Broadview
www.broadview.com.cn

高质量程序设计指南

C++/C语言

第3版 修订版

林锐 韩永泉 编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高质量程序设计指南 C++/C语言

第3版 修订版

林锐 韩永泉 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

高质量程序设计是软件行业的薄弱环节，大部分企业为此付出了高昂的代价，只能通过大量的测试和改错来提高软件产品的质量。因此，如何让程序员熟练地掌握编程技术和编程规范，在开发过程中内建高质量代码，是IT企业面临的主要挑战之一。

本书以轻松幽默的笔调向读者论述了高质量软件开发方法与C++/C编程规范，而这也是作者多年从事软件开发工作的经验总结。全书共17章，第1章到第4章重点介绍软件质量和基本的程序设计方法；第5章到第16章重点阐述C++/C编程风格、面向对象程序设计方法和一些技术专题；第17章阐述STL的原理和使用方法。

本书第1版和第2版部分章节曾经在网上广泛流传，被国内IT企业的不少软件开发人员采用。本书的附录C《大学十年》是作者在网上发表的一个短篇传记，文中所描述的充满激情的学习和生活态度，感染了大批莘莘学子。

本书的主要读者对象是IT企业的程序员和项目经理，以及大专院校的本科生和研究生。

图书在版编目（CIP）数据

高质量程序设计指南：C++/C语言 / 林锐，韩永泉编著. — 3版（修订本）. — 北京：电子工业出版社，2012.10

ISBN 978-7-121-18617-2

Ⅰ. ①高… Ⅱ. ①林… ②韩… Ⅲ. ①C程序—程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2012）第226923号

策划编辑：张春雨
责任编辑：贾　莉
印　　刷：中国电影出版社印刷厂
装　　订：三河市鹏成印业有限公司
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：787×1092　1/16　　印张：25.75　　字数：576千字
印　　次：2012年10月第1次印刷
定　　价：65.00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。

前　言

本书出版后一直受到不少软件公司和 C++程序员的关注，但不知不觉间也绝迹很久。不断有读者问我从何处可以买到本书、什么时候再版。现在这一版本的推出，也可视作对这一询问的一种回答。

说来惭愧，我从 2002 年写完本书第 1 版后，再也没有接触过 C++编程，现在对 C++已经很陌生了。2004 年 1 月我离开上海贝尔，创办了上海漫索计算机科技有限公司，专注于 IT 企业的研发管理整体解决方案（包括软件产品和咨询服务）。我自己已经从技术专家转型为企业管理者，关注商务多于软件技术。对于出版本书第 3 版，我的确心有余而力不足。幸好第 2 版的作者韩永泉仍然从事应用软件开发，宝刀未老，让我对第 3 版的质量充满信心。

在撰写这一版的时候，为了更进一步突出本书一贯强调的“高质量程序设计”理念，对原书前版的内容做了一些调整：

首先是进行了全面的修订，改正了所有已经发现的错误，并对原有部分章节的内容进行了补充；

其次，删除了第 2 版的第 2 章和第 17 章（名字空间和模板）。根据我们的观察，除非是开发类库等通用程序，第 17 章的内容在现阶段对应用软件开发人员一般不具有实际指导价值；

最后，增加了大约 10 个小节的内容，分散在各章中。这些增加的内容是实际应用软件开发过程中经常会用到的技术，可以显著地提高编程效率，增强软件的健壮性和可移植性。

不论本书第 1 版和第 2 版是好是差，它都被过度地使用了，产生了令作者始料不及的影响。本书的试题被国内软件公司大面积地用于 C++程序员招聘考试，结果事先看过答案的应试者考了高分而被录取，还真有人向我致谢；也有不少人未看过答案而考了低分未被录取，在网上把作者骂一通。本书的试题和答案早在 2002 年就公开了，不知有多少人看过，我很奇怪怎么到现在还被煞有介事地用于考试。

希望读者正确地使用本书：请您学习和应用您（或公司）认为好的东西，不要把本书当作标准来看待，不要全部照搬，也不必花费很多时间去争议本书是好还是坏。如果你发现书中的错误或不妥之处，请及时告知作者韩永泉，或发邮件至 northwest_wolf@sina.com，或直接上他的 Blog 与他交流：http://blog.csdn.net/northwest_wolf/。

林锐

上海漫索计算机科技有限公司 总经理

睿泰科技集团董事、首席研发管理专家

http://www.mansuo.com

linrui@mansuo.com

目　录

第 1 章　高质量软件开发之道

本章讲述高质量软件开发的道理。

为了深入理解软件质量的概念，本章阐述了 10 个重要的软件质量因素，即正确性、健壮性、可靠性、性能、易用性、清晰性、安全性、可扩展性、兼容性和可移植性，并介绍了消除软件缺陷的基本方法。

人们开发软件产品的目的是赚钱。为了获得更多的利润，人们希望软件开发工作“做得好、做得快，并且少花钱”，所以软件质量并不是人们唯一关心的东西。本章论述了“质量、生产率和成本”之间的关系，并给出了能够“提高质量、提高生产率，并且降低成本”的软件开发方法。

1.1　软件质量基本概念

1.1.1　如何理解软件的质量

什么是质量？

词典的定义是：① 典型的或本质的特征；② 事物固有的或区别于其他事物的特征或本质；③ 优良或出色的程度。

CMM 对质量的定义是：① 一个系统、组件或过程符合特定需求的程度；② 一个系统、组件或过程符合客户或用户的要求或期望的程度。

上述定义很抽象，软件开发人员看了准会一脸迷惘。软件的质量不容易说清楚，但我们今天非得把它搞个水落石出不可。

就以健康做类比吧。早先人们以为长得结实、饭量大就是健康，这显然是不科学的。现代人总是通过考察多方面的生理因素来判断是否健康，如测量身高、体重、心跳、血压、血液、体温等。如果上述因素都合格，那么表明这人是健康的。如果某个因素不合格，则表明此人在某个方面不健康，医生会对症下药。同理，我们也可以通过考察软件的质量属性来评价软件的质量，并给出提高软件质量的方法。

一提起软件的质量属性，人们首先想到的是“正确性”。“正确性”的确很重要，但运行正确的软件就是高质量的软件吗？不见得，因为这个软件也许运行速度很低，并且浪费内存，甚至代码写得一塌糊涂，除了开发者本人谁也看不懂，也不会使用。可见正确性只是反映软件质量的一个因素而已。

软件的质量属性很多，如正确性、精确性，健壮性、可靠性、容错性、性能、易用性、安全性、可扩展性、可复用性、兼容性、可移植性、可测试性、可维护性、灵活性等。除此之外还可以列出十几个，新词可谓层出不穷。

上述这些质量属性“你中有我，我中有他”。如果开发人员每天都要面对那么多

的质量属性咬文嚼字，不久就会迂腐得像孔乙己，因此我们有必要对质量属性做些分类和整合。质量属性可分为两大类：“功能性”与“非功能性”，后者有时也称为“能力”（Capability）。

从实用角度出发，本章将重点论述“10 大”质量属性，如表 1-1 所示。

表 1-1 “10 大”软件质量属性

功 能 性	正确性（Correctness）
	健壮性（Robustness）
	可靠性（Reliability）
非 功 能 性	性能（Performance）
	易用性（Usability）
	清晰性（Clarity）
	安全性（Security）
	可扩展性（Extendibility）
	兼容性（Compatibility）
	可移植性（Portability）

其中，功能性质量属性有 3 个：正确性、健壮性和可靠性；非功能性质量属性有 7 个：性能、易用性、清晰性、安全性、可扩展性、兼容性和可移植性。

为什么碰巧是“10 大”呢？

不为什么，只是方便记忆而已（如同国际、国内经常评“10 大”那样）。

为什么“10 大”里面不包括可测试性、可维护性、灵活性呢？它们不也是很重要的吗？

它们是很重要，但不是软件产品的卖点，所以挤不进“10 大”行列。我认为如果做好了上述“10 大”质量属性，软件将会自然而然地具备良好的可测试性、可维护性。人们很少纯粹地去提高可测试性和可维护性，勿要颠倒因果。至于灵活性，它有益处也有坏处，该灵活的地方已经被其他属性覆盖，而不该灵活的地方就不要刻意去追求。

根据经验，如果你想一股脑儿地把任何事情都做好，结果通常是什么都做不好，做事总是要分主次的。什么是重要的质量属性，应当视具体产品的特征和应用环境而定，请读者不要受本书观点的限制。最简单的判别方式就是考察该质量属性是否被用户关注（即卖点）。

1.1.2 提高软件质量的基本方法

质量的死对头是缺陷，缺陷是混在产品中的人们不喜欢、不想要的东西。缺陷越多质量越低，缺陷越少质量越高。

Bug 是缺陷的形象比喻，人们喜欢说 Bug 是因为可以把 Bug 当作“替罪羊”。软件的缺陷明明是人造成的，有了 Bug 这个词后就可以把责任推给 Bug——“都是 Bug 惹的祸”。唉，当一只 Bug 真是太冤枉了！

软件存在缺陷吗？是的，有以下典故为证。

编程大师说："任何一个程序，无论它多么小，总存在着错误。"

初学者不相信大师的话，他问："如果有个程序小得只执行一个简单的功能，那会怎么样？"

"这样的程序没有意义，"大师说，"但如果这样的程序存在的话，操作系统最后将失效并产生错误。"

但初学者不满足，他问："如果操作系统不失效，那会怎么样？"

"没有不失效的操作系统，"大师说，"但如果这样的操作系统存在的话，硬件最后将失效并产生错误。"

初学者仍不满足，继续问："如果硬件也不失效，那会怎么样？"

大师长叹一声道："没有不失效的硬件。但如果这样的硬件存在的话，用户就会想让那个程序做一件不同的事，这件事也是个错误。"

没有错误的程序世间难求。（摘自《编程之道》）

错误是严重的缺陷。医生犯的错误最终会被埋葬在地下，从此一了百了。但软件的错误不会自动消失，它会一直骚扰用户。据统计，对于大多数的软件产品而言，用于测试与改错的工作量和成本将占整个软件开发周期的 30%，这是巨大的浪费。如果不懂得如何有效地提高软件质量，项目会付出很高的代价，你（开发人员）不仅没有功劳，也没人欣赏你的苦劳，你拥有最多的将只是疲劳。

怎样才能提高软件的质量呢？

还是先来听一个中国郎中治病的故事吧！

在中国古代，有一家三兄弟全是郎中。其中有一人是名医，人们问他："你们兄弟三人谁的医术最高？"

他回答说："我常用猛药给病危者医治，偶尔有些病危者被我救活，于是我的医术远近闻名并成了名医。我二哥通常在人们刚刚生病的时候马上就治愈他们，临近村庄的人都知道他的医术。我大哥深知人们生病的原因，所以能够预防家里人生病，他的医术只有我们家里才知道。"

提高软件质量的基本手段是消除软件缺陷。与上述三个郎中治病很相似，消除软件缺陷也有三种基本方式：

（1）在开发过程中有效地防止工作成果产生缺陷，将高质量内建于开发过程之中。这就是"预防胜于治疗"的道理，无疑是最佳方式，但是要求开发人员必须懂得正确地做事（门槛比较高）。**我们学习"高质量编程"的目的就是要在干活的时候一次性编写出高质量的程序，而不是在程序出错后才去修补。**

（2）当工作成果刚刚产生时马上进行质量检查，及时找出并消除工作成果中的缺陷——这种方式效果比较好，人们一般都能学会。最常用的方法是技术评审、测试和质量保证等（详见本章 1.4 节），已经被企业广泛采用并取得了成效。

（3）当软件交付给用户后，用着用着就出错了，赶紧请开发者来补救，这种方

式的代价最高。可笑的是，当软件系统在用户那里出故障了，那些现场补救成功的人倒成了英雄，好心用户甚至还寄来感谢信。☺

1.1.3 “零缺陷”理念

质量的最高境界是什么？是尽善尽美，即“零缺陷”。

“零缺陷”理念来源于国际上一些著名的硬件厂商。尽管软件的开发与硬件生产有很大的区别，但我们仍可以借鉴，从中得到启迪。

人在做一件事情时，由于存在很多不确定的因素，一般不可能 100%地达到目标。假设平常人做事能完成目标的 80%。如果某个人的目标是 100 分，那么他最终成绩可达 80 分；如果某个人的目标只是 60 分，那么他最终成绩只有 48 分。我们在考场上身经百战，很清楚那些只想混及格的学生通常都不会及格。即使学习好的学生也常有失误，因而捶胸顿足。

做一个项目通常需要多人协作。假设某系统的总质量是 10 个开发人员的工作质量之积，即最高值为 1.0，最低值为 0。如果每个人的质量目标是 0.95，那么 10 个人的累积质量不会超过 0.598。如果每个人的质量目标是 0.9，那么 10 个人的累积质量不会超过 0.35。只有每个人都做到 1.0，系统总质量才会是 1.0。只要其中一人的工作质量是 0，那么系统总质量也就成了 0。因系统之中的一个缺陷而导致机毁人亡的事件已不罕见。

上述比喻虽然严厉了一些，但从严要求只有好处没有坏处。如果不严以律己，人的堕落就会很快。如果没有“零缺陷”的质量理念，也许缺陷就会成堆。

从理念到行动还是有一定距离的，企业在开发产品时应当根据自身实力和用户的期望值来设定可以实现的质量目标。过低的质量目标会毁坏企业的声誉，而过高的质量目标则有可能导致成本过高而拖累企业（请参见本章 1.3 节）。

1.2 细说软件质量属性

1.2.1 正确性

正确性是指软件按照需求正确执行任务的能力。这里“正确性”的语义涵盖了“精确性”。正确性无疑是第一重要的软件质量属性。如果软件运行不正确，将会给用户造成不便甚至损失。技术评审和测试的第一关都是检查工作成果的正确性。

正确性说起来容易做起来难。因为从“需求开发”到“系统设计”再到“实现”，任何一个环节出现差错都会降低正确性。机器不会主动欺骗人，软件运行出错通常都是人造成的，所以不要找借口埋怨机器有毛病。开发任何软件，开发者都要为“正确”两字竭尽全力。

1.2.2 健壮性

健壮性是指在异常情况下，软件能够正常运行的能力。正确性与健壮性的区别