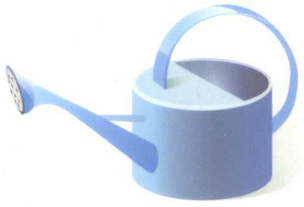
谁能够让自己像浑浊的大水一样安静下来，
慢慢得到澄清？
——《老子》

# 更锋利的C#代码

## ——编写高质量C#程序

- 一个**好的程序**，**不仅仅是**能得出正确的**运行结果**。
- 每个章节的内容**似乎**都为大家**所熟悉**，然而**视角却完全不同**。
- 通过对那些几乎被人们**忽视了的细节**的精心处理，不断地提高**每一行代码的质量**。
- 它们为什么是必须的，而**并非形式主义**。
- C#提供的每种**语言机制**的功能**背后**，体现了怎样的**逻辑含义**。
- 读完此书，你会站在**更高的角度**对C#体系拥有**更深的认识和把握**。

包善东 著

清华大学出版社

原创经典

# 更锋利的C#代码

## ——编写高质量C#程序

包善东 著

清华大学出版社
北 京

## 内 容 简 介

一个好的程序，不仅仅是能得出正确的运行结果，而且还应在其内部保持清晰的代码逻辑和语义，否则，跟随在正常结果之后的也许是艰难的代码维护工作，对程序进行一处修改往往会牵一发而动全身，一不小心就会埋下深深的陷患。从另一个角度来说，如果每一行代码的质量都很高，那么这个软件产品也一定是高质量的。这就像 ISO 9000 的质量体系认证一样，与其在产品生产完成之后再进行检验，不如控制每一步生产环节的质量。

本书由浅入深、由表及里地讲述存在于 C#编码开发中的各种质量问题，让读者清楚地了解什么是应该做的，什么是不应该做的。C#提供的每种语言机制的功能背后，体现了怎样的逻辑含义。当遇到具体的问题时，应该如何选择与取舍。阅读完此书的每一个章节，都会让读者站在更高的角度 C#体系拥有更深的认识和把握，不断向软件开发的更高层次迈进。



**图书在版编目（CIP）数据**

更锋利的 C#代码——编写高质量 C#程序/包善东著. —北京：清华大学出版社，2008.10
ISBN 978-7-302-17942-9

Ⅰ. 更…　Ⅱ. 包…　Ⅲ. C#语言－程序设计　Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2008）第 092612 号

**责任编辑**：陈　冰
**责任校对**：徐俊伟
**责任印制**：王秀菊
**出版发行**：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
**社　总　机**：010-62770175　　**邮　　购**：010-62786544
**投稿与读者服务**：010-62776969，c-service@tup.tsinghua.edu.cn
**质　量　反　馈**：010-62772015，zhiliang@tup.tsinghua.edu.cn
**印 刷 者**：北京鑫丰华彩印有限公司
**装 订 者**：三河市金元印装有限公司
**经　　销**：全国新华书店
**开　　本**：203×260　**印　张**：21.5　**字　数**：618 千字
**版　　次**：2008 年 10 月第 1 版　　**印　　次**：2008 年 10 月第1 次印刷
**印　　数**：1～5000
**定　　价**：49.00 元

# 前　言

这是一本以 C# 语言为基础，着眼于代码质量的编程指导。虽然没有砖头书的厚重，但却汇集了许许多多开发人员大量的实践经验。每个章节的内容似乎都为大家所熟悉，然而视角却完全不同，通过对那些几乎被人们忽视了的细节的精心处理，不断地提高每一行代码的质量。相信无论是 C# 初学者，还是具有 .NET 经验的开发人员，都能从本书中得到启发，写出质量更好的代码，开发出更加专业的程序。

## 为什么要写这本书？

很多人在编程时，仅仅追求其可以运行出正确的结果。这也许与学校中长期的数学式的编程教学方式有关，教学中涉及的都是一些局部问题，即使有完整的程序，规模也非常有限，通常只考虑理想状况下算法与逻辑的正确性，很少去全面地考察它的工程质量。然而，当软件开发成为一种工业时，团队成员之间的高效合作成为首要的因素。任何一个具有一定规模的软件公司都深知：软件的工程质量往往比代码的算法效率更为重要。而现代化的质量管理并非仅仅是对产品的检验，更多的来自于过程的质量控制——用最佳的方式编写每一行代码，确保每一行代码都是高质量的。

可惜的是，涉及这方面知识的著作并不多，即使提及，也只是非常肤浅的一些关于缩进之类的书写格式上的建议。深入到代码之间的语义与逻辑关系层面，专门阐述代码质量的书可谓是凤毛麟角。难得的几部专著也大多是以 C/C++ 语言为主，很多规则及其背后的理由对于 C# 语言及 .NET 类库设计并不适用，读者在使用 C# 中的许多新特性（如索引器、事件、委托、异常等）之时也无法得到针对性的指导与帮助。本书不但紧密地结合了 C# 自身的特点，给出了丰富的原则性指导并进行了详细的解释，而且更具有实用操作性。

## 什么样的人应该读这本书？

本书虽然遍及 C# 的各种语法现象，但目的并不在于语言本身的教学，而是帮助读者发现那些影响代码质量的细节，并着手进行改进。对于初学者来说，可以配合相关的 C# 教材一同阅读；对于有经验的 C# 使用者来说，则可以作为一本开发过程中的手边参考。

如果您能遵照本书建议的原则进行开发，那么当别人看到您写出的代码或者发布的类库时，会相信该程序出自一位经验丰富的专业编程人员或团队，从而进一步对其产生充分的好感和信任。同时，代码阅读起来也非常易于理解，其他开发人员也能以最快的速度学习并掌握如何使用您开发的类库。因为您在编写每一行代码、定义每一个类型、设计每一个接口的时候，都从它的使用者的角度进行了充分的考

虑——即使这个使用者是您自己。良好的开发习惯不仅仅是为了方便他人，大多数的时间内，更重要的是方便了开发人员自己。这就如同对函数、类这些机制的使用一样，就算离开了团队开发，我们仍然会使用函数来理清结构并重用代码。同样，我们自己在一处写下的函数或类，终究还是会被自己在程序的另一个地方使用。高质量的程序编码无疑会给自己带来极大的便利——这种便利并不是可以让你在写程序时省事偷懒，而是可以不必将大量的精力耗费在弄懂一个星期之前写好的代码逻辑之上。

如果您是开发团队中的负责人、项目经理或总设计师，但你的团队缺少统一的开发规范，那么有了本书的内容作基础，您将可以轻松地建立起团队开发的编码标准和开发守则。并且您可以使用本书中所描述的充分的理由告诉您的团队成员，它们为什么是必须的，而并非形式主义。

# 本书内容

本书共 16 章，涵盖从 C# 代码风格（Coding Style）到国际化开发中的种种细节问题。第 1 至 3 章深入阐述了通用的代码书写格式与命名规范，第 4 至 7 章从面向过程的角度对开发质量准则进行了讨论，第 8 至 10 章则转向关于面向对象设计的指导原则，第 11 至 15 章针对 C# 及 .NET 所特有的机制进行相关讨论，第 16 章专门介绍了国际化应用开发时应当考虑的一些问题。由于前后章节的内容并没有严格的依赖关系，因此读者完全可以根据自己的兴趣与需要来决定阅读的顺序，而不必遵照原始章节顺序。

# 术语使用说明

术语的统一向来是计算机书刊的一个重要问题。同一个概念在不同时期、不同领域都可能有不同的具体表述方式。C# 是 Microsoft 公司专有的产品，并不像 C/C++ 那样开放，术语的统一相对较为容易。然而即便如此，仍有一些“官方”术语可能令人产生混淆。

重载与重写是最容易引起误解的一对术语，它可能与其他语言中类似概念的表述存在偏差。重载（Overload）指的是多个拥有不同函数签名的同名函数，相互通过参数的个数、数据类型、顺序上的差异进行区分。而重写（Override）指的是在派生类中对基类型中的虚成员或抽象成员提供自己的实现版本。

属性是另一个容易引起误解的术语，事实上，它是两个不同概念在中文里的重名翻译。大多数情况下，“属性”指“Property”，即结构或类的一种成员，通过 get/set 访问器进行读写。而“属性”有时所指的是“Attribute”，表示所有从 System.Attribute 派生的类型，用于对编程元素进行描述性声明。在下面的代码中：

```
[Serialzable]
public class Employee
{
    private string name = null;
```

```
        public string Name
        {
            get { return name; }
            set { name = value; }
        }
    }
```

Name 是 Employee 类的一个 Property，而 Serialzable 则是用于对 Employee 类进行补充声明的一个 Attribute。遗憾的是，官方并没有提供区分这两个概念的中文解决方案，好在 Property 使用的频率远远高于 Attribute。如果没有特殊的说明或者上下文的提示，本书中所有的属性均指 Property。

为了准确地表达术语，在本书中，所有术语在正文中被首次提及时，或者其他必要的位置，都会附带相应的英文表述，供读者参考。

# 排版约定

本书中包含大量的 C# 代码示例，为了便于阅读，在排版时使用了类似代码加亮的方式：关键字以**粗体**方式显示，而伪代码则用*斜体*显示。注意：为了加以区别，非 C# 语言的代码示例不使用代码加亮格式。

在整体上，本书还采用了带底纹与无底纹两种格式。带底纹的代码示例表示正确或推荐的做法，并附有示例编号与小标题；而无底纹的代码则表示错误或不恰当的做法，以及一般的补充说明之用。请看下面的示例：

```
Console.WriteLine("这是不带底纹的代码示例。");
```

以及，

代码示例 x-x：带底纹的代码示例

```
Console.WriteLine("这是带底纹的代码示例。");
```

# 关于截图的使用

书中所有 Microsoft Visual Studio 软件的屏幕截图的使用符合微软公司的相关许可规定。

# 致谢

这本书的创作过程真可谓是一场战斗，回想我写过的大大小小的程序、网站和文章，这几乎是时间

跨度最大的一桩。从 2006 年底开始构思、进行内容和框架整理，到 2007 年夏天开始正式动工，再到现在整本书的完稿、出版，自己都觉得不可思议。在我看来，时间如此之长主要有两个原因。一是这本书的内容主要来源于经验的总结、归纳与提炼，并不像一般的技术教程写起来那么顺畅。很多时候，内容要点本身的整理就占据了很多时间，更不要说如何将它们表述清楚了。另一个原因是客观的，这本书开工后不久，工作就越来越忙，每天能够用于专心写书的时间越来越少。以至于到 2007 年底时，实际进度已经比计划慢了整整一个月。回想这半年多的时间里，多少次觉得精疲力尽难以继续，而又是在亲人和朋友们的多少次鼓励中坚持了下来。毫不夸张地说，他们对这本书的完稿起到了决定性的作用。

在此，我要感谢我的父亲包宗顺、母亲瞿忆玲、以及我的女友应筠之。他们与我朝夕相处，对我的写书工作给予了全力的支持。

我想对清华大学出版社的编辑陈冰，以及参与本书编辑出版的所有工作人员表示致意。如果没有他们对我的信任与支持、以及认真负责的工作，我就无法通过这本书来与大家分享我的经验。

感谢胡妍妍同学，她帮我指出并改正了许多语言表述上的错误，为这本书的“中文编码质量”作出了极大的贡献。我还要感谢杜文远、郑明鉴、以及从未谋面的何晓杰，他们提出了许多宝贵的意见和建议，并在本书创作最困难的阶段给予我关心与鼓励。

包善东

2008 年 6 月 30 日

# 目　录

# 第 9 章 封装

# 第 14 章 LINQ 查询



# 第 15 章 异常



# 第 16 章 全球化与本地化