

唱着歌，写着代码，吃火锅。

由此，我就选择了对眼儿的.NET姑娘，在未来的日子比翼双飞，直至今日。

我喜欢托管世界的自由自在，不必为内存管理分心；

我痴迷IL和CLR底层的无限奥秘，把揭开真相作为乐趣；

我欣赏Lambda表达式和LINQ造就的优雅；

我沉醉C#行云流水般的简洁。

Broadview®
www.broadview.com.cn

INSIDE .NET

你必须知道的

.NET

(第2版)



王涛 著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

时间过得太快，
盖茨退休了，
云计算来了，
.NET都论4.0了，
Silverlight可以离线了，
WCF支持Restful了，
MVC借Razor展现，
微软以Windows Phone重整旗鼓，
Facebook连接了全世界，
Google埋头苦干Google+，
而苹果将iPhone扔向全世界，于是
《你必须知道的.NET》
第2版了。



電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书来自于微软 MVP 的最新技术心得和感悟,将技术问题以生动易懂的语言展开,层层深入,以例说理。全书主要包括了.NET 基础知识及其深度分析,以.NET Framework 和 CLR 研究为核心展开.NET 本质论述,涵盖了.NET 基本知识几乎所有的重点内容。全书分为 5 个部分,第 1 部分讲述.NET 与面向对象,从底层实现角度分析了.NET 如何实现面向对象机制,进一步分析了面向对象设计原则;第 2 部分论述了.NET 类型系统和 CLR 的内存管理机制,并对 IL 语言进行了相应介绍;第 3 部分论述.NET Framework 框架的方方面面,详细分析了.NET 框架的所有重点、难点和疑点内容,对框架类库的全貌进行了必要的专题性探讨;第 4 部分重点介绍了.NET 泛型和安全性的相关知识和本质解密;第 5 部分对.NET 3.0/3.5/4.0 新特性进行了详细的介绍和引导,对于快速入门.NET 新特性提供了方便之门。

本书适于对.NET 有一定了解的技术学习者、软件工程师和系统架构师阅读,同时也有助于.NET 初学者进行快速提高,可作为大中专院校和.NET 技术培训机构的参考教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

你必须知道的.NET / 王涛著. —2 版. —北京: 电子工业出版社, 2011.8
ISBN 978-7-121-14128-7

I. 你… II. 王… III. ①互联网络—程序设计 IV. ①TP393.4

中国版本图书馆 CIP 数据核字(2011)第 144550 号

责任编辑: 孙学瑛

文字编辑: 江 立

印 刷: 北京天宇星印刷厂

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 860×1092 1/16 印张: 34.25 字数: 700 千字

印 次: 2011 年 10 月第 2 次印刷

印 数: 4001~7000 册 定价: 79.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

○○○ 推荐序一

算起来，这是我第三次动笔为这本书写推荐。一开始以为写一个推荐非常容易，但是实际动笔才发现比我想象的要难很多。仿佛我们在准备开发一个系统的时候，实际开发的人都是准备项目可能是困难重重，而旁的人却经常一脸不屑而认为很好完成。牛和鸡的故事（注 1）一次又一次地上演，只不过这一次，我又当牛，又当了鸡。以前也以为写一本书很容易，这主要是源自我经常看到书店里琳琅满目的技术书籍，标榜以“N 天搞定×××”、“×××从入门到精通”以及“玩转×××”，但是每每翻开一看几乎都是官方教材的中文翻译版，或者是某某工作室中十几位同学不断复制粘贴的产物。所以便认为技术类的书籍基本上就是国外资料翻译加国内同行“借鉴”。而鲜有的几本精品往往也淹没在成千上万的图书海洋中，想要找到它们除了自己有孙猴子般火眼金睛的视力和如来佛祖般宽广的人脉推荐，还要有巴菲特一样足够资金支持——在国内图书市场淘到一本好的原创技术书籍，难度不比在潘家园搞到一个宣德炉低。这也是为什么很多人希望国内的技术高人能够肩负起培养下一代的重任，为像我这样的后生多多推荐好的技术书籍的原因。毕竟能够花大笔银子在潘家园买宣德炉的人并不多。

其实我无论如何也没想到王涛会邀请我为他的这部力作写序，而且还是推荐序。一来本人觉得自己能力水平差得太远，自己还需要身边牛人帮我辨识高质量的作品。二来自己在.NET 的圈子里着实算是个新人。虽然近几年也陆续认识了一些高手，但是大都属于对他们高山仰止的状态，所谓身不能至心向往之——这种水平又怎能为别人推荐呢？所以最开始接到王涛的邀请我自然表示力不能及而且层次有限。不过最终还是勉强答应了下来，一方面是整日和王涛胡聊乱侃，不能太折了兄弟的面子；二来，也是主要打动我的原因，我深知这几年他倾注在这部书上的心血。与其让这本好书淹没在一排排“赝品”之中，不如我暂且做个浮标，虽然不及灯塔那么耀眼和挺拔，但是也算增大了它的影响范围，让作为读者的我们更容易看到和知道，而不会被那些粗制滥造的东西蒙蔽了双眼走错了路。

我不清楚翻开这本书的你是否看过了《你必须知道的.NET》第 1 版。如果你认为这本书只是上一本书的添头或者修改那就大错特错了。现在音乐界流行老歌翻唱，几十年前的歌曲，随便换个编曲就可以再卖一次；电影界也是动不动就来个什么什么怀旧版，什么什么经典再映。归根到底就是再从我们这些劳苦大众兜里套点银子出来。但是这本书却不是前一本的所谓“新歌加精选”。虽然我只是看到这部书的两个样章，但是还要惊叹于这本书所涉及的内容之广、见解之深，以至于我看完了样章之后便向王涛提出了个修改意见：一定要加上两个副标题“.NET 程序员面试宝典”和“.NET 应用架构指南”。因为在这本书当中，我看到的不仅仅是和第 1 版一样对于.NET 底层深入的研究和完整的介绍，还能够看到作为一个在.NET 阵营打拼了多年的架构师对于系统架构、设计模式、面向对象等诸多方面的经验、体会与探索。关于某个具体的技术或工具的书籍在国内可能非常普遍，譬如介绍 ASP.NET 的图书可能不下几十种，但是从作者本人经验出发介绍软件设计架构的书籍便是凤毛麟角，偶有几本也是国外图书的翻译版本或者影印版本。而这本书在设计方面的部分我认为其最大的亮点，没有照本宣科的介绍，没有千篇一律的观点，所有内容都是作者本人的经验分享——有成功的经验，也有失败的经验。这其中可能不免有些内容不尽完美，有些观点尚需推敲，但这正是我们技术

人员所希望看到的：相互交流，集思广益，共同进步。而不是像国内的一些博客站点那样，一遇到观点不同就开始在评论中挖苦鄙视甚至破口大骂。虽然说我们没必要像职业书评家那样，承担着指导读者咒骂作者的使命。所以这样一部呕心沥血的作品，又怎能不让我为之吐血推荐呢？

记得有一次和王涛聊天的时候，我提到了“指月之指”的故事（注 2）。如果说像我这样水平的人写出来的书只能是传递知识的话，那么这本《你必须知道的.NET（第 2 版）》就是在传递智慧。知识只是关于知道和不知道，而智慧是无法传授的，只能自己通过实践的积累慢慢感悟。虽然说和“指月之指”的典故一样，这本书不可能就是软件设计本身，但是正如那指向明月的手指一样，能够让我们可以沿着它的方向去寻找软件设计的精髓。

写到这里，突然心中一凛，这篇推荐序写着写着更多的都是我自己的心情和感受。难道在不经意间我也成了之前所说的“书评家”对这本书开始评头论足起来。还是到此停笔吧，上面的话权当一个疯子在项目折磨之后的自言自语，书的好坏最终还是要看书的您自己去品评。至少我不想成为《伊索寓言》中所写的那个苍蝇，坐在车轴上嗡嗡大叫：“车的开动，全都是我的功劳”。

徐子岩

2011 年 6 月

推荐人简介

徐子岩，北京工业大学计算机学院毕业。现就职于宇思信德科技（北京）有限公司.NET 开发部架构师、Azure 专家、微软 Windows Azure MVP。精通.NET 平台多项技术，包括 ASP.NET MVC、WCF 等。目前专注于微软 Windows Azure 云计算平台的研究、咨询、设计和开发工作。

注 1：敏捷开发中一个著名的故事，用来说明项目会议是否需要项目组之外的人员参与发言。例如在准备牛排加煎蛋的早餐这个项目中，牛由于是贡献者（贡献自己的肉）所以它的发言是对项目有实际意义的，而鸡只作为参与者（下个蛋完事）所以会提出很多对项目进展不负责任的观点。

注 2：出自《楞伽经》卷四，“如愚见指月，观指不观月；计著名字者，不见我真实。”

○○○ 推荐序二

软件工程师，一个曾经是多么耀眼的职业，如今却沦落为 IT 民工，造成这个局面的原因是多方面的，我不想在这里深入分析。对于软件开发者来说，其心态越来越急躁，不愿意踏踏实实静下心来研究一点技术，做几年的开发工作都争着转向管理方面；而对于 IT 出版业来说，各种粗制烂造的书籍层出不穷，导致软件开发者对国人的书丧失了信心。

让人欣慰的是，国内还有一大批优秀的一线软件工程师，他们仍然保持着对技术的热情，愿意把自己在实践中的经验积累分享给广大软件开发者。本书的作者王涛（Anytao）正是其中一位，屈指算来，我们认识也有五年之久了，从最开始网上认识，到后来成为同事，再后来各自在不同的公司供职，彼此之间的联系并没有中断。我一直比较钦佩 Anytao 对于技术的热情和执着，一个人对技术保持热情不难，难的是把这种热情长期保持下去。

距离本书第 1 版的出版，已经过去三年时间了，在这几年时间里，Anytao 对于 .NET 又有了更深的理解，他把自己实践积累的经验和思考整理出书，为广大 .NET 爱好者送上了一本精品，这是 .NET 爱好者的幸事，也是 IT 出版业的幸事。虽然由于工作原因，我本人现在很少写 .NET 方面的程序，但我还是向各位 .NET 爱好者强烈推荐本书，如果你真的对软件开发有兴趣，能够在软件开发中体会到编程的使命感和荣誉感，那就静下心来认真读读本书。

最后，在本书出版之际，我和 Anytao 也将踏上人生一段新的征程。在未来的未知世界，继续走在技术的康庄大道上，编织着技术改变世界的梦想。但当我们老去的那一天，再回过头看自己走过的路，一定不会后悔自己现在的选择和对未知的执着，至少我们曾经拼搏过，对酒当歌，夫复何求！

李会军

2011 年 6 月 17 日 于北京

推荐人简介

李会军，网名 TerryLee，互联网公司架构师，编程语言爱好者，关注动态语言及函数式编程，致力于高伸缩高性能网站架构、互联网应用安全、并行计算、分布式存储、分布式计算相关技术的研究。著有《Silverlight 2 完美征程》一书，业余时间喜欢读书，尤其喜好研究历史。

前言 Expect the unexpected

时间就是一条生生不息的河，总在流淌，永不回头。时光激荡在沉淀、叛逆和重生的人生里，总在告诉你，答案就在下一站。《你必须知道的.NET（第2版）》距离第1版的问世，已经过去整整三年了，三年的时光里，技术与人生都在前行的路上，彼此影响。而留在下一站的答案，迟迟不见你来，直到三年之后。

变化每天发生，创新无所不在，.NET 正昂首阔步迈向成熟与融合，面对充满变化的技术世界，我们要做的就是：Expect the unexpected。

新版有什么

辗转三年，第2版延期而至，增加的不多，修改的不少。然而，二版的写作却是一个艰难的过程，我发现想和读者分享的还有很多很多，长长的写作 List 让人望而却步。如果照着趋势继续，可能还需要另外的700页和另外一年多的时间，选择和舍弃变得两难，痛定思痛最终呈现在你面前的就是手中的这本《你必须知道的.NET（第2版）》。

新增章节

- 第3章“OO之美”，面向对象的魅力体现在设计的多个方面，借助于.NET平台语言的特性，领略面向对象更多的奥秘。从设计的分寸体验OO，以依赖的哲学铺开讨论，认识闭包的.NET体现，然后知道什么是好代码和坏代码。
- 第14章“跟随.NET 4.0脚步”，通过.NET十年的历史演义，为.NET 4.0的新特性大幕拉开边角，从此踏上并行计算、动态编程的康庄大道，同时还有很多激动人心的4.0新体验。
- 4.4节“管窥元数据和IL”，离开了元数据支持，CLR的很多奥秘将变得苍白和空洞，而本节为很多疑惑给出了答案。
- 7.4节“认识全面的null”，null关键字是神奇而普通的一等公民，认识全面的null将能收获更多语言细节。
- 9.3节“疑而不惑：interface‘继承’争议”，通过一段争议，展开一段讨论，其中有质疑也有深入，究竟谁是谁非，且看文中乾坤。
- 9.4节“给力细节：深入类型构造器”，类型构造器的讨论，深入到了这个话题的底端，体现了细节深处的语言美丽。另一方面，细节也折射了平常编码中，需要特别关注的部分，为写出高质量代码打好基础。
- 9.8节“Name这事儿”，你认识Name吗，或许认识，或许还没有全面的认识，一个简单的话题，丰富Name这事儿。

- 11.4 节 “实践泛型”，作为最佳实践系列的代表篇章，泛型的应用有很多值得推敲和总结的建设性思想，以条款性的律条，总结出泛型应用中的最佳答案。
- 13.4 节 “LINQ 江湖”，从历史角度演义 LINQ 发展的脉络，从演义的点滴故事中，认识语言的特性，是如何而来，又因何而来。我们调侃的不光是 LINQ，还有杰出工程师们的划时代工作。

除了新增内容，还有某些章节的修订和完善。

修改章节

- 8.3 节 “历史纠葛：特性和属性”，增加了较常见的 Attribute 属性应用。
- 8.9 节 “集合通论”，为 Hashtable 实现顺序输出逻辑。
- 9.1 节 “万物归宗：System.Object”，认识好玩的 EditorBrowsableAttribute，为 Object 成员装扮隐身符。
- 9.5 节 “如此特殊：大话 String”，更多关于字符串本质和字符串驻留的讨论，认识更全面的 string 类型。

篇章的修订，为更多的内容打好补丁，还有将来更多的新内容，将不断的发布在本书支持站点。

支持

虽然作者、审稿和编辑花费大量的时间对书稿进行了反复的修改和推敲，但是限于时间和水平，仍难免失误或错误。为了使本书能更好地服务于读者，请您通过以下方式与作者或者出版社联系：

- 本书支持网站：<http://book.anytao.net/>
- 博文视点网络：<http://www.broadview.com.cn/>
- 作者个人信息：anytao@live.com（邮箱）、<http://weibo.com/anytao>（微博）
- 策划编辑：<http://weibo.com/sunnypub>（微博）

我们将竭力解决所有的问题，并向您的指正致谢。读者可以在本书的支持网站中查找相应的勘误表来避免错误。您也可以通过邮件、作者博客（<http://anytao.cnblogs.com/>）或者作者微博（<http://weibo.com/anytao>）进一步取得技术支持联系。

本书支持网站（<http://book.anytao.net/>）提供了所有代码资源、工具资源、勘误、更新内容及与作者的互动，这些资源和信息是对全书内容的有效补充与最佳辅助。

致谢

感谢子岩和会军的审稿和推荐，他们的神来之笔，为本书的品质注入坚实的保证，也让我信心满满地将自己对技术的理解再次展现在世界面前。

感谢我的父亲、母亲还有家人，他们永远是我可以归宿的港湾，是我人生的支点；感谢 Emma 的体贴入微，让我呼吸在自由的空气里，越发精神抖擞；感谢妹妹王佳，她长大了。

感谢我的朋友徐子岩、罗炳桥、徐彦华、张玉斌、高泽东、易湘、管伟、申毅、达伟、杜勇、吴宏杰、李嘉对我一如既往的关爱和支持；感谢汤文海老师、陈桦老师、Philana、Olav、方浩，他们指导我人生的历程；感谢大磊、春雨、会军、德宇、天卓、吴飞，我们即将揭开新的人生篇章，并肩向前，为技术改变世界的梦想挥洒激情。

本书的出版历经了岁月的考验和折磨，一直坚持不懈的是本书的编辑孙学瑛老师，她的专业精神和专注品质，支持我向前努力。

当然，最重要的感谢要说给一直以来关注作者和这本书的技术同行者，是你们的热情、肯定、反驳与真诚，鼓励一个以软件为人生目标的靠谱年轻人，坚持走在技术的美丽云彩下。

生活是妥协的艺术，技术又何尝不是，然而妥协不是无谓的选择，而是有选择的 *Expect the unexpected*，因此生活才有了一副美景：唱着歌，写着代码，吃火锅。

还等什么，我们出发。



2011年6月，于北京

1 版前言 Thinking More

“你站在桥上看风景，看风景的人在楼上看你”。

技术探求，正是如此的富有哲理。在.NET 世界里，每个程序设计者都是站在桥头的守望者，渴望品味所有的美景，将技术的各个方面尽收眼底。而现实往往是，你看到的并非全部真实的，技术的理解往往也需要辅助一个望远镜才能看得更加透彻。这本《你必须知道的.NET》既是一本技术的风景画卷，涵盖了.NET 基本知识的几乎所有的重点内容；又为你送上手中的望远镜，与作者一起力求对每个技术要点的探讨都更进一步。

走近这幅画卷，除了品味每一处风景，还应学会拨开表象、认识本质、探求细微，更重要的是在这个过程中，你将能收获如何为自己搭建一处技术美景。在楼上看你的人，是否会觉得风景这边独好，就看你的技艺精湛与否了。

面对技术，你别无选择，.NET 世界是如此精彩，而我们要做的就是：Thinking More。

本书是什么

对于技术，大部分著作都是从整体角度进行系统性的论述，知识体系一脉相承。拿起这样的书，我们习惯循规蹈矩地从前言看到后记，往往会陷入其系统之中，被其思想所固化，而无法找出什么是更值得关注的要点。本书显然不是一本系统性论述技术的专著，因此也无法兼顾.NET 技术的所有概念和知识，但是本书力图从重点分析与突出把握的角度来阐释技术，分析问题，将所有.NET 开发人员最关心、最困惑的技术内容形成体系进行深度遍历、挖掘和探索。

《你必须知道的.NET》正揭示了这样的一种诉求，将.NET 技术中的核心内容以一个个专题的形式来深度刻画，然后形成体系。综观全书内容：一方面，以最少的语言表达最多的技术、体察更深的本质。佛家传道，以例说理，丝丝入扣，环环揭密。本书以“你必须知道”而自诩，唯有意图达到以实例为基点，以归纳为方法的技术论述特点：对于技术的论述和分析，力求做到深入浅出、娓娓道来；对于晦涩艰深的问题以故事性的分析来引导；对于典型的问题以对比的角度来揭密；对于知识性的内容以归纳总结形成纲要。作者对每个技术要点的论述，均结合浅显易懂的实例来展开，将复杂的技术问题化解在循序渐进的思考中。让你的“悟”道，快乐而轻松。

另一方面，.NET 技术就是一座美丽的花园，里面开满了各种各样的花朵，就像类型系统、内存机制、垃圾回收、关键字、泛型、安全性、语言特性、框架格局、面向对象等，一支一朵娇艳绽放，要想品味整个花园的芬芳，你就必须了解每朵花的美丽。本书不仅告诉你如何来鉴赏这些花朵，而且告诉你如何通过施肥、除草、浇水来经营这些美丽，一步一步建立对核心技术要点的理解，从而“悟”到整个.NET 框架体系和运行机制。

.NET 技术正是一个大花园的集合，每个程序开发者也必须经历一次深入的磨练，在基本认识的水平上，

进一步，才能发现更多。就像练武之人，除了研习一招一式，了解常用的控件，了解典型的框架；还得修炼内功，认识运行机制，理解框架类库，品味设计架构。

这些正是本书呈现于读者的内容，也体现了不同于其他.NET 专著的风格。

本书有什么

对于.NET 来说，应用的范围千头万绪，但至少有一件事必须去做，那就是无限接近和触摸它的内核：CLR，这正是本书所阐述的最核心内容。下面，我们来了解一下《你必须知道的.NET》由哪些绚丽的色彩组成：

- 第一部分：**渊源**，探讨面向对象基本要素和设计原则，建立一个程序设计的基础架构思维，并结合.NET 技术来实现相关的面向对象机制，进而探求相关的面向对象原则。从底层角度认识高层本质，是深入理解的不二法门。
- 第二部分：**本质**，在梳理 IL 基本内容的基础上，了解和掌握探求.NET 本质的方法；品味类型系统，了解值类型与引用类型的底层奥秘，揭示参数传递的不惑之解；深入内存管理，认识垃圾回收，以循序渐进的分析，通晓运行时底层机制。
- 第三部分：**格局**，将.NET 关键字逐个把玩，深入浅出了解你不知道的关键字秘密；实现巅峰对决，将 const 和 readonly、class 和 struct、is 和 as、特性和属性、接口和抽象类、覆写和重载、浅拷贝和深拷贝、静态与非静态以及集合，这些技术重灾区一一澄清，走出理解误区；通过框架诠释，揭开.NET 基本技术的本质，深度诠释 Object、对象判等、String、枚举、委托和异常等.NET 核心话题；最后以命名空间为主线建立对.NET 框架的全局纵览，通过梳理命名空间和典型类型，把握.NET 框架类库的心脏和骨架。
- 第四部分：**拾遗**，通过对.NET 泛型的理解和深入，着重把握建立泛型编程的思维方式；并适度介绍.NET 安全性的主要角落，通过对代码访问安全和基于角色的安全论述，来铺陈.NET 在安全编程方面的技术体验。
- 第五部分：**未来**，以.NET 3.0/3.5 新特性为基点，全面阐述.NET 新特性的方方面面，在引导性的论述中建立对 C# 3.0、LINQ、WCF、WPF、WF 等新技术和 Visual Studio 2008 工具的基本认知和学习指导，吹响新技术的号角。

通过 5 个部分的全面讲述，将基本建立对于面向对象设计与原则，.NET 框架体系与运行时机制、.NET 框架类库格局与高级特性、.NET 安全与新特性的深入理解，对于.NET 的认识将在底层把握和设计应用上更进一步。

本书为谁而写

本书起源于作者在国内最专注的.NET 技术网站博客园 (<http://www.cnblogs.com>) 的写作经历，并在博客园的 2007 年末大盘点 Top10 的五大排行榜中位列其中 3 个榜单。作者的系列文章深受大家的关注和讨论，因此本书的内容反映了最直接的技术关注话题，适合于对.NET 技术有意进一步提高的所有学习者和开发者。

本书涵盖.NET 基本知识的几乎所有的重点内容，如果读者有以下问题、需求或者困惑，那么选择本书非

你莫属：

- 本书并不是从“什么是.NET”这一概念开始的，对于想要了解.NET 基础的读者来说，全书以一个个的专题形式来展开，可以快速建立起对.NET 基本概念的切入。
- 读完了大部头的.NET 巨著，还意犹未尽，抑或是不知所措。本书给你补充未尽的本质，解答未知的困惑，为你迅速进入.NET 底层研究，提供最好的入口。
- 你已经做得够好了，系统地学习了 C#或者 VB.NET 语言的基础，了解了基本的应用规则，但还是觉得游离于技术之外，并未接触本质。基础研究和高级教程之间往往存在着断层，想在基础之上更进一步，本书可以为你提供更多思考和研究平台，为你揭开 CLR 的神秘面纱打好基础。
- 对.NET 框架的体系架构和运行机制，有意补充认知的读者，可以通过本书建立起快速的理解。
- 本书没有 ASP.NET，没有 Web Service，也没有.NET Remoting，然而本书的内容对于深刻的理解所有.NET 应用大有裨益。只有从本质上抓住这些基础内容，才能在.NET 应用领域游刃有余，从方法学的角度来看，这才是最有效的技术学习曲线。
- 本书是一部方法论，除了探讨.NET 的基本问题，对.NET 的学习方法和学习工具均有所涉猎。了解一种科学的学习方法，有助于你以更好的质量读完本书，并取得收获。
- 本书是应对技术面试的圣经，综合了来自现实世界的问题和答案，为你快速成长提供了良好的辅助教材。
- 本书并非想创造新的技术和技巧，而是将技术以简单的方式更深一步的讲明白。如果你总是对学习的方法充满了困惑和怀疑，那么以本书作为起点会找到一个更好的方法。
- 对于每个问题的探讨，本书力求深入浅出，让人有胃口读完所关注的话题，并展开思考和讨论。对于厌倦了枯燥论述的读者而言，本书的轻松论述不会让你心感疲惫。

本书如何阅读

关于.NET，本书着眼于基础、本质和方法，对于阅读本书的读者而言，带着思考进行基础和本质的探索，同时也能体验技术学习的有效方法。作者在论述大部分的知识要点时，都会总结和归纳其重要的规律和注意事项，这些归纳为实际的编程提供了良好的遵守法则，读者应该花必要的精力熟练掌握所有的归纳内容。

技术之间是有联系的，平铺直叙的写作和由前到后的阅读都是没有意义的，本书把握从技术的联系点入手阐述基本知识，从技术的关联中形成有层次的认知角度，能够更加清晰的了解.NET 框架的全局。所以，阅读本书应该在不同的章节间切换，按照作者指引的关联进行跳跃式的阅读，能够收获更多的心得。

关于语言，本书以 C#语言实现所有的代码示例，这是因为全书虽然以.NET 为核心来论述，但也无可避免的对 C#语言的某些特性进行了分析。从广义的角度来看，C#语言本身也是.NET 体系中不可分割的一部分，对于某些语言特性的了解也能从更全面的角度来透视.NET 框架。

关于代码，读者可以通过 <http://www.broadview.com.cn> 或 <http://book.anytao.com> 来下载本书的源代码，解压缩之后按照代码使用说明，通过 Visual Studio 工具进行编译和调试。

支持

虽然作者、审稿和编辑花费了大量的时间对书稿进行了反复的修改和推敲，但是限于时间和水平，仍难免避免失误或错误。为了使本书能更好地服务于读者，请您将关于本书的任何错误信息发至以下任何链接：

- 作者个人邮箱：anytao@live.com
- 作者个人微博：http://weibo.com/anytao
- 本书支持网站：http://book.anytao.net/
- 博文视点网络：http://www.broadview.com.cn/

我们将竭力解决所有的问题，并向您的指正致谢。读者可以在本书的支持网站中查找相应的勘误表来避免错误。您也可以通过邮件或者作者博客（http://anytao.cnblogs.com/）进一步取得技术支持联系。

本书支持网站提供了所有代码资源、工具资源及其他导航信息支持，这些资源和信息是对全书内容的有效补充与最佳辅助。

致谢

首先感谢为本书审稿的蒋金楠，他的技术功底和专业素质令我钦佩，他的审阅和建议为本书增色不少，这本书有他的心血和付出。

本书的出版离不开我在博客园的成长和锻炼，感谢杜勇（dudu）站长为.NET 技术人员提供了难得的纯学术环境和氛围，感谢所有在博客园中与我笑谈技术、品论人生的朋友；感谢蒋金楠与我一起创建和支持 CLR 研究团队；感谢杜勇、李会军、程杰、刘彦博、张大磊几位朋友在百忙中对本书的审阅及点评；感谢装配脑袋、Jeffrey Zhao、Bruce Zhang 对我的指导和帮助；感谢阿不、宋国安、Volnet、Justin、EagleFish、刘荣华、Jill Zhang、随风流月、丁学、怪怪等对本书的建议和关注；还要感谢我的朋友吴宏杰、管伟、高泽东、党明、达伟对我一直以来的支持。

将最重要的感激送给养育我的父母和伴我成长的妹妹王佳，慈母严父是我人生的灯塔，激励我努力前行。感谢岳父岳母对我的关心和爱护，并将爱送给 Emma，感谢她每天在身边的鼓励与关怀，品尝她愈发炉火纯青的厨艺，让我的思绪在逻辑和理性间飞舞。

最后要感谢电子工业出版社孙学瑛编辑，正是她的不懈努力和不断支持才使我的写书过程充满了自信和快乐。还有对本书投入精力、提出建议的胡辛征编辑和其他博文视点同仁，他们的专业素质和敬业精神令我感动，才使得本书有机会服务于大众。

这本《你必须知道的.NET》送给所有技术之路上的同伴，让我们一起远航。进一步，你便是大内（dotnet）高手。

王涛

2008 年 1 月，于北京

目 录

第 1 部分 渊源——.NET 与面向对象

第 1 章 OO 大智慧	2	1.4.7 结论	33
1.1 对象的旅行	3	1.5 玩转接口	34
1.1.1 引言	3	1.5.1 引言	34
1.1.2 出生	3	1.5.2 什么是接口	34
1.1.3 旅程	3	1.5.3 .NET 中的接口	35
1.1.4 插曲	4	1.5.4 面向接口的编程	38
1.1.5 消亡	6	1.5.5 接口之规则	40
1.1.6 结论	7	1.5.6 结论	40
1.2 什么是继承	7	参考文献	40
1.2.1 引言	7	第 2 章 OO 大原则	41
1.2.2 基础为上	7	2.1 OO 原则综述	42
1.2.3 继承本质论	9	2.1.1 引言	42
1.2.4 秘境追踪	13	2.1.2 讲述之前	42
1.2.5 规则制胜	16	2.1.3 原则综述	43
1.2.6 结论	17	2.1.4 学习建议	44
1.3 封装的秘密	17	2.1.5 结论	44
1.3.1 引言	17	2.2 单一职责原则	44
1.3.2 让 ATM 告诉你, 什么是封装	17	2.2.1 引言	44
1.3.3 秘密何处: 字段、属性和方法	19	2.2.2 引经据典	45
1.3.4 封装的意义	23	2.2.3 应用反思	45
1.3.5 封装规则	23	2.2.4 规则建议	47
1.3.6 结论	24	2.2.5 结论	48
1.4 多态的艺术	24	2.3 开放封闭原则	48
1.4.1 引言	24	2.3.1 引言	48
1.4.2 问题的抛出	24	2.3.2 引经据典	48
1.4.3 最初的实现	25	2.3.3 应用反思	49
1.4.4 多态, 救命的稻草	27	2.3.4 规则建议	52
1.4.5 随需而变的业务	30	2.3.5 结论	53
1.4.6 多态的类型、本质和规则	31	2.4 依赖倒置原则	53

2.4.1 引言	53	3.2.4 解构控制反转 (IoC) 和依赖注入 (DI)	79
2.4.2 引经据典	53	3.2.5 典型的设计模式	82
2.4.3 应用反思	54	3.2.6 基于契约编程: SOA 架构下的依赖	83
2.4.4 规则建议	56	3.2.7 对象创建的依赖	84
2.4.5 结论	56	3.2.8 不规则总结	87
2.5 接口隔离原则	56	3.2.9 结论	87
2.5.1 引言	56	3.3 模式的起点	87
2.5.2 引经据典	56	3.3.1 引言	87
2.5.3 应用反思	57	3.3.2 模式的起点	88
2.5.4 规则建议	59	3.3.3 模式的建议	90
2.5.5 结论	59	3.3.4 结论	91
2.6 Liskov 替换原则	59	3.4 面向对象和基于对象	91
2.6.1 引言	59	3.4.1 引言	91
2.6.2 引经据典	59	3.4.2 基于对象	91
2.6.3 应用反思	60	3.4.3 二者的差别	91
2.6.4 规则建议	61	3.4.4 结论	92
2.6.5 结论	62	3.5 也谈.NET 闭包	92
参考文献	62	3.5.1 引言	92
第3章 OO之美	63	3.5.2 什么是闭包	92
3.1 设计的分寸	64	3.5.3 .NET 也有闭包	93
3.1.1 引言	64	3.5.4 福利与问题	95
3.1.2 设计由何而来	64	3.5.5 结论	96
3.1.3 从此重构	65	3.6 好代码和坏代码	96
3.1.4 结论	67	3.6.1 引言	96
3.2 依赖的哲学	67	3.6.2 好代码、坏代码	97
3.2.1 引言	67	3.6.3 结论	105
3.2.2 什么是依赖, 什么是抽象	68	参考文献	105
3.2.3 重新回到依赖倒置	73		

第2部分 本质——.NET 深入浅出

第4章 一切从IL开始	108	4.2.1 引言	113
4.1 从Hello, world 开始认识IL	109	4.2.2 使用工具	113
4.1.1 引言	109	4.2.3 为何而探索	115
4.1.2 从Hello, world 开始	109	4.2.4 结论	116
4.1.3 IL 体验中心	109	4.3 教你认识IL 代码——IL 语言基础	116
4.1.4 结论	113	4.3.1 引言	116
4.2 教你认识IL 代码——从基础到工具	113	4.3.2 变量的声明	116
		4.3.3 基本类型	117

4.3.4	基本运算	118	5.2.4	对症下药——应用场合与注意 事项	158
4.3.5	数据加载与保存	118	5.2.5	再论类型判等	159
4.3.6	流程控制	119	5.2.6	再论类型转换	159
4.3.7	结论	120	5.2.7	以代码剖析	160
4.4	管窥元数据和 IL	120	5.2.8	结论	167
4.4.1	引言	120	5.3	参数之惑——传递的艺术	167
4.4.2	初次接触	120	5.3.1	引言	168
4.4.3	继续深入	123	5.3.2	参数基础论	168
4.4.4	元数据是什么	125	5.3.3	传递的基础	169
4.4.5	IL 是什么	128	5.3.4	深入讨论, 传递的艺术	170
4.4.6	元数据和 IL 在 JIT 编译时	130	5.3.5	结论	174
4.4.7	结论	134	5.4	皆有可能——装箱与拆箱	175
4.5	经典指令解析之实例创建	134	5.4.1	引言	175
4.5.1	引言	134	5.4.2	品读概念	176
4.5.2	newobj 和 initobj	134	5.4.3	原理分拆	176
4.5.3	ldstr	136	5.4.4	还是性能	179
4.5.4	newarr	137	5.4.5	重在应用	180
4.5.5	结论	139	5.4.6	结论	182
4.6	经典指令解析之方法调度	140	参考文献	182	
4.6.1	引言	140	第 6 章	内存天下	184
4.6.2	方法调度简论: call、callvirt 和 calli	140	6.1	内存管理概要	185
4.6.3	直接调度	142	6.1.1	引言	185
4.6.4	间接调度	146	6.1.2	内存管理概观要论	185
4.6.5	动态调度	147	6.1.3	结论	186
4.6.6	结论	147	6.2	对象创建始末	186
参考文献	147	6.2.1	引言	187	
第 5 章	品味类型	148	6.2.2	内存分配	187
5.1	品味类型——从通用类型系统开始	149	6.2.3	结论	193
5.1.1	引言	149	6.3	垃圾回收	193
5.1.2	基本概念	149	6.3.1	引言	193
5.1.3	位置与关系	150	6.3.2	垃圾回收	193
5.1.4	通用规则	151	6.3.3	非托管资源清理	197
5.1.5	结论	152	6.3.4	结论	204
5.2	品味类型——值类型与引用类型	152	6.4	性能优化的多方探讨	204
5.2.1	引言	152	6.4.1	引言	204
5.2.2	内存有理	152	6.4.2	性能条款	204
5.2.3	通用规则与比较	156	6.4.3	结论	210
			参考文献	211	

第3部分 格局——.NET 面面俱到

第7章 深入浅出——关键字的秘密	214	7.6.7 结论	245
7.1 把 new 说透	215	7.7 非主流关键字	245
7.1.1 引言	215	7.7.1 引言	245
7.1.2 基本概念	215	7.7.2 checked/unchecked	246
7.1.3 深入浅出	217	7.7.3 yield	247
7.1.4 结论	219	7.7.4 lock	250
7.2 base 和 this	219	7.7.5 unsafe	252
7.2.1 引言	219	7.7.6 sealed	253
7.2.2 基本概念	219	7.7.7 结论	254
7.2.3 深入浅出	220	参考文献	254
7.2.4 通用规则	224	第8章 巅峰对决——走出误区	255
7.2.5 结论	224	8.1 什么才是不变: const 和 readonly	256
7.3 using 的多重身份	224	8.1.1 引言	256
7.3.1 引言	224	8.1.2 从基础到本质	257
7.3.2 引入命名空间	225	8.1.3 比较, 还是规则	259
7.3.3 创建别名	225	8.1.4 进一步的探讨	260
7.3.4 强制资源清理	227	8.1.5 结论	263
7.3.5 结论	230	8.2 后来居上: class 和 struct	263
7.4 认识全面的 null	230	8.2.1 引言	263
7.4.1 引言	230	8.2.2 基本概念	263
7.4.2 从什么是 null 开始	230	8.2.3 相同点和不同点	264
7.4.3 Nullable<T> (可空类型)	232	8.2.4 经典示例	265
7.4.4 ??运算符	234	8.2.5 结论	268
7.4.5 Null Object 模式	235	8.3 历史纠葛: 特性和属性	268
7.4.6 结论	238	8.3.1 引言	268
7.5 转换关键字	238	8.3.2 概念引入	268
7.5.1 引言	239	8.3.3 通用规则	270
7.5.2 自定义类型转换探讨	239	8.3.4 特性的应用	271
7.5.3 本质分析	240	8.3.5 示例	273
7.5.4 结论	242	8.3.6 结论	277
7.6 预处理指令关键字	242	8.4 面向抽象编程: 接口和抽象类	277
7.6.1 引言	242	8.4.1 引言	277
7.6.2 预处理指令简述	242	8.4.2 概念引入	277
7.6.3 #if、#else、#elif、#endif	243	8.4.3 相同点和不同点	279
7.6.4 #define、#undef	244	8.4.4 经典示例	281
7.6.5 #warning、#error	244	8.4.5 他山之石	283
7.6.6 #line	245	8.4.6 结论	283