高质量程序设计指南

——C++/C语言

林　锐　　编著
顾晓刚　谢义军
飞思科技产品研发中心　　审校

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

内 容 简 介

高质量程序设计是软件行业的薄弱环节，这使得大部分企业只能依靠大量地测试和改错来提高软件产品的质量，并为此付出了高昂的代价。因此，如何熟练地掌握编程技术，有效地提高软件产品的质量是IT企业面临的主要挑战之一。

本书作者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。它是作者多年从事软件开发工作的经验总结，也是当前在C++/C语言编程中经常会遇到的难题。本书共15章。第1章到第4章，重点介绍软件质量和面向对象程序设计方法；第5章到第15章，重点阐述C++/C编程风格和一些技术专题。

本书前期版本部分章节，曾经在Internet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D《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，顾晓刚，谢义军编著．—北京：电子工业出版社，2002.6
ISBN 7-5053-6218-6

Ⅰ.高… Ⅱ.①林…②顾…③谢… Ⅲ.C语言－程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字（2002）第036906号

责任编辑：王树伟 何郑燕
印　　刷：北京市增富印刷有限责任公司
出版发行：电子工业出版社　http://www.phei.com.cn
　　北京海淀区万寿路173信箱　邮编：100036
经　　销：各地新华书店
开　　本：787×980　1/16　印张：17.25　字数：386.4千字
版　　次：2002年6月第1版　2002年6月第1次印刷
定　　价：28.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077

序1

北京航空航天大学软件工程研究所
名誉所长、教授、博士生导师

我国计算机教育对高质量软件开发技术尚未给予足够的重视，很多程序员虽然能熟练地掌握编程技术，但不懂得应该在开发过程中内建软件的质量。《高质量程序设计指南——C++/C 语言》一书论述了高质量软件开发方法与 C++/C 编程规范。可以认为，本书在某种程度上填补了这一领域的空白，值得 IT 企业的程序员和项目经理以及计算机软件和计算机应用专业的本科生和研究生认真阅读和参考。

全书共 15 章，前 4 章重点介绍了软件的质量属性和面向对象的程序设计方法，后 11 章重点阐述了 C++/C 编程风格和一些技术专题。

第 1 章在阐述了 10 个重要的软件质量因素的基础上，介绍了消除软件缺陷的基本方法，论述了质量、生产率和成本之间的关系，从软件过程规范、复用、分而治之、优化与折衷、技术评审、测试、质量保证和改错等 8 个方面介绍了提高质量、提高生产率和降低成本的软件开发方法。在这一章中，作者还介绍了软件过程改进的概念。应该指出，五十多年来计算机事业的发展已经使人们认识到，要高效率、高质量和低成本地开发和生产软件，必须依靠高素质的人才、先进的技术和优化的过程。这三者是软件开发和生产的不可分割的三要素。其中，过程定义了为达到某个确定的目标、需要什么人在什么时间以何种方式进行的一系列活动。每一个活动则可用输入及其输入准则、输出及其输出准则、控制（或称约束）、资源（或称支持）、活动内容以及对活动性能的度量等 8 个元素来描述。关于过程的历史遗产方面，作者指出了瀑布模型的历史作用和现实意义，并立志在吸收能力成熟度模型（CMM）的经验和教训的基础上，创建一个基于能力成熟度模型集成（CMMI）的称之为“精简并行过程”的软件过程模型。CMMI 是 CMM 的逻辑后继，是过程改善模型的新方向，不仅值得我国软件界、而且值得我国计算机应用界的密切关注。我热切地期望作者能够按

照理论与实践相结合、继承与创新相结合的原则，在具体项目的实施过程中创建并不断进化与完善这个软件过程模型。

第 2 章从“做人”和“做事”两个方面论述了程序员应该具有的素质，提倡程序员的职业道德，上班时专心工作，不损害集体利益，不干危害社会的事情，工作认真负责，努力为客户服务，并自觉地学习新技术以提高综合才能。这些论述可以纳入软件工程的基本内容。事实上，越来越多的事实已经证明，软件工程是技术科学、人文科学与实验科学的交叉，“做人”和“做事”两者必须同时强调，才能真正达到预期的效果。

第 3 章扼要介绍了编程语言的发展简史，指出每一种语言都有其优点和缺点，能够很好地解决应用问题的编程语言就是好的编程语言。在选择编程语言时，开发人员首先应该根据实际情况，选择业界推荐的并且是自己擅长的编程语言来开发软件，才能保证有较好的质量与生产效率。

第 4 章描述了面向对象程序设计方法的重要概念，指出 C++是一种应用广泛的面向对象语言，学好了 C++/C，再学习其他编程语言就比较容易。诚然，正如作者所指出的，在开发软件时，系统分析和系统设计是“战略和战术”问题，编程是实现“战略和战术”的具体技术。学习与掌握面向对象程序设计的基本方法，是学习与掌握系统分析和系统设计技术的必要前提。然而，开发软件系统的最终目标，是要用语言实现能在计算机上运行的程序。我殷切地期望，每一个软件工作者、特别是青年软件工作者，既要学习与掌握系统分析和系统设计技术，又要学习与掌握高质量编程技术，做一个具有深厚“功力”的软件开发人员。

第 5~10 章讨论了 C++/C 程序的编程风格和基本编程技术。其中第 5 章讨论了 C++/C 程序的文件结构，第 6 章详细介绍了 C++/C 程序的版式，对空行、代码行、代码行内的空格、对齐、长行拆分、修饰行的位置、注释、类的版式等提了建议。第 7 章讨论了命名规则，作者建议规定一种令大多数项目成员满意的命名规则，并在项目中贯彻实施。第 8 章归纳了正确使用表达式和语句的一些规则与建议。第 9 章介绍了常量的定义规则。第 10 章重点论述了函数的接口设计和内部实现的一些规则。特别详细地介绍了初学者非常容易出错的引用与指针的区别。众所周知，程序版式应该清晰、美观。如果版式不好，虽然不会影响程序的功能，但会影响程序的可读性，从而会影响程序的使用价值。可以认为，这几章中的论述是作者多年来宝贵经验的总结，值得倡导与推广。我也希望在软件开发第一线的技术人员，能够根据这些原则进行认真的实践、验证、补充和完善，并归纳出适合组织特征与自身特点的原则，以使这些编程风格能

够锦上添花。

第 11~15 章讨论了高质量 C++/C 程序设计的技术专题。其中第 11 章介绍了内存分配技术，内存分配错误大多没有明显的症状，时隐时现，编译器不能自动发现，本章很好地总结了防止内存分配错误的编程规范。第 12 章探讨了重载和内联的含义、优点以及使用局限性，指出两个函数同名并不一定构成重载，而且要注意隐式类型转换导致重载函数的歧义性等。第 13 章介绍了类的构造、析构和赋值函数等基本概念和实现技术。第 14 章总结了使用 C++的标准模版库进行编程的经验，介绍了容器类、迭代器、算法、函数对象、适配器和内存分配器等基本概念和用法，以帮助程序员不仅提高开发效率，而且有利于开发出结构良好的程序。第 15 章则讨论了如何使用 const 来修饰函数的参数、返回值以及函数的定义体，从而提高函数的健壮性。这些都是高质量 C++/C 程序设计的重要指南，很值得读者借鉴。

在本书正文之后还有几个附录，其中附录 A 中给出了 C++/C 试题，内容限于 C++/C 的常用语法，以求反映出读者对编程质量和 C++/C 的理解程度。附录 B 给出了这些试题的评分标准及其相应的答案。附录 C 给出了 C++/C 代码审查表，分别对文件结构、程序版式、命名规则、表达式与基本语句、常量、函数设计、内存管理、C++函数的高级特性、类的函数和高级特性以及其他常见问题（如数据类型、变量值、逻辑判断、循环和出错处理等问题）等方面给出了八十多条审查规则，并指出了其中的重点。可以认为，这是 C++/C 程序的编程风格和基本编程技术的检测单的经验总结，读者可以从中裁剪出适合组织特征与自身特点的清单，以提高代码审查的效率。

最后作者在附录 D“大学十年”中，以生动、诙谐的笔调，描写了作者在大学十年勤奋、求实但又艳丽、浪漫的生涯，并以无限的激情对年轻的朋友们说了两句肺腑之言：**主动去创造环境，否则你无法设计人生；生活和工作要充满激情，否则你无法体会到淋漓尽致的欢乐与痛苦。**

让我们与作者一起为创造美好的环境，在发展我国民族软件产业的奋斗中体会欢乐与痛苦。

序2

上海贝尔阿尔卡特
董事、副总裁、CTO
徐智铎

企业开发产品的目的是提高市场竞争力，获取利润。为了使利润最大化，人们总是希望开发工作“做得好、做得快并且少花钱”。用软件工程的术语来讲，即“提高质量、提高生产率并且降低成本”。

然而国内 IT 企业长期面临“产品质量低下”、“进度延误”、“费用超支”等问题，解决这些问题没有灵丹妙药，唯有踏踏实实地走“规范化”的研发与管理道路。“规范化”不会抑制人们的创造力，相反地，它使得团队可以大规模地复用前人积累的智慧和财富。

企业建立规范和推广规范的工作通称为过程改进。国内大中型 IT 企业只有持续地改进过程，不断地提高规范化水平，才能谋求长远的发展。

近几年来，上海贝尔平均每年有近百个研发项目，研发经费达数亿元。公司有千余名研发人员，半数以上是软件开发人员。可以说，上海贝尔面临的软件工程和项目管理问题在很大程度上覆盖了国内 IT 业界面临的共性问题。

两年前林锐博士加入本公司之际，我考虑他在软件工程方面的专长和公司的实际需求，授权林锐博士组建 SEPG（软件工程过程组），专门从事 CMM/CMMI 的研究与推广工作。为此，公司为 SEPG 提供了一流的研究与实践环境。

半年前，我在了解他的课题进展情况时，他曾喜悦地说自己的“内功”有了长足的提升。我给他立了更大的目标：不仅要提高自己的知识水平，成为业内顶尖的专家，还要不断地创作出具有很高实用价值的成果，让企业真正获益。我认为任何一位优秀的专家都应当具备这样的实力和境界。

《高质量程序设计指南——C++/C 语言》即为作者的工作成果之一。

程序设计是软件开发过程中的一个重要环节。一年前，为了提高网络应用事业部软件产品的质量，作者编写了这本书，并亲自给程序员们培训。本书的突出理念是“内建高质量，而不是修补质量”。作者总结了大量的编程规范，并

以风趣的笔调阐述其中的道理。本书曾在网上广泛流传，国内很多知名 IT 企业的程序员以及高校师生给予了很好的评价。

尽管程序设计是成熟的领域，相关书籍不计其数，但这并不妨碍精品佳作脱颖而出。我欣然为本书作序。

前言

软件质量是被许多程序员挂在嘴上而不是放在心上的东西！

除了完全外行和真正的编程高手外，初读本书，你最先的感受将是惊慌："哇！我以前捏造的 C++/C 程序怎么会有那么多的毛病？"

别难过，作者只不过比你早几年、多几次惊慌而已。

请花一两个小时认真阅读这本百页经书，你将会获益匪浅，这是前面 N-1 个读者的建议。

一、编程老手与高手的误区

自从计算机问世以来，程序设计就成了令人羡慕的职业，程序员在受人宠爱之后容易发展成为毛病特多却常能自我臭美的群体。

如今在 Internet 上流传的"真正"的程序员据说是这样的：

（1）真正的程序员没有进度表，只有讨好领导的马屁精才有进度表，真正的程序员会让领导提心吊胆。

（2）真正的程序员不写使用说明书，用户应当自己去猜想程序的功能。

（3）真正的程序员几乎不写代码的注释，如果注释很难写，它理所当然也很难读。

（4）真正的程序员不画流程图，原始人和文盲才会干这事。

（5）真正的程序员不看参考手册，新手和胆小鬼才会看。

（6）真正的程序员不写文档也不需要文档，只有看不懂程序的笨蛋才用文档。

（7）真正的程序员认为自己比用户更明白用户需要什么。

（8）真正的程序员不接受团队开发的理念，除非他自己是头头。

（9）真正的程序员的程序不会在第一次就正确运行，但是他们愿意守着机器进行若干个小时的调试改错。

（10）真正的程序员不会在上午 9:00 到下午 5:00 之间工作，如果你看到他在上午 9:00 工作，这表明他从昨晚一直干到现在。

……

具备上述特征越多，越显得水平高，资格老。所以别奇怪，程序员的很多缺点竟然可以被当做优点来欣赏。就像在武侠小说中，那些独来独往、不受约束且带点邪气的高手最令人崇拜。我曾经也这样信奉，并且希望自己成为那样的“真正”的程序员，结果没有得到好下场。

我从读大学到博士毕业十年来一直勤奋好学，累计编写了数十万行 C++/C 代码。有这样的苦劳和疲劳，我应该称得上是编程老手了吧？

我开发的软件都与科研相关（集成电路 CAD 和 3D 图形学领域），动辄数万行程序，技术复杂，难度颇高。这些软件频频获奖，有一个软件获得首届中国大学生电脑大赛软件展示一等奖。在 1995 年开发的一套图形软件库到 2000 年还有人买。罗列出这些“业绩”，可以说明我算得上是编程高手了吧？

可惜这种个人感觉不等于事实。

读博期间我曾用一年时间开发了一个近 10 万行 C++代码的 3D 图形软件产品，我内心得意表面谦虚地向一位真正的软件高手请教。他虽然从未涉足过 3D 图形领域，却在几十分钟内指出该软件多处重大设计错误。让人感觉那套软件是用纸糊的华丽衣服，扯一下掉一块，戳一下破个洞。我目瞪口呆地意识到这套软件毫无实用价值，一年的心血白花了，并且害死了自己的软件公司。

人的顿悟通常发生在最心痛的时刻，在沮丧和心痛之后，我作了深刻反省，“面壁”半年，重新温习软件设计的基础知识。补修“内功”之后，又觉得腰板硬了起来。博士毕业前半年，我曾到微软中国研究院找工作，接受微软公司一位资深软件工程师的面试。他让我写函数 strcpy 的代码。

太容易了吧？

错！

这么一个小不点的函数，他从三个方面考查：

（1）编程风格；

（2）出错处理；

（3）算法复杂度分析（用于提高性能）。

在大学里从来没有人如此严格地考查过我的程序。我花了半个小时，修改了数次，他还不尽满意，让我回家好好琢磨。我精神抖擞地进“考场”，大汗淋漓地出“考场”。这“高手”当得也太窝囊了。我又好好地反省了一次。

我把反省后的心得体会写成文章放在网上，引起了不少软件开发人员的共鸣。我因此有幸和国内大型 IT 企业如华为、上海贝尔、中兴等公司的同志们广泛交流。大家认为提高质量与生产率是软件工程要解决的核心问题。高质量程

序设计是非常重要的环节，毕竟软件是靠编程来实现的。

我们心目中的老手们和高手们能否编写出高质量的程序来？

不见得都能！

就我的经历与阅历来看。勤奋好学的程序员长期在低质量的程序堆中滚爬，吃尽苦头之后才有一些心得体会，长进极慢，我就是一例。

现在国内 IT 企业拥有学士、硕士、博士文凭的软件开发人员比比皆是，但他们在接受大学教育时就“先天不足”，岂能一到企业就突然实现质的飞跃。试问有多少软件开发人员对正确性、健壮性、可靠性、性能、易用性、可读性、可扩展性、安全性、兼容性、可移植性等质量属性了如指掌？并且能在实践中运用自如？“高质量”可不是干活小心点就能实现的！

我们有充分的理由疑虑：

（1）编程老手可能会长期用隐含错误的方式编程，习惯成自然后，被人指出发现毛病时都不愿相信那是真的！

（2）编程高手可以在某一领域写出极有水平的代码，但未必能从全局把握软件质量的方方面面。

事实证明如此。我到上海贝尔工作后，陆续面试或测试过近百名“新”、“老”程序员的编程技能，合格率低于 50%。很少有人能够写出完全符合质量要求的 if 语句，很多程序员对指针、内存管理一知半解，……

领导们不敢相信这是真的。我做过现场试验：有一次部门新进 14 名硕士生，在开欢迎会之前对他们进行“C++/C 编程技能”摸底考试。我问大家试题难不难？所有的人都回答不难。结果没有一个人及格，有半数人得零分。

竞争对手如华为、中兴、大唐等公司的朋友们也做过试验，也是类似结果。真的不是我“心狠手辣”或者要求过高，而是很多软件开发人员对自己的要求不够高。要知道这些大公司的员工素质在国内 IT 企业中是比较前列的，倘若他们的编程质量都如此差的话，我们怎么敢期望中小公司拿出高质量的软件呢？连程序都编不好，还谈什么振兴民族软件产业，岂不胡扯。

我打算定义编程老手和编程高手，请您别见笑。

定义 1：能长期稳定地编写出高质量程序的程序员称为编程老手。

定义 2：能长期稳定地编写出高难度、高质量程序的程序员称为编程高手。

根据上述定义，马上得到第一推论：我既不是高手也算不上是老手。

在写此书前，我阅读了不少程序设计方面的英文著作，越看越羞惭。因为发现自己在编程基本技能方面都未能全面掌握，顶多算是二流水平，还好意思谈什么老手和高手。希望和我一样在国内土生土长的程序员朋友们能够做到：

（1）知错就改；

（2）经常温故而知新；

（3）坚持学习，天天向上。

二、本书导读

首先请做附录 A 的 C++/C 试题（答题前不要看答案），考查自己的编程质量究竟如何。然后参照附录 B 的答案严格打分。

（1）如果你只得了几十分，请不要声张免得让人知道，也不要太难过。编程质量差往往是由于不良习惯造成的，与人的智力、能力没有多大关系，还是有药可救的。成绩越差，进步的空间就越大，中国不就是在落后中赶超发达资本主义国家吗？只要你能下决心改掉不良的编程习惯，第二次考试就能及格了。

（2）如果你考及格了，表明你的技术基础不错，希望你能虚心学习、不断进步。

（3）如果你考出 85 分以上的好成绩，你有义务和资格为你所在的团队作“C++/C 编程”培训。如果你还没有找到合适的工作单位，不妨到上海贝尔试一试。一年前我偶然地在某个事业部发现一颗无用武之地的好苗子，就把他挖过来了。希望你能和作者多多交流，大家相互促进。

（4）如果你在没有任何提示的情况下考了满分，希望你能收我做你的徒弟。

编程考试结束后，请阅读本书的正文。本书挂着编程的名义，却讲了不少做人做事的道理，请原谅我这些职业病吧。

第 1 章论述“软件质量”。编程只是软件开发中的一个环节而已，如果程序编不好，产生的软件估计也不是个好东西。但是即使编程非常棒，却未必就能产生高质量的软件。本章在漫谈软件质量之际，阐述了“提高质量、提高生产率并且降低成本”的软件开发方法。

第 2 章论述“做好程序员”。我绝对没有过分推崇程序员这个职业，也不规劝你一辈子只当一名好程序员。如果你想在软件行业干下去，我建议你首先成为一名好程序员，然后成为技术专家、管理者或者老板等，这样的道路比较踏实。我自己这么说也是这么做的。

第 3 章介绍“编程语言发展简史”。近半个世纪来，一些了不起的企业及其英雄好汉们创造了不少优秀的编程语言，使得大批程序员拥有了好饭碗和好梦想。回忆历史是为了现在，趁着计算机工业还没有被人类淘汰，我们得赶紧做点有益于社会的事情。

第 4 章介绍“C++面向对象程序设计方法”。如果把程序员比做武林人士，那么“面向对象程序设计”是上乘的刀法，而 C++则是宝刀。先讲刀法后耍刀，这样学习起来比较有效。鉴于江湖上讲刀法的和耍刀的人非常多，不少已经出版的秘籍声称能让人在几周内速成高手，作者没有秘籍，只好写个概述，免得献丑。

第 5 章至第 10 章主要论述 C++/C 编程风格，难度不高，但是细节比较多。别小看了，提高质量就是要从这些点点滴滴做起。世上不存在最好的编程风格，一切因需求而定。团队开发讲究风格一致，如果制定了大家认可的编程风格，那么所有组员都要遵守。如果读者觉得本书的编程风格比较适合你的工作，那么就采用它、改进它，不要只看不做。人在小时候如果发音不准、写字潦草，倘若不改正，长大了总有后悔的时候。编程也是同样道理。

本书的所有示例程序均在 Visual C++ 6.0 环境下编写与测试。印刷出来的示例程序并没有严格按照上述“C++/C 编程风格”书写，显得自相矛盾。但这是为了节省书籍的版面，不得已而为之。

第 11 章至 15 章是技术专题论述，难度相对比较高，看书时要积极思考。读了并不表示懂了，懂了并不表示就能正确使用。有一位同事看了第 11 章“内存管理”后觉得“野指针”写得不错，与我切磋了一把。可是过了两周，他告诉我，他忙了两天追查出一个 Bug，想不到又是“野指针”出问题，只好重读“内存管理”。

附录 D《大学十年》是个短篇传记，记录了一个程序员的成长过程，很有趣，并且能催人上进。

另外，建议大家阅读本书的参考文献，很多是经典名著。

如果你（或者团队）的编程质量已经过关了，不要就此满足，有待于提高的能力多着呢。一个企业的软件开发能力如果达到了 CMM（CMMI）3 级或更高水平，那么内功算是到家了。作者目前致力于研究和推广“CMMI 3 级软件过程改进解决方案”，相比之下“高质量编程”显得单纯多了。

读完本书后，你十有八九想看看作者后续的几本书，我先做个伏笔，电子工业出版社将陆续推出关于我们研究成果的这几本书。

欢迎读者对本书提出批评和建议，您可以以邮件 linrui@sbell.com.cn 的方式或者登录 http://www.fecit.com.cn 的网上论坛进行交流。

三、致谢

一年前我给上海贝尔网络应用事业部的开发人员们培训软件工程和 CMM，

那时公司尚缺乏内容详细的编程规范，我就写了一本“高质量 C++/C 编程指南”（书稿）当做培训教材。当时那本书稿还不到 100 页，由于我自己在编程方面没有专长，不想出版，于是就放到网上了事。书稿在网上转悠了一年，很多热心的同行们修正了书稿中的不少错误，并出乎意料地给予了一些称赞。

我到上海贝尔有限公司从事软件工程研究与实践已经将近两年了，我与合作者们取得了一些成果。我们掂量后觉得写几本书不会误人子弟，于是就联系名望较好的电子工业出版社。出版社几个小时后就决定出版，效率高得让我措手不及，那就先出版《高质量程序设计指南——C++/C 语言》吧。

我的主要合作伙伴是董军、王慧文、阙雪松、朱洪海、文少华、闵勇、顾晓刚、谢义军和彭小澎。顾晓刚和谢义军起草了本书的第 3 章“编程语言发展简史”、第 4 章“C++面向对象程序设计方法概述”和第 14 章“C++ STL 应用编程建议”。

感谢上海贝尔有限公司为我们提供了国内一流的软件工程研究和实践环境。

若干年前，在我第一次创业失败时，周鸿祎把我拉出困境，其教诲使我长期受益。我在读博士和工作期间，一直得到杭州神弓电子实业有限公司董事长高振华老先生的慷慨援助。

衷心感谢长期给予我们友好帮助的朋友们和同事们。

2002 年 6 月于上海贝尔有限公司

目录