HZ BOOKS
华章科技

张子阳 著

.NET之美

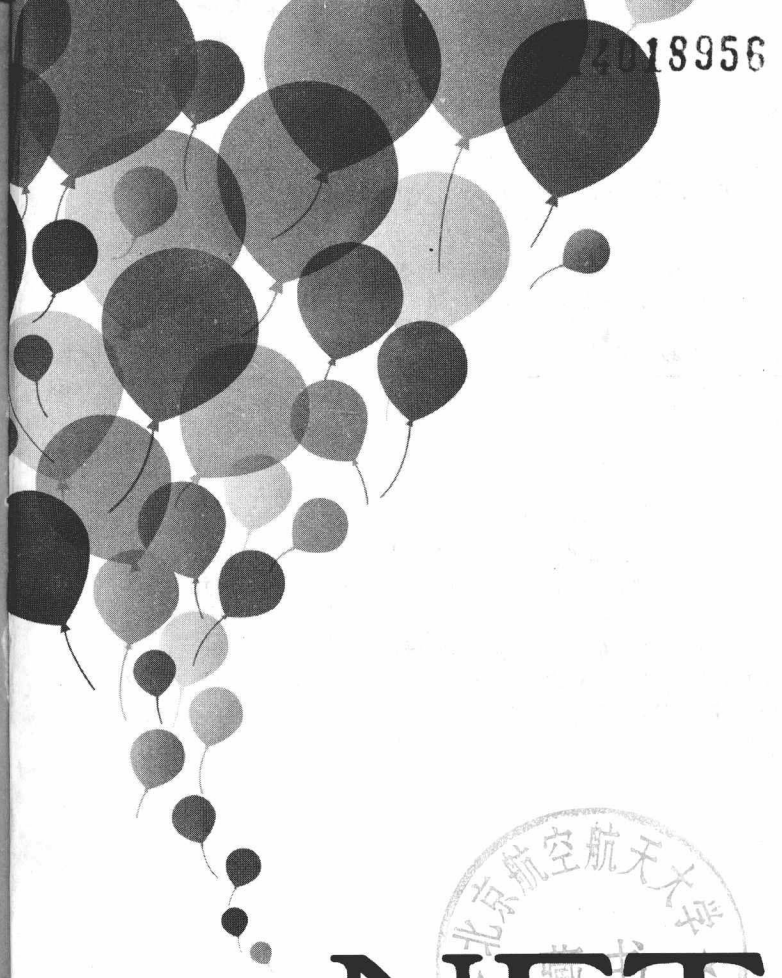
.NET关键技术深入解析

Beautiful .NET: In-depth Analysis of Key Points of .NET

资深.NET技术专家多年开发经验结晶，悉数对C#和.NET中实用的、关键的和难以理解的知识点进行了深入解析，尽显.NET之美

每个知识点都辅之以精心设计的案例，理论与实践并重，是修炼为高级.NET程序员必读的著作之一

机械工业出版社
China Machine Press



张子阳 著

.NET之美

.NET关键技术深入解析

Beautiful .NET: In-depth Analysis of Key Points of .NET

机械工业出版社
CHINA MACHINE PRESS

图书在版编目（CIP）数据

.NET之美：.NET关键技术深入解析 / 张子阳著. —北京：机械工业出版社，2014.1

ISBN 978-7-111-44532-6

I. N… II. 张… III. ①C语言－程序设计 ②计算机网络－程序设计 IV. ①TP312 ②TP393

中国版本图书馆CIP数据核字（2013）第252910号

本书是.NET程序员进阶修炼的必读之作，由拥有多年开发经验的资深.NET技术专家对C#和.NET中实用的、关键的和难以理解的知识点进行了深入解析，旨在帮助读者在尽可能短的时间内以尽可能低的学习成本去掌握那些最应该被掌握的知识。书中的每个知识点都辅之以精心设计的案例，易于理解，实践性强。

全书共17章，分为两个部分：第一部分（1~5章）主要讲解了C#语言中的一些关键知识点，如类型、泛型、委托、事件、对象、LINQ等；第二部分（6~17章）则对.NET中的关键知识点进行了深入剖析，如程序集、流和序列化、加密与解密、网络编程、.NET Remoting、在.NET中操作XML、.NET应用程序配置、基于角色的安全性、反射、多线程、对象生存期与垃圾回收等。

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：姜 影

藁城市京瑞印刷有限公司印刷

2014年1月第1版第1次印刷

186mm×240mm·27.75印张

标准书号：ISBN 978-7-111-44532-6

定 价：79.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991 88361066 投稿热线：（010）88379604

购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

前　言

为什么要写这本书

我一直认为写总结是学习的最好方式之一，因为总结的过程中也是一个不断思考的过程，可以让你的领悟更加深刻。作为一名 .NET 开发人员，有很多关键的知识点是必须掌握的，否则就会遇到很多障碍。不善于总结的人很难成为一名合格的 .NET 工程师。我就习惯将所遇到的各种知识点尽可能地研究透彻，然后总结成文章。起初，我将一些文章发表在了我的博客上，出乎意料的是大部分的文章都很受欢迎。有不少朋友留言或者发邮件给我，鼓励我出一本关于 .NET 的书。我想如果我的文章能帮助到更多的朋友，那么出一本又何妨？于是，就有了现在这本书。

写书需要很高的技巧，对于一个技术或者知识点，自己理解它或许还算容易，但把自己理解的内容写出来让别人也能理解却不是一件容易的事情。书中的每个章节，我都尽量采用循序渐进的方式进行讲解，由一个看似简单、微不足道的知识点进行切入，随后不断延伸，从而展开整个主题。这种方式对于一些朋友来说可能稍嫌累赘，但对于大多数朋友来说会更容易接受。

微软推出 .NET 已经有很多年了，现在市场已经不乏 .NET 的书籍，很多 .NET 开发人员的枕边案头已经堆放了不止一本的 .NET 书籍。因此，在本书中，我不想去重复一些简单基础的内容，因为很多书中都会涉及这方面的知识，例如类型声明、语法、循环语句等；也不想去写一些高深莫测几乎永远也用不到的特性和功能，例如代码访问安全性和一些很底层的东西。本书选择的主题，大多是关键、重要且不是很好理解的，相信认真学完本书，应该就可以帮你打通“.NET 任督二脉”了。

.NET 在过去十年当中的发展可谓是日新月异，经历了多次重要版本更新。2002 推出了 .NET 的第一个版本 1.0；2005 年推出了 .NET 2.0，2.0 的最大变化就是引入了泛型，同时新增了大量的类型；2006 年推出了 .NET 3.0，并预装在 Vista 操作系统中，3.0 主要引入了 WF、WCF、WPF 几项技术；2007 年推出了 .NET3.5，3.5 中最激动人心的变化就是引入了 LINQ，LINQ 的推出在很大程度上改变了以前程序员编程的习惯和方式；2010 年 .NET 迎来了 4.0 版本，其中的主要革新是加入了动态编程、并行计算、默认参数、协变和逆变；目前，最先的版本是 .NET 4.5，于 2012 年发

布，4.5 是对 4.0 的一个就地更新。展望未来，.NET 还会不断地变化和发展，函数式编程、并行计算、动态特性都有可能成为 .NET 继续延伸的方向。

显然，本书不可能涉及所有这些版本中引入的新特性，很多的单一主题就足够写一整本书了。但是，这本书将会帮助你奠定一个良好的 .NET 功底，有了这个功底以后，再去学习这些琳琅满目的新特性，就会变得得心应手了。

读者对象

本书的读者对象是已经入门且正在向中高级进阶的 .NET 开发人员，包括：

- .NET 工程师
- 由其他技术转向学习 .NET 的技术人员
- 学习 .NET 的高校学生
- 开设相关课程的大专院校的师生

如何阅读本书

本书分为两大部分：

第一部分为 C# 语言，C# 是为 .NET 而生的语言，该部分重点讲解了类型基础、泛型、委托、LINQ 几个部分。

第二部分为 .NET 平台，共挑选了 11 个 .NET 中的重点主题进行讲解。

本书各章的知识点是相对独立的，因此学习其中一个章并不需要你通读前面的所有章节，可以自由地选择感兴趣和薄弱的环节进行阅读。

勘误和支持

由于我的水平有限，加之编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。如果在书中发现错误或疏漏，可以发送邮件至 JimmyZhang@SoftComz.com 与我联系，我会将勘误发表在我的博客上 http://jimmyzhang.cnblogs.com。同时，如果在阅读中遇到任何问题，也可以给我发送邮件与我进行讨论。

致谢

首先要感谢那些鼓励我出一本书的朋友们，如果不是一次次看到你们的留言、电子邮件、手机

短信催促我出一本书，我是很难下定决心的。

感谢机械工业出版社华章公司的编辑杨福川老师，当我提出出版一本书的时候，你给了我最大的支持。感谢华章公司的姜影编辑，在写作的过程中不断修缮我的稿件，显著提升了我的文字水平和稿件的质量。

感谢 .NET 社区 —— 博客园 cnblogs，通过这个平台我认识了非常多的 .NET 行业的朋友，与你们交流是一件很幸福的事。

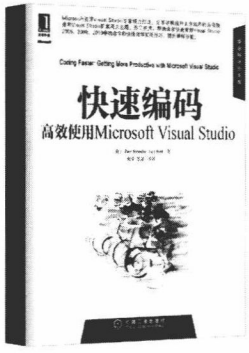
最后感谢我的家人，尤其是我的妻子，在编写本书的过程中，牺牲了很多照顾和陪伴你们的时间。

谨以此书献给众多热爱 .NET 的朋友们!

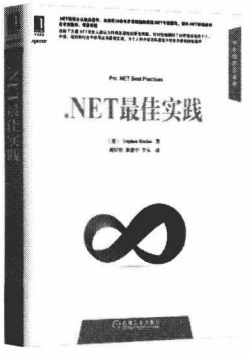
张子阳（JimmyZhang）

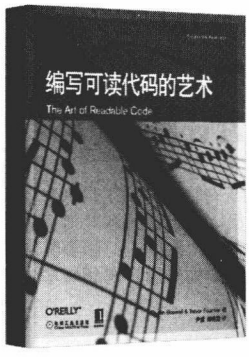
于中国深圳

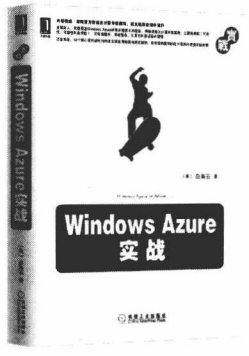
推荐阅读

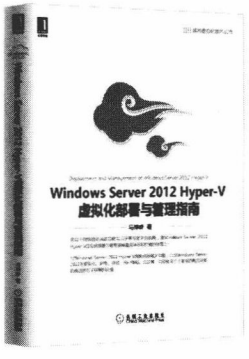

快速编码
高效使用Microsoft Visual Studio

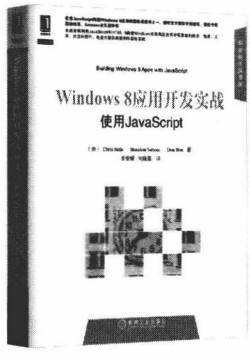

代码之殇
（原书第2版）

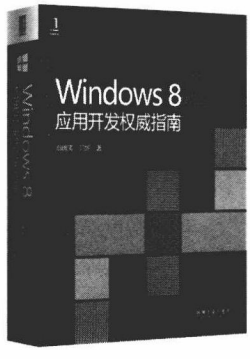

.NET最佳实践


编写可读代码的艺术
The Art of Readable Code
O'REILLY

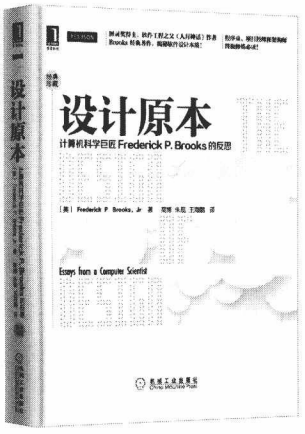

Windows Azure
Windows Azure
实战

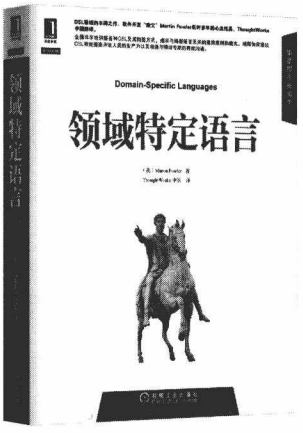

Windows Server 2012 Hyper-V
虚拟化部署与管理指南

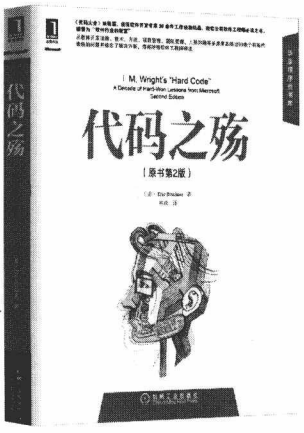

Windows 8应用开发实战
使用JavaScript


Windows 8
应用开发权威指南

推荐阅读







设计原本（精装本）

如果说《人月神话》是近40年来所有软件开发工程师和项目经理们必读的一本书，那么本书将会是未来数十年内从事软件行业的程序员、项目经理和架构师必读的一本书。它是《人月神话》作者、著名计算机科学家、软件工程教父、美国两院院士、图灵奖和IEEE计算机先驱奖得主Brooks在计算机软硬件架构与设计、建筑和组织机构的架构与设计等领域毕生经验的结晶，是计算机图书领域的又一史诗级著作。

领域特定语言

本书是DSL领域的丰碑之作，由世界级软件开发大师和软件开发“教父”Martin Fowler历时多年写作而成。全面详尽地讲解了各种DSL及其构造方式，揭示了与编程语言无关的通用原则和模式，阐释了如何通过DSL有效提高开发人员的生产力以及增进与领域专家的有效沟通，能为开发人员选择和使用DSL提供有效的决策依据和指导方法。

代码之殇

本书是《代码大全》的姊妹篇，被誉为“软件行业的财富”，资深软件开发专家30余年工作经验的结晶，是微软公司软件工程师必读之书。它从软件开发流程、技术、方法、项目管理、团队管理、人际沟通等多角度总结出90余个具有代表性的问题（大多数问题都可能会给公司或软件项目带来毁灭性灾难），并给出了问题的解决方案和最佳实践，值得所有软件工程师和项目管理者研读。

目　录

第一部分

C# 语言基础

第 1 章
C# 类型基础

学习任何一门语言，都需要搞清楚语言的类型系统，C# 也是一样，本章是全书的第一章，将详细介绍 C# 中的两种类型：值类型和引用类型，以及和它们所有关的一些其他概念。

1.1 值类型和引用类型

C# 中的类型一共分为两类，一类是值类型（Value Type），一类是引用类型（Reference Type）。值类型和引用类型是以它们在计算机内存中是如何被分配的来划分的。值类型包括了结构和枚举，引用类型则包括了类、接口、委托等。还有一种特殊的值类型，称为简单类型（Simple Type），比如 byte，int 等，这些简单类型实际上是 BCL 基类库类型的别名。比如，声明一个 int 类型，实际上是声明一个 System.Int32 结构类型。因此，在 Int32 类型中定义的方法或属性，都可以在 int 类型上调用，比如“123.Equals(2)”。

所有的值类型都隐式地继承自 System.ValueType 类型（注意 System.ValueType 本身是一个类类型）。之所以说是“隐式地”，是因为在 C# 代码中，是看不到这个继承关系的，这个关系只有通过 MSIL 代码才可以看到。System.ValueType 类型和所有的引用类型都继承自 System.Object 基类。

C# 不支持多重继承，因为结构已经隐式地继承自 ValueType，所以结构不支持继承。

说明

栈（stack）是一种后进先出的数据结构，在内存中，变量会被分配在栈上来进行操作。堆（heap）是用于为引用类型的实例（对象）分配空间的内存区域，在堆上创建一个对象，会将对象的地址传给栈上的变量（反过来叫变量指向此对象，或者变量引用此对象）。

1.1.1 值类型

现在我们更详细地看一下值类型。当声明一个值类型的变量（Variable）的时候，变量本