

Broadview®
www.broadview.com.cn

构建高质量的 C#代码

曹化宇◎编著

代码不是私人财产,而是艺术品

创建高质量的代码是每一个真正的程序员
坚持不懈的追求



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

构建高质量的 C#代码

曹化宇◎编著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

作为软件的基石，代码的质量决定了最终产品的质量，本书从这一点出发，介绍了高质量 C#代码的成就过程，即从基础代码到软件结构，以及不断优化和重构的过程。本书主要内容包括 C#的基本语法、结构与应用特点，常用开发资源与技术要点，设计模式在 C#中的应用等，以及特别重要的一点，即我们应该养成良好的开发习惯，不仅要注重技术细节，还要从更广阔的视野角度来重新审视代码的构建工作。

无论你是 C#初学者，还是正在应用 C#的开发者，如果你关注代码的质量，愿意进一步提高软件项目的质量，本书能为你提供帮助。

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

构建高质量的 C#代码 / 曹化字编著. —北京：电子工业出版社，2013.4
ISBN 978-7-121-19713-0

I. ①构… II. ①曹… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2013) 第 039936 号

策划编辑：孙学瑛

责任编辑：贾 莉

印 刷：北京中新伟业印刷有限公司

装 订：北京中新伟业印刷有限公司

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：787×980 1/16

印张：28.75

字数：644 千字

印 次：2013 年 4 月第 1 次印刷

印 数：4000 册 定价：69.00 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 zltts@phei.com.cn，盗版侵权举报请发邮件到 dbqq@phei.com.cn。

服务热线：(010) 88258888。

本书关注的主题是如何使用 C# 写出高质量的代码。

我们知道，高质量的代码是优秀软件的基础，创建高质量的代码是每一个真正的程序员坚持不懈的追求，同时，也是一项艰巨而又充满乐趣的工作。我们认为，在编写高质量代码的过程中，需要考虑以下三个方面：

- **编写高质量的基础代码。**我们会使用某种编程语言写出语句或基础结构，比如我们使用 C# 语言定义变量，或者是创建选择语句、循环语句结构，又或者定义方法、结构、类、接口，等等。这些内容作为软件代码中的基本元素，与所使用的编程语言的特点是息息相关的。只有当我们熟练掌握并合理使用了编程语言的特点，才能写出高质量的基础代码。
- **创建高质量的代码结构。**在这里所说的代码结构，是指在软件开发中，用于实现业务模型的软件架构。高质量的软件架构可以帮助我们有效完成业务模型所要达到的设计目标，同时，也可以应对不断变化的软件需求。在软件开发和经验积累的过程中，产生了众多的软件开发技术和方法，它们可能与编程语言无关，但对应于每一种编程语言，实现起来又有一定的独特性。设计模式就是在不断的开发和积累过程中，发现并总结出的一系列用于优化代码结构的方法。在软件开发中合理地使用成熟的设计模式，可以帮助我们高效地创建灵活、易扩展、富有弹性的软件架构。这也是我们在应对“需求不断变化”这一软件开发中唯一不变真理时的常用且行之有效的开发技术。
- **逐步改进。**无论是基础代码还是代码结构，要想在初次开发时就能达到很高质量的可能性并不是很大。因此，这就需要我们对代码进行不断的改进，而改进的步骤应该是循序渐进的，而不是暴风骤雨似的。在软件开发中，代码和架构的改进过程，也就是重构（Refactoring）的过程，需要我们对每一条语句或语句结构、算法、架构进行逐步的改进和优化。其中，对于软件架构的重构过程，往往就是应用或去除设计模式的过程。而我们应该注意的是，重构是在不改变或少量修改代码外在行为的基础上，对其内部结构的优化，以便达到代码和架构设计更合理、更简洁、更灵活的目的。重要的是，只有在不断的尝试和挫折中才能让我们积累到更多的、有价值的软件开发经验。此外，我们应该注意代码逐步改进的过程，在这个不断尝试的过程中，会产生多个代码和架构的版本，只有通过对比，我们才能找出最合适的解决方案，这也是经验积累的重要过程。

本书将结合 C#编程语言、重构与设计模式，在掌握 C#语言基础知识的同时，进一步学习如何使用 C#语言实现一些常用的设计模式，而重构则是不断地进行基础代码与软件架构逐步优化的过程，在这一过程中，我们将需要更多地理解高质量代码的进化过程，从而可以帮助我们在使用 C#语言开发软件时，能够更合理、更高效地创建出高质量的代码，并使软件架构变得更易维护、更灵活、更富有弹性。

本书特点

- **享受充满乐趣的学习与开发过程。**软件开发应该是一个充满乐趣的过程，而软件开发的学习过程也应该是这样。本书使用了通俗易懂的语言，结合比较有趣的示例代码，比如大量使用了模拟创建游戏的代码示例，让大家在一个比较轻松的氛围中学习和应用知识点。
- **丰富而实用的代码示例。**本书包含了大量的示例代码，并在其中穿插了很多实用的内容，比如随机排序算法、中国农历信息的获取、树状结构，以及大量的设计模式模型，等等。大家可以在自己的软件中直接使用这些代码，也可以根据需求对其进行修改、扩展或者简化，从而可以更有效、更合理地运用这些算法和代码结构。
- **大量实用的开发技巧和方法。**在讨论知识点的时候，我们列举了大量的实用开发技巧和方法，并通过实际测试来解答一些技术上的疑问，真正做到能见到树木，亦能见到森林。

本书内容

本书主要包括以下几个部分的内容：

C#编程语言，涵盖了使用 C#开发应用软件过程中常用的内容，如数据类型、结构化语句、数组、结构类型、枚举类型、类、委托、泛型、接口、多线程与资源同步等。并对一些不太常用的内容做了简单的介绍，如预处理、特性、不安全代码、分部类型和可空类型等。

设计模式，介绍了在 C#和.NET Framework 中应用的四种设计模式，包括访问者、迭代器、适配器和模板方法模式。然后，我们详细介绍了 11 种设计模式，它们是策略、工厂方法、抽象工厂、生成器、单件、观察者、享元、组合、装饰者、状态和桥接模式，并创建了这些设计模式的基本模型，这些都是在 C#开发应用软件过程中可能会经常使用到的模式。读者应该在学习和使用设计模式的过程中领悟到软件开发的精髓和本质所在，而不只是照葫芦画瓢。读者应该明白，应用设计模式并不是我们的目标，我们的目标是创建高质量的软件，任何技术的取舍都应该基于这一目标。

在贯穿全书的内容中，我们穿插介绍了各种编写高质量代码、架构，以及逐步改进的方法、技巧，并给出了一些代码演化的过程。

最后，我们讨论了 C#、设计模式、重构，以及软件构建中涉及的一些主题，感兴趣的读者可以选择一些主题进行深入的学习和研究，以扩展自己在软件开发方面的知识面。

读者类型

对于本书的读者，你可以是：

- 软件开发的初学者，可以从基础的 C#编程语言开始学习，然后，逐步学习设计模式的应用过程，以提高自己对代码结构整体的把握能力。
- 从其他语言转到 C#语言的读者，无论你是不是已经会使用 VB、C++，或者 Java，都可以很快地适应 C#开发的便利性，并且可以通过设计模式在 C#中的应用，更深入地理解 C#语言的特点和软件构建方法。
- C#使用者，如果你正在使用 C#语言进行软件的开发工作，可以从本书学习到如何在 C#中应用一些常见的设计模式，以便提高我们对软件代码的重构和优化能力。同时，从 C#编程语言的角度，本书也可以作为参考手册，供你随时查询相关的知识点。

本书的使用

如果你对 C#语言和 .NET Framework 不太熟悉，可以从第 1 章开始逐章阅读，一步步深入学习相关知识。

如果你已经对 C#比较熟悉，可以直接阅读在 C#中使用设计模式的相关内容。

本书也可以作为 C#语言以及设计模式的参考手册，读者在开发的过程中，可以随时在本书中查询 C#语言与应用设计模式，以及了解如何改进和优化软件代码和架构。

此外，代码与架构重构的过程介绍，穿插于不同的主题之中，方便读者直观地了解应用相关技术需要注意的问题，以及代码和架构优化，即逐步改进的过程。

学习方法

你可以先通读一遍。如果你的时间比较紧，至少应该先看一下目录，了解一下本书有哪

些内容，这样，可以做到对全书的结构和内容有一个初步的了解。

然后，你可以从第 1 章开始，或者选择自己感兴趣的内容阅读，但你应该充分理解每一个知识点和示例。学习软件开发最好的方法就是实践，读者应该多读代码、多写代码。建议有条件的读者可以自己输入本书中的所有示例，并尽可能地去尝试修改代码，测试不同的方式带来的运行结果上的差异。相信这么多的代码敲下来，读者对代码的理解会更深入。另外，这样做还会有一个额外的收获，那就是你的打字速度可能会有所提高☺。在阅读过程中，对于不太明白的地方，不用着急，当你系统地学习后，一些难题就会迎刃而解。

请注意，如果你对哪个知识点暂时不能理解，可以先放松一下，如玩玩休闲小游戏、睡一觉，或者出去散散步，再静下心来时，答案就很可能自然地出现在脑海里了。

最后一点，在学习的过程，手边一定要有笔和纸，随便写写、画画对我们的学习都会有所帮助，有时还会有意外的收获。

交流与反馈

就像开发软件一样，撰写一本书也不太可能在一开始就达到完美的境地。如果读者发现问题，或者有什么建议，又或者只是交流一下软件开发的心得，都可以给我发来 E-mail，我的邮箱地址是 chydev@vip.163.com。

祝大家在软件开发的过程中玩得开心！

作者

2013 年 1 月

九载耕耘奠定专业地位

博文视点诚邀精锐作者加盟

以书为证彰显卓越品质

《代码大全》、《Windows内核情景分析》、《加密与解密》、《编程之美》、《VC++深入详解》、《SEO实战密码》、《PPT演义》……

“圣经”级图书光耀夺目，被无数读者朋友奉为案头手册传世经典。

潘爱民、毛德操、张亚勤、张宏江、管辉Zac、李刚、曹江华……

“明星”级作者济济一堂，他们的名字熠熠生辉，与IT业的蓬勃发展紧密相连。

九年的开拓、探索和励精图治，成就博古通今、文圆质方、视角独特、点石成金之计算机图书的风向标杆：博文视点。

“凤翱翔于千仞兮，非梧不栖”，博文视点欢迎更多才华横溢、锐意创新的作者朋友加盟，与大师并列于IT专业出版之巅。

英雄帖

江湖风云起，代有才人出。

IT界群雄并起，逐鹿中原。

博文视点诚邀天下技术英豪加入，

指点江山，激扬文字

传播信息技术，分享IT心得

● 专业的作者服务 ●

博文视点自成立以来一直专注于IT专业技术图书的出版，拥有丰富的与技术图书作者合作的经验，并参照IT技术图书的特点，打造了一支高效运转、富有服务意识的编辑出版团队。我们始终坚持：

善待作者——我们会把出版流程整理得清晰简明，为作者提供优厚的稿酬服务，解除作者的顾虑，安心写作，展现出最好的作品。

尊重作者——我们尊重每一位作者的技术实力和生活习惯，并会参照作者实际的工作、生活节奏，量身定制写作计划，确保合作顺利进行。

提升作者——我们打造精品图书，更要打造知名作者。博文视点致力于通过图书提升作者的个人品牌和技术影响力，为作者的事业开拓带来更多的机会。



联系我们

博文视点官网: <http://www.broadview.com.cn>

新浪官方微博: <http://weibo.com/broadviewbj>

投稿电话: 010-51260888 88254368

CSDN官方博客: <http://blog.csdn.net/broadview2006/>

腾讯官方微博: <http://t.qq.com/bowenshidian>

投稿邮箱: jsj@phei.com.cn

关于本书用纸的温馨提示

亲爱的读者朋友：您所拿到的这本书使用的是 **环保轻型纸**！

环保轻型纸在制造过程中添加化学漂白剂较少，颜色更接近于自然状态，具有纸质轻柔、光反射率低、保护读者视力等优点。其成本略高于普通纸。为给您带来更好的阅读体验并与读者共同支持环保，我们在没有提高图书定价的前提下，享纸质图书的阅读乐趣！

第 1 章 开启 C#之旅	1
1.1 软件开发概述	1
1.1.1 开发应用软件	1
1.1.2 软件开发的一般步骤	2
1.2 代码构建的重要性	3
1.3 认识 C#编程语言	4
1.3.1 C#简史	5
1.3.2 C#关键字及相关标识符	6
1.3.3 C#中的语句和表达式	7
1.4 构建 C#开发环境	8
1.4.1 安装开发环境	8
1.4.2 C#程序的基本结构	10
1.4.3 如何创建窗体程序	12
1.5 高质量代码准则：可阅读	15
1.5.1 养成好的代码编写习惯	15
1.5.2 代码不是私人财产，而是艺术品	16
1.5.3 代码应该具有可维护性	16
1.5.4 注释你的代码	16
1.6 高质量代码准则：用实践证明一切	17
1.7 高质量代码准则：好心情带来高质量	18
1.8 准备你的开发工具箱	19
第 2 章 处理数据	21
2.1 C#数据处理基础	21
2.1.1 C#数据类型	21
2.1.2 变量与常量	23
2.1.3 高质量代码准则：给变量起个好名字	24
2.1.4 值类型与引用类型	25

2.1.5	字面值	26
2.2	整数.....	26
2.2.1	整数的一般运算.....	26
2.2.2	溢出检查	28
2.2.3	++与--运算	31
2.2.4	移位运算	32
2.2.5	高质量代码准则：准确使用整数.....	33
2.3	浮点数与 decimal 类型.....	34
2.3.1	浮点数与 decimal 数据的运算	34
2.3.2	高质量代码准则：合理使用浮点数和 decimal 类型.....	35
2.4	布尔类型.....	40
2.4.1	布尔运算	40
2.4.2	布尔类型数据的用途.....	41
2.5	字符.....	41
2.5.1	char 类型	41
2.5.2	ASCII 码表（0~127）	42
2.6	字符串.....	44
2.6.1	转义字符	44
2.6.2	逐字字符串（@）	45
2.6.3	字符串的常用操作.....	46
2.6.4	单个字符时使用 char 类型.....	51
2.6.5	使用 StringBuilder 类型	52
2.7	日期与时间的处理	53
2.7.1	使用 DateTime 结构类型.....	53
2.7.2	获取中国农历信息.....	55
2.8	数据类型的转换.....	58
2.8.1	显式转换	58
2.8.2	隐式转换	60
2.8.3	TryParse()方法转换.....	60
2.8.4	使用 Convert 类.....	61
2.8.5	转换是有代价的.....	63

2.9	object 类型	63
2.9.1	一切皆对象	63
2.9.2	装箱和拆箱	64
2.9.3	高质量代码准则: object 不应作为万能类型	65
2.10	C#运算符	67
第 3 章	控制程序流程	69
3.1	if 语句	69
3.1.1	使用 if 语句	69
3.1.2	注意事项	76
3.2	switch 语句	77
3.2.1	使用 switch 语句	77
3.2.2	注意事项	81
3.3	for 语句	81
3.3.1	使用 for 语句	81
3.3.2	注意事项	84
3.4	while 语句	85
3.4.1	使用 while 语句	85
3.4.2	注意事项	86
3.5	do 语句	86
3.5.1	使用 do 语句	86
3.5.2	注意事项	87
3.6	foreach 语句	88
3.6.1	使用 foreach 语句	88
3.6.2	注意事项	89
3.7	break 语句	89
3.8	continue 语句	90
3.9	goto 语句与标签	91
第 4 章	处理异常	93
4.1	C#中的异常处理	93
4.1.1	Exception 类型	93
4.1.2	try-catch 语句结构	93
4.1.3	处理具体的异常类	95

4.1.4	使用 finally 语句块	96
4.1.5	throw 语句	98
4.2	高质量代码准则：处理异常智能化、模式化	100
4.3	在 Visual C#中调试代码	101
第 5 章	数组	106
5.1	C#中的数组	106
5.1.1	简单数组	106
5.1.2	洗牌——随机排序算法	109
5.1.3	多维数组	110
5.2	使用 Array 类	113
5.2.1	排序	113
5.2.2	反序排列	114
5.2.3	修改成员数量	115
5.3	使用 ArrayList 类	116
5.4	使用 Hashtable 类	118
5.5	表驱动法——完善中国农历信息的处理	120
第 6 章	结构	124
6.1	创建结构类型	124
6.2	结构成员	125
6.2.1	结构中的方法	125
6.2.2	结构中的属性	126
6.2.3	其他成员	128
6.3	结构的应用	128
第 7 章	枚举	130
7.1	C#中的枚举	130
7.2	枚举的应用	133
第 8 章	类	135
8.1	创建类的作用与目的	135
8.2	C#中的类	135
8.2.1	创建类	135
8.2.2	创建类的实例	136
8.2.3	构造函数与析构函数	137

8.2.4	构造函数的重载.....	139
8.2.5	构造函数链.....	140
8.3	类成员的可访问性.....	141
8.4	变量.....	142
8.5	属性.....	142
8.6	方法.....	145
8.6.1	创建方法.....	145
8.6.2	方法的参数.....	147
8.6.3	方法的返回值.....	152
8.6.4	方法的重载.....	152
8.6.5	高质量代码准则：创建高质量的方法.....	154
8.7	索引器.....	154
8.8	静态类与静态成员.....	157
8.8.1	静态成员.....	157
8.8.2	静态类.....	159
8.9	关于类的更多内容.....	160
8.10	高质量代码准则：创建高质量的类.....	160
8.11	封装中国农历信息类.....	161
8.11.1	第一个版本.....	161
8.11.2	重构版本.....	165
第9章	类的继承.....	170
9.1	继承概述.....	170
9.1.1	现实世界中的继承.....	170
9.1.2	C#中的继承.....	170
9.1.3	能否继承.....	175
9.2	子类.....	176
9.2.1	扩展功能.....	177
9.2.2	隐藏基类成员.....	178
9.2.3	区分子类与基类成员.....	179
9.2.4	重写虚成员.....	181
9.3	抽象类与抽象成员.....	182
9.4	继承的局限性.....	187

第 10 章 接口	188
10.1 接口的特点	188
10.2 创建接口	189
10.3 实现接口	190
10.4 接口的继承	191
10.5 实现多个接口	193
第 11 章 委托与事件	197
11.1 访问者模式	197
11.2 委托	199
11.3 事件	201
11.4 多路广播委托	205
第 12 章 命名空间	207
12.1 命名空间的成员	207
12.2 命名空间的组织	208
12.3 引用与别名	209
12.3.1 引用命名空间	209
12.3.2 命名空间别名	210
12.4 使用 Microsoft.VisualBasic 命名空间	212
12.4.1 获取操作系统与内存信息	213
12.4.2 播放波形文件	215
12.4.3 网络基本应用	216
第 13 章 泛型	217
13.1 泛型方法	217
13.2 泛型类	218
13.3 约束	220
第 14 章 运算符重载	221
14.1 在结构中重载运算符	221
14.2 在类中重载运算符	223
第 15 章 资源同步与自动清理	226
15.1 多线程	226
15.2 易失域 (volatile field)	229
15.3 lock 语句	231

15.4	using 语句.....	232
第 16 章	关于 C# 的其他主题	235
16.1	预处理.....	235
16.1.1	根据条件编译代码	235
16.1.2	发布警告或错误	237
16.1.3	定义代码区域.....	238
16.2	特性.....	238
16.2.1	使用 DllImportAttribute 特性	239
16.2.2	自定义特性.....	241
16.3	不安全代码	243
16.3.1	指针与寻址运算	244
16.3.2	sizeof 运算	246
16.4	分部类型 (partial type)	247
16.5	可空类型 (nullable type)	249
第 17 章	代码的进化.....	251
17.1	从代码到架构.....	251
17.1.1	基础代码	251
17.1.2	代码集成	252
17.2	重构.....	252
17.3	设计模式	253
17.4	关注代码改进的过程	254
17.5	找寻 .NET Framework 中的设计模式.....	255
17.5.1	迭代器模式.....	255
17.5.2	适配器模式.....	258
17.5.3	模板方法模式	264
第 18 章	策略模式	266
18.1	虚拟战争游戏示例：使用策略模式组合作战单位	266
18.1.1	第一设计方案	266
18.1.2	陆、海、空.....	269
18.1.3	真正的设计方案	271
18.1.4	组合第一个作战单位.....	275
18.1.5	这一切是怎样发生的.....	277

18.2	应用分析	278
18.2.1	问题	279
18.2.2	解决方案	280
18.2.3	应用特点	284
第 19 章	工厂方法模式	285
19.1	虚拟战争游戏示例：控制作战单位的创建	285
19.1.1	收起自由创建单位的权力	285
19.1.2	控制作战单位类型	288
19.1.3	统一作战单位的创建方法	290
19.2	应用分析	294
19.2.1	问题	294
19.2.2	解决方案	295
19.2.3	使用空对象	299
19.2.4	应用特点	300
19.3	工厂方法的应用	301
19.3.1	代替构造函数	301
19.3.2	隐藏特殊对象的创建	303
第 20 章	抽象工厂模式	304
20.1	示例：模拟组装电脑	304
20.2	应用分析	310
20.2.1	问题	310
20.2.2	解决方案	311
20.2.3	应用特点	316
20.2.4	与工厂方法模式的对比	317
第 21 章	生成器模式	319
21.1	示例：创建汽车配置清单	319
21.2	应用分析	324
21.2.1	问题	324
21.2.2	解决方案	325
21.2.3	应用特点	331
第 22 章	单件模式	332
22.1	虚拟战争游戏示例：唯一的重要人物	332

22.1.1	重要人物登场	332
22.1.2	应用单件模式	334
22.2	应用分析	337
22.2.1	问题	337
22.2.2	解决方案	337
22.2.3	应用特点	338
22.2.4	多线程中的单件	338
第 23 章	观察者模式	341
23.1	订阅邮件——现实版观察者模式	341
23.2	虚拟战争游戏示例：空降作战	342
23.2.1	准备登机	342
23.2.2	诺曼底大空降	346
23.3	.NET Framework 4.0 中的观察者	347
23.4	应用分析	351
23.4.1	问题	351
23.4.2	解决方案	351
23.4.3	应用特点	358
第 24 章	享元模式	359
24.1	虚拟战争游戏示例：点缀作战地图	359
24.1.1	第一个方案	359
24.1.2	使用享元模式	361
24.2	应用分析	364
24.2.1	问题	364
24.2.2	解决方案	364
24.2.3	应用特点	368
第 25 章	组合模式	369
25.1	虚拟战争游戏示例：多武器平台	369
25.1.1	一个平台，多种武器	369
25.1.2	使用组合模式	369
25.2	应用分析	372
25.2.1	问题	372