<<更锋利的C#代码>>

图书基本信息

书名：<<更锋利的C#代码>>

13位ISBN编号：9787302179429

10位ISBN编号：7302179425

出版时间：2008-10

出版时间：清华大学出版社

作者：包善东

页数：326

<<更锋利的C#代码>>

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。

<<更锋利的C#代码>>

内容概要

《更锋利的C#代码：编写高质量C#程序》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。
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ISO9000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。

作者简介

包善东（网名Richard Bao），群硕软件开发有限公司的一名交互设计师和软件工程师。
9岁时萌生了对编程的浓厚兴趣，从此走上了软件开发的道路，至今已积累了十多年的编程经验。
作者还曾是其学校交响乐团的大提琴兼钢琴演奏员，在英、法、德、港、台及内地多次进行演出。
也许是音乐与艺术思想对编程的渗透，使其在编程中往往善于寻找和谐之美，避免一切生搬硬套。
这也许才是《更锋利的C#代码》思想的根源吧。

<<更锋利的C#代码>>

书籍目录

章节摘录

第1章基本的代码风格　　假设我们写的是文章而不是程序，那么你一定觉得诸如文章应该分为若干个自然段、每段开头空两格之类的规则是理所当然的。
如果段落的开头不空两格，或者干脆把整个文章写成单独的一段，仔细想来似乎也不会影响文章实质内容的表达。
既然如此，我们为什么还要在形式上下功夫呢？
设想一下，如果你手中的这本书既无章节也无目录，正文中的不同内容都使用同样的字体字号印刷，几百页纸从头至尾洋洋洒洒如念经般地“一气呵成”，你还有耐心看下去吗？
　　这是一个人人都能理解的道理，可是当文章变成程序的时候，就不是每个人都能想得通的了。
不仅仅是初学者，甚至一些熟练的开发人员，也会写出凌乱不堪的代码。
许多人一定有过这样的经历：一年半载之后，自己原来写的程序就完全看不懂了。
如果这段程序只是为了交作业，或者临时一用，那还可以不去追究，但如果这是一个商业软件，现在需要根据客户的要求进行修改的话，工作量可就大了——你不得不先花时间把你原来的思路看懂。
　　肯定会有人反驳：代码是给机器运行的，又不是给人看的，写那么好看有什么用？
　　他的话只对了前半句：代码确实是给机器运行的，可是机器总共才需要看它几分钟？
你花一个月编写的程序，机器顶多两三分钟就编译好了——在这两三分钟之前，这代码不都是你在看吗？
开发软件编写代码不是一朝一夕的事情，更多的情况下，一个软件的开发要经历很长的时间，并且常常由多人合作完成。
一个庞大的软件项目，可能会动用上千名程序员工作数年！
如果把代码写得连自己都看不明白，怎么与别人交流？
同一个开发团队内，一定要保持良好且一致的代码风格，才能最大化地提高开发效率。
　　有的初学者会问：我现在只是一个人写程序，并不需要和其他人合作，这些条条框框还有什么必要吗？
　　要知道，团队协作只是一个方面。
我经常遇到这类情况，一些初学者拿着他的程序来说：“这个怎么不能编译？”我帮他把代码整理了半天，发现有一个地方丢了半个大括号。
如果他写程序的时候能够稍加注意一些的话，相信此类错误完全可以避免。
保持良好的编程习惯，能够避免的错误还远不止这些。

<<更锋利的C#代码>>

编辑推荐

《更锋利的C#代码：编写高质量C#程序》由清华大学出版社出版。

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。

它们为什么必须是，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与C＃体系拥有更深的认识和把握。

<<更锋利的C#代码>>

版权说明