除对象4.9.1 IDisposable接口4.9.2 using关键字4.10 方法参数类型4.10.1 值参数4.10.2 ref参数4.10.3 out参数4.10.4 参数数组4.11 方法重载4.12 继承和虚方法4.12.1 虚方法和抽象方法4.12.2 override和new方法4.12.3 密封方法4.12.4 关于C#虚方法再啰嗦几句4.13 继承, 包含和委托4.13.1 接口继承和类继承的选择4.13.2 委托和组合与继承4.14 小结第5章 接口和契约5.1 接口定义类型5.2 定义接口5.2.1 接口中可以有什么5.2.2 接口继承与成员隐藏5.3 实现接口5.3.1 隐式接口实现5.3.2 显式接口实现5.3.3 派生类中覆盖接口实现5.3.4 小心值类型实现接口的副作用5.4 接口成员匹配规则5.5 值类型的显示接口实现5.6 版本考虑5.7 契约5.7.1 类实现契约5.7.2 接口契约5.8 在接口和类之间选择5.9 小结第6章 重载操作符6.1 只因为:可以并不意味着应该6.2 重载操作符的类型和格式6.3 操作符不应该改变其操作数6.4 参数顺序有影响么6.5 重载加法运算符6.6 可重载的操作符6.6.1 比较操作符6.6.2 转换操作符6.6.3 布尔操作符6.7 小结第7章 异常处理和异常安全7.1 CLR如何对待异常7.2 C#里的异常处理机制7.2.1 抛出异常7.2.2 .NET 2.0开始的未处理异常的变化7.2.3 try语句语法预览7.2.4 重新抛出异常和转译异常7.2.5 finally代码块抛出的异常7.2.6 终结器抛出的异常7.2.7 静态构造函数抛出的异常7.3 谁应该处理异常7.4 避免使用异常来控制流程7.5 取得异常中立7.5.1 异常中立代码的基本结构7.5.2 受限执行区域7.5.3 临界终结器和SafeHandle7.6 创建定制的异常类7.7 使用分配的资源 and 异常7.8 提供回滚行为7.9 小结第8章 使用字符串8.1 字符串概述8.2 字符串字面量8.3 格式指定和全球化8.3.1 Object.ToString、IFormattable和CultureInfo8.3.2 创建和注册自定义CultureInfo类型8.3.3 格式化字符串8.3.4 Console.WriteLine和String.Format8.3.5 自定义类型的字符串格式化举例8.3.6 ICustomFormatter8.3.7 字符串比较8.4 处理来自外部的字符串8.5 StringBuilder8.6 使用正则表达式搜索字符串8.6.1 使用正则表达式搜索8.6.2 搜索和分组8.6.3 使用正则表达式替换文本8.6.4 正则表达式创建选项8.7 小结第9章 数组、容器类型和迭代器9.1 数组介绍9.1.1 隐式类型化数组9.1.2 类型的转换和协方差9.1.3 排序和搜索9.1.4 同步9.1.5 向量与数组9.2 多维矩形数组9.3 多维锯齿数组9.4 容器类型9.4.1 比较ICollectionT和ICollection9.4.2 容器同步9.4.3 列表9.4.4 字典9.4.5 集合9.4.6 System.Collections.ObjectModel9.4.7 效率9.5 IEnumerableT、IEnumeratorT、IEnumerable和IEnumerator9.6 迭代器9.7 容器初始化器9.8 小结第10章 委托, 匿名方法和事件10.1 委托概览10.2 委托的创建和使用10.2.1 单委托10.2.2 委托链10.2.3 迭代委托链10.2.4 非绑定(公开实例)的委托10.3 事件10.4 匿名方法10.4.1 注意捕获变量的使用10.4.2 匿名方法作为委托参数绑定器10.5 Strategy模式10.6 小结第11章 泛型11.1 泛型和C++模板之间的区别11.2 泛型的效率和类型安全11.3 泛型的类型定义和构造类型11.3.1 泛型类和结构11.3.2 泛型接口11.3.3 泛型方法11.3.4 泛型委托11.3.5 泛型转换11.3.6 默认值表达式11.3.7 Nullable类型11.3.8 构造类型访问权限控制11.3.9 泛型和继承11.4 约束11.5 泛型系统容器11.6 泛型系统接口11.7 精选的问题和解决方

<<C#捷径教程>>

案11.7.1 泛型类型中的转化和操作符11.7.2 动态地创建构造类型11.8 小结第12章 C#中的线程12.1 C#和.NET中的线程12.1.1 开始线程编程12.1.2 IOU模式和异步方法调用12.1.3 线程状态12.1.4 终止线程12.1.5 停止和唤醒休眠线程12.1.6 等待线程退出12.1.7 前台和后台线程12.1.8 线程本地存储12.1.9 非托管线程和COM套件如何适应12.2 线程间同步工作12.2.1 用Interlocked类实现轻量级的同步12.2.2 Monitor类12.2.3 锁对象12.2.4 信号量12.2.5 事件12.2.6 Win32的同步对象和WaitHandle12.3 使用线程池12.3.1 异步方法调用12.3.2 定时器12.4 小结第13章 C#规范形式探索13.1 引用类型的规范形式13.1.1 类默认是密封的13.1.2 使用非虚拟接口模式13.1.3 对象是否可克隆13.1.4 对象是否可清除13.1.5 对象需要终结器吗13.1.6 对象相等意味着什么13.1.7 如果重写了Equals方法,那么也应该重写GetHashCode方法13.1.8 对象支持排序吗13.1.9 对象需要形式化显示吗13.1.10 对象可以被转换吗13.1.11 在所有情况下都保证类型安全13.1.12 使用非可变的引用类型13.2 值类型的规范形式13.2.1 为了获得更好的性能而重写Equals方法13.2.2 值类型需要支持接口吗13.2.3 将接口成员和派生方法实现为类型安全的形式13.3 小结13.3.1 引用类型的检查单13.3.2 值类型的检查单第14章 扩展方法14.1 扩展方法介绍14.1.1 编译器如何发现扩展方法14.1.2 探究内部实现14.1.3 代码易读性与代码易懂性14.2 使用建议14.2.1 考虑扩展方法优先于继承14.2.2 分离的命名空间中的隔离扩展方法14.2.3 修改一个类型的契约可能会打破扩展方法14.3 转换14.4 链式操作14.5 自定义迭代器14.6 访问者模式14.7 小结第15章 lambda表达式15.1 lambda表达式介绍15.1.1 lambda表达式15.1.2 lambda语句15.2 表达式树15.2.1 对表达式的操作15.2.2 函数的数据表现15.3 lambda表达式的有益应用15.3.1 迭代器和生成器重访问15.3.2 闭包(变量捕获)和缓存15.3.3 currying15.3.4 匿名递归15.4 小结第16章 LINQ:语言集成查询 16.1 连接数据的桥梁 16.1.1 查询表达式 16.1.2 再谈扩展方法和lambda表达式16.2 标准查询操作符16.3 C#查询关键字 16.3.1 from子句和范围变量 16.3.2 join子句 16.3.3 where子句和过滤器 16.3.4 orderby子句 16.3.5 select子句和投影 16.3.6 let子句 16.3.7 group子句 16.3.8 into子句和持续性 16.4 偷懒的好处 16.4.1 C#迭代器鼓励懒惰 16.4.2 不能偷懒 16.4.3 立即执行查询 16.4.4 再谈表达式树16.5 函数式编程中的技术16.5.1 自定义标准查询操作符和延迟求值 16.5.2 替换foreach语句16.6 小结

<<C#捷径教程>>

章节摘录

第1章 C#预览 1.4 C# 3.0新特性概览 C# 3.0包括一些很好的新特性。大部分新特性是支持语言集成查询（Language Integrated Query, LINQ）的基石。然而，即使在LINQ的范围外单独使用，它们也非常有用。其中的大部分特性使程序员使用函数式编程技术更容易。

通过使用新的关键字var，C#现在支持隐式的类型化局部变量。需要注意的是，这些变量不是无类型的，它们的类型是编译时推断出的。这点在第3章作介绍。

你是否曾想创建一个简单类型来保存有关数据，但又烦恼于需要创建整个新类？在很多情况下，新支持的匿名类型帮你减轻了这个负担。使用匿名类型可以在一个复合语句中定义并实例化一个类型。匿名类型将在第4章进行介绍。

自动实现的属性（auto-implemented property）是另一个有用的新特性，它为我们省去了打字的麻烦，减少了引入bug的可能性。多少次你曾为保存一点数据而声明一个类，然后疲于为这些数据创建属性访问函数？毕竟，这样做符合良好封装的实践。谢天谢地，自动实现的属性极大地减少了定义属性的输入量。第4章将介绍自动实现属性的技术细节。

<<C#捷径教程>>

媒体关注与评论

“作者的C#造诣极深，而且善于引导读者在轻松愉快的氛围中学习C#的特性。
通过学习此书内容，你的C#水平必将今非昔比。

” ——Wes dyer，微软C# 开发小组成员
“我读过不下100本.NET方面的书，自己也写过几本书，但是此书精彩绝伦，远远超出我的预期。
强烈推荐！”

” ——William Ryan，微软MVP

<<C#捷径教程>>

编辑推荐

如果你已经有一定C#或者其他语言的编程经验，就完全不用去读那些肤浅的入门书或者冗长繁复的大部头。

《Accelerated C#2008 C#捷径教程》将为你提供一条快速掌握C#的捷径。

作者在不大的篇幅里容纳了丰富扎实的内容，不仅全面涵盖了C#的语言特性，包括3.0引入的Lambda表达式、扩展方法和LINQ等，总结了正确使用C#的习惯用法和设计模式。

而且深入探讨了CLR。

书中大量富于启发性的实例和经验之谈。

更能让你的软件开发水平得到质的飞跃。

好评如潮的C#实战图书 汲取.NET技术精髓的捷径 专章讲述习惯用法与设计模式

<<C#捷径教程>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>