

高质量程序设计指南 — C++/C 语言

林 锐 顾晓刚 谢义军

电 子 工 业 出 版 社

内容提要

高质量软件开发是国内计算机教育的薄弱环节，很多程序员虽然能熟练地掌握编程技术但是缺乏强烈的质量意识，不懂得在开发过程中内建高质量。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，为此付出了高昂的代价。因此，如何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。

本书论述高质量软件开发方法与 C++/C 编程规范，共十五章，分两部分：

- (1) 第一部分共四章，重点介绍软件质量和面向对象程序设计方法。
- (2) 第二部分共十一章，重点阐述 C++/C 编程风格和一些技术专题。

本书以幽默的笔调对软件工程和编程作了很好的论述。前期版本曾经在 Internet 上广泛流传，被国内 IT 企业的不少软件开发人员采用。

本书的主要读者对象是 IT 企业的程序员和项目经理，以及高年级本科生和研究生。

前言

软件质量是被大多数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，你将会受益匪浅，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 30 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作优点来欣赏。就象在武侠小说中，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，一年的心血白化了，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。我化了半个小时，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，国内大学的计算机教育压根就没有灌输高质量程序设计的观念，教师们和学生们也很少自觉关心软件的质量。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。因为发现自己连编程基本技能都未能全面掌握，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

首先请做附录的 C++/C 试题（不要看答案），考查自己的编程质量究竟如何。然后参照答案严格打分。

(1) 如果你只得了几分，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，可以进步的空间就越大，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。

第一章论述 “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。

第二章论述 “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、管理者或者老板等等, 这样的道路比较踏实。我自己这么说也是这么做的。

第三章介绍 “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。

第四章介绍 “C++面向对象程序设计方法”。如果把程序员比作武林人士, 那么 “面向对象程序设计” 是上乘的刀法, 而 C++ 则是宝刀。先讲刀法后要刀, 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, 不少已经出版的秘笈声称能让人在几周内速成高手, 作者没有秘笈, 只好写个概述, 免得献丑。

第五章至第十章主要论述 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。

第十一章至十五章是技术专题论述, 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。有一位同事看了第十一章 “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《大学十年》是个短篇小说, 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我们的研究成果。

如果你买了本书后觉得不值得，我一定赔偿你的损失。

欢迎读者对本书提出批评和建议，请用电子邮件 linrui@sbell.com.cn 与作者联系。

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）当作培训教材。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 2 年了，我和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之弟，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三章“编程语言发展简史”、第四章“C++ 面向对象程序设计方法概述”和第十四章“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。

感谢那些长期给予我们友好帮助的同事们和朋友们。

林锐

2002 年 4 月

上海贝尔有限公司

目 录

第一部分 综述

- 第一章 高质量软件开发之道
- 第二章 做好程序员
- 第三章 编程语言发展简史
- 第四章 C++面向对象程序设计方法概述

第二部分 C++/C 编程规范

- 第五章 文件结构
- 第六章 程序的版式
- 第七章 命名规则
- 第八章 表达式与基本语句
- 第九章 常量
- 第十章 函数设计
- 第十一章 内存管理
- 第十二章 C++函数的高级特性
- 第十三章 类的构造函数、析构函数与赋值函数
- 第十四章 C++ STL 应用编程建议
- 第十五章 其它编程经验
- 参考文献

第三部分 附录

- 附录 A: C++/C 试题
- 附录 B: C++/C 试题的答案与评分标准
- 附录 C: C++/C 代码审查表
- 附录 D: 《大学十年》及后记