Java

代码与架构之完美优化

——实战经典

成就架构师梦想之路

颜廷吉 编著

机械工业出版社
CHINA MACHINE PRESS

Java 代码与架构之完美优化
——实战经典

颜廷吉　编著

机械工业出版社

本书囊括了笔者多年编程经验总结出的六项编程密技：完美规约（架构大于编码）、完美视角（设计者角度）、完美利用（站在巨人肩上）、完美改造（快速编码）、完美优化（高质量代码）、完美突破（架构师之路），这六项密技是完美编程的精髓，亦是完美编程的指导思想与灵魂。本书还包含77个经典优化案例及28种常用优化技巧。本书的目的不仅仅是“授人以鱼”，更要“授人以渔”——提高读者的编程及优化能力，而这种能力正是架构师的必备技能。

本书是IT365学院网站软件架构师系列培训教程体系中的基础读本，属于品质管理实战部分的内容，是培养具有高质量代码技术水平的优秀架构师所必备的利器之一。优秀的代码品质是程序员走向架构师神圣殿堂的必经之路，本书将是这条路上的一盏明灯，帮助读者早日实现架构师之梦。

图书在版编目（CIP）数据

Java 代码与架构之完美优化：实战经典 / 颜廷吉编著. —北京：机械工业出版社，2015.5

ISBN 978-7-111-51509-8

Ⅰ. ①J… Ⅱ. ①颜… Ⅲ. ①JAVA 语言—程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2015）第 216244 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策划编辑：丁 诚 责任编辑：丁 诚 张 恒

责任校对：张艳霞 责任印制：乔 宇

北京铭成印刷有限公司印刷

2015 年 9 月第 1 版 • 第 1 次印刷

184mm×260mm • 15.25 印张 • 376 千字

0001—3000 册

标准书号：ISBN 978-7-111-51509-8

定价：49.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

服务咨询热线：（010）88379833

读者购书热线：（010）88379649

网络服务

机 工 官 网：www.cmpbook.com

机 工 官 博：weibo.com/cmp1952

教育服务网：www.cmpedu.com

金 书 网：www.golden-book.com

序

虽说网络科技对60后有些陌生，却实实在在改变着现代人的生活方式。国家诸多领域的发展也与网络科技密不可分。网络科技的发展不仅提升了产品品质，同时也在改变着世界的格局。

改革开放三十余年来，我国国民经济得到了跨越式发展，并在高精尖领域不断有所突破，核心技术变得越来越弥足珍贵，发达国家垄断的技术禁区已经不再遥不可及。去糙取精、取长补短、突破瓶颈，已成为当今高端领域的一大课题。务实态度与专业精神，作为核心价值的重要组成部分，更是当代精英不可或缺的内在追求。

颜廷吉贤弟不愧是北大的高材生，从置身Java技术领域那一刻起，始终对具有世界领先品质的相关技术情有独钟。十年磨一剑，引领同业间。他在东京近十年的一线工作中，对其核心技术领纳于心，并在实践中升华，在细节上完善，把经验汇成文字，把智慧传给同仁。矢志不渝的努力只为填补国内空缺、提升产品品质与专业人才素质，更为打造国际一流架构师队伍奉献一己之力。

作者对软件架构师培训系列教程与品质管理培训系列教程进行了整体设计，《Java 代码与架构之完美优化——实战经典》是配套培训教材之一，并附有精美的教学视频。本系列教程将成为业内不可多得的实用经典之作。有志驰骋IT行业并有所作为的业内高手精读此书，不仅可以炼就沉潜内敛的品格，提升自身的技能，而且可以开发自主品牌，更好地展现人生价值。

信息化社会使我们的生活变得快捷和丰富，而实现中国梦也离不开有识之士的敬业与专注。世界一流品质的背后是世界一流的用心。我们都向往拥有成功的人生，那么就让我们徜徉在作者的文字画面中，用真诚的心去品味和领受其中的内涵吧！

贺印普谨序

2014年10月10日

前　言

在武侠世界里，凡是顶尖的高手，要么是经历了一些奇遇偶得真传，要么就是经过几十年脚踏实地的修炼，才得以炉火纯青。虽然路不同，但他们最核心的修炼内容就是增强自己的内力。内功心法才是核心，因为剑法、拳法之类对聪明人来讲看一遍就会，而内力却很难速成。到达一定的境界后，武功就不分什么门派了，因为这都是融会贯通的。

现实科技世界与武侠科幻世界同理，Java 程序世界里也不乏顶尖高手，比如 Bruce Eckel（《Thinking In Java》作者）、Joshua Bloch（Google 首席 Java 架构师，《Effective Java》作者）、 Kent Beck（敏捷之父）、Martin Fowler（优化之父）等等。这些人之所以如此成功，并让追求者顶礼膜拜，并不是因为他们写了汗牛充栋的程序代码，而在于他们一出手就能展现绝顶高手的风范，写出令人拍案叫绝的高质量代码。因为这代表了他们的品牌与实力，他们会一直精益求精，不断优化，他们把自己的悟道——如何修炼内功秘籍——写成了书，就成了经典。

程序员修炼内功心法的终极目标就是成为我们梦寐以求的架构师。众所周知的《Java 编程思想》《Java 高效编程》《敏捷软件开发》《设计模式》《优化》《人月神话》等巨著的核心内容之一就是教授这些内功心法，本书也正是作者多年修炼的总结。品质体现于细节，本书不但从宏观到细节进行了全面系统的介绍，而且形成了代码质量优化的理论与技巧体系，是进行代码优化不可多得的宝典。

Java 是目前的主流开发技术，如何更好地发挥其技术优势实现最佳资源配置和获得更高商业价值，一直是 Java 技术发展的趋势。然而 Java 体系庞大、技术精深，如何写出优质代码，如何设计与优化系统架构，是高级开发者必须掌握的核心技术之一。

作者在 Java 技术领域从事一线工作十多年，其中 8 年时间是在东京度过的。日本的品质管理技术世界领先，作者来日之前就下定决心要掌握日本先进品质的相关技术，报效祖国——师夷长技以制夷！在这期间作者指导过 NTT、日立、富士通等日本一流 IT 开发公司主导的大型商业项目（国家就业劳动项目、全国饮料以及香烟自动贩卖机贩卖信息统计项目等）的研发与代码优化，也使得作者有机会与日本一流架构师交流及研究他们独有的架构与质量优化技术之奥妙。

学生时代的我们，所学到的是基本的技术理论知识；就业后作为初级程序员的我们，几乎不能胜任商业项目的质量要求标准，不能适应技术的深度与广度，即使能够完成的编码，也只是仓促地完成任务；经过 3～5 年磨练后的我们，回想以前写的代码，就会意识到当时写的代码是那么的不优雅！是的，根据作者实战经验，也根据通用的 8020 法则，按照要求写出可运行代码，只是完成了 80%的工作，另外 20%的代码持续优化工作，往往被大家忽略了。代码优化是一个综合的系统体系，包含的内容很多，本书将把最核心的部分进行剖析并分享给读者。

本书从实际策划到完成书稿历时近 2 年，这期间作者一直不断吸收、整理、优化、提炼其内容。在 Java 世界里编写高质量代码并非易事，各种开源代码检测工具也对各种技术细节

进行了规定，Checkstyle 里有 134 项，FindBugs 里有 408 项，PMD 里有 368 项，Jtest 里大概有 1000 项，面对这么多的规则，可能有人会说这还怎么下手写代码啊？是的，学习与研究这些枯燥无味的规则估计就要花掉很多时间，而且即使学习了也不一定能记住，况且有些规则是站在技术研究者的角度而做出的，在实际商业上根本用不到。基于实用原则，作者从这些代码检查规则中提炼了经典内容，吸收了日本代码优化培训以及品质管理培训内容精华，深入研读了本书参考文献里所列的书籍以及网络上各种资源，并结合本人多年的实战经验，提取出了 100 个有代表性的经典优化实战案例，本书素材来源如图 1 所示。后来又经过进一步分析整理，精选了最有代表性的 77 个案例，希望读者可以用最少的时间学到最有价值的代码优化技术。

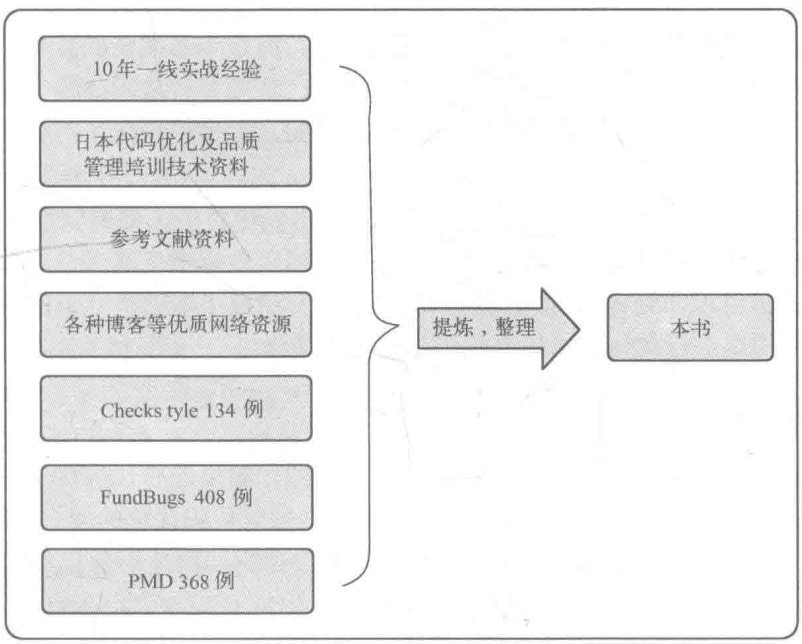

图 1　本书素材来源

本书与软件质量

故事一：完美我们做得到么？

曾经有一家美国公司到日本一家小企业定制了 10000 件某种电子产品部件，要求次品率是 1/1000。这家企业在规定的时间内邮寄去了产品，美国公司产品部打开一看，有些吃惊——怎么有两个盒子？一个大的，一个小的？再打开一看，大的里面放满了产品，同时有一个说明：全部合格产品 9990 件；另外一个小盒里放的是什么，大家就不言自明了。

感悟：100%的合格就是一种完美，我们能做到么？日本之所以在软件行业乃至其他很多行业都做得如此出色，很重要的一点就是因为质量做得绝对世界第一。而我们国家则基本是粗放型经济，无论从对产品质量的认识程度，还是产品质量的管理方法，都离日本尚有一段很大的距离。希望本书的问世可以缩小这段距离。

软件质量（品质）包含的内容在我们规划的教程体系中，分为开发篇和管理篇，前者就

是本书所包含的内容。

谈到代码质量，首先要辨析什么样的代码是劣质代码，什么样的代码是优质代码，可以用哪些工具自动检测劣质代码，以及如何把劣质代码转化成优质代码，本书将带领读者一起解析。

物皆有其位，而后物尽其位。世道乱了，就要治理，使其有序。同理，Java 世界里有很多保留字，如果在保留字所控制区域里写有不合理的代码，那么将会给软件带来灾难性的后果。因此，每段代码都应该在其所在的位置——如果不在其位，那就需要优化。优化是确保代码质量的重要手段之一。本书不但系统展示了常见劣质代码，而且总结了 28 种常用优化技巧，并给出了优化的具体步骤，向读者演绎了从劣质到优质的变化过程，可谓理论与实践的最佳组合。

一个项目的成功不仅仅是靠某一个程序员的能力，而是项目团队的默契配合与努力的结果。要保证全体开发人员统一的编程风格与代码质量，就需要有编程规约来进行思想统一。虽然已有很多论述编程规约的文章，但要么太钻牛角尖、太偏理论化了，要么太粗浅、不够精细，没有太大实际商业价值。本书附录里给出的 Java 编程规约、JSP 编程规约、CSS 编程规约是作者经过多年实战经验所总结的，可以直接应用到项目中，提高软件质量。

无规矩不成方圆，任何商业系统，都应该坚持遵循工业标准。本书是优质代码编写规范的提炼与总结，也可以作为评判优秀代码的标准之一。违反了本书所介绍的规则之代码，需要根据项目实际情况进行优化，以提高代码与软件的质量。

本书与软件架构师

一款软件的最终体现就是代码，而作为软件架构师，如果没有代码优化的意识与技术，就不能称之为软件架构师。任何合格的软件架构师，必须对代码优化的概念与技术烂熟于胸，信手拈来。一个连代码质量都不能控制好的架构师所设计的架构是不会有人信任的。

本书囊括了作者多年编程经验总结出的六项编程密技：完美规约（架构大于编码）、完美视角（设计者角度）、完美利用（站在巨人肩上）、完美改造（快速编码）、完美优化（高质量代码）、完美突破（架构师之路），这六项密技是完美编程的精髓，亦是完美编程的指导思想与灵魂。本书的目的不仅仅是“授人以鱼”更要“授人以渔”——提高读者的编程及优化能力，而这种能力正是架构师的必备技能。

本书是 IT365 学院网站软件架构师系列培训教程体系中的基础读本，属于品质管理实战部分的内容，是培养具有高质量代码技术水平的优秀架构师所必备的利器之一。优秀的代码品质是程序员走向架构师神圣殿堂的必经之路，本书将是这条路上的一盏明灯，帮助读者早日实现架构师之梦。

本书与翻转课堂模式

翻转课堂（The Flipped Classroom），是 2011 年在美国兴起的新型教学模式。与传统的课堂教学模式不同。在“翻转课堂”教学模式下，学生课下完成知识的学习，而课堂变成了老师与学生之间和学生与学生之间互动的场所，包括答疑解惑、知识的运用等，从而达到更好的教育效果。教与学颠倒，即翻转课堂，它颠覆了传统意义上的课堂教学模式。

本书充分参考了“翻转课堂”模式，在内容安排上，对于每一个案例，都是先展示劣质

代码（优化前代码），也是有意让读者自己研究其瑕疵在哪里；之后给出优质代码（优化后代码），进一步让读者感受优质代码带来的效果；最后给出从劣质到优质演化过程的解析。这样在阅读技巧上进行了革新，可以让读者更好地吸收与理解本书精华。

本书配套源码

本书的配套源码以及其他模板文件可以在www.365itedu.com（本书官网）下载。

在源码资源里，例如 best0101 包下面的代码为优化前的劣质代码，一般类的命名为 Before，测试代码类的名称为 BeforeTest；而 best0102 包下面的代码为经过优化后的优质代码，一般类的命名为 After，测试代码类的名称为 AfterTest，如图 2 所示。所有代码均是可执行代码，读者可以随时运行查看优化前后执行效果。

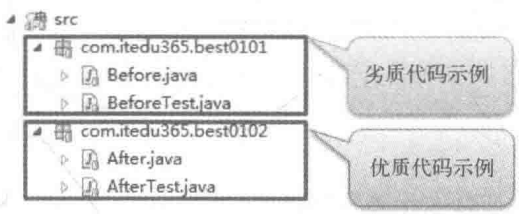

图 2　源代码说明

书中所选用代码比较简单，并对相对复杂的代码进行了背景说明。读者在学习过程中，可以自己先琢磨劣质代码的瑕疵在哪里，自己是否可以发现？可以用哪些手段来发现？如何优化？带着这些问题阅读本书，会达到更好的学习效果。

本书配套教学视频

传统教育模式中对程序设计的教学方式大多已经落伍，这使得学生在真正进入社会就业前不得不再进行二次培训；而现在社会上的培训机构也是鱼龙混杂，培训的内容很多也只是基础的入门级别。因此，需要一套从实际项目研发中提炼而出的、具有实际价值的、拿来就能用的、同时具有前瞻技术设计与研发引导性的高级软件架构师培训教程。本书配套培训教程与教材同步出炉，可以帮助读者更好、更轻松地学好本书的精髓。配套培训教程也可以到 365IT 学院下载。

本书特色

1．内容精炼、超值：作者将十多年一线实战心得倾情奉献——说代码之美是重实用，讲架构优化是接地气。

2．授人以鱼，不如授人以渔：作为编程指导思想与灵魂的六项密技，可全面提高读者自身的编程与代码优化能力。

3．案例驱动，脚踏实地：不讲单独代码片段，而是以案例驱动（根据实际商业代码提炼之后的可运行代码）进行实战解析；不仅仅是经验与理论的总结，更重要的是用最直接的案例代码来说明技术应用方式。

4．图解技术，形象生动：利用图解方式避免了乏味难懂的文字描述，使繁冗复杂的事物

一目了然，是对理论进行深刻理解的形象记忆。“图+代码”是技术学习的最佳方式。

本书所面向的读者

故事二：敬业之神

敬业之神——野田圣子。

她 37 岁就当上了日本内阁邮政大臣。

她的第一份工作是刷厕所：她把厕所刷得光洁如新，一尘不染，她直接把冲厕所的水舀一勺一饮而尽，来证明其工作质量。

她有一句名言：就算这一辈子都在洗厕所，也要当个最出色的洗厕人。

敬业之神的启示：通过不断修炼内功，不断优化，刷厕所都可刷成行业的佼佼者！高端的品质目标，究极的爱业敬业态度，给予了我们极大的心灵震撼与鼓舞！——想出人头地么？想有所成就么？那么把一件事做好，把其品质做到极致，就成功了！

本书所面向的读者主要是那些想在技术领域成为佼佼者的朋友：

1．走在架构师之路上的工程师。

2．希望提高自己代码质量水平的程序员。

3．追求完美的技术爱好者。

总之，无论是在校大学生还是刚刚走上工作岗位的新员工，无论是做编码的程序员还是做测试的技术人员，无论是架构师还是项目经理，都可以从本书中获得有益的收获。

如何阅读本书

本书的各个章节都有一定的独立性，之间也有相互的关联性，如图 3 所示。

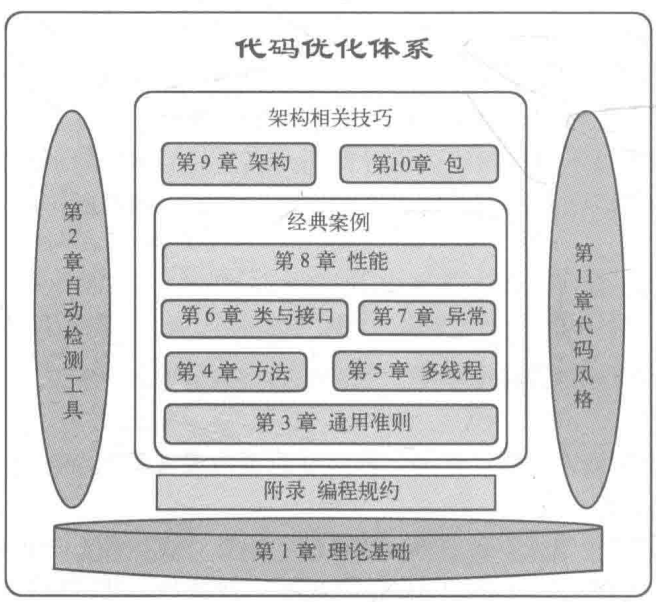

图 3　代码优化体系构成

第 1 章主要解释与代码质量相关的理论基础，其余章节都是介绍与代码优化相关的技巧体系。

第 2 章介绍了自动代码质量检查工具的原理与实战技巧。

第 3 章到第 8 章主要介绍了代码质量优化经典案例解析，从编码通用准则、方法、多线程、类与接口、异常以及性能方面进行了详细分析说明且都有可用自动工具进行检查的案例。其中，第 3 章是代码优化的基础，也是其他章节的基础，读者对第 3 章所述内容掌握程度的优劣，将直接影响到其他章节代码优化的效果；而第 3 章到第 8 章经典案例代码优化的效果，又直接影响到第 9 章与第 10 章架构优化的效果。

第 9 章与第 10 章主要论述了架构、包优化的各种技巧与原则。

第 11 章介绍代码风格，优秀的代码还需要有很好的展现形式。这一章详细说明了优化代码风格所需的模板技术以及优化代码的手段。代码风格体现于整个代码文件，因此本章是第 3 到第 10 章优化内容的一个升华。

本书正文对于代码质量优化进行了宏观的介绍，附录中的编程规约（Java、JSP、CSS）则更加详细地描述了代码优化实践经验，是对本书代码优化技巧的补充，使本书形成了全面系统的代码优化体系。

本书每个经典案例，都是对相关技术的总结。对于每个案例，都从优化前代码、现象描述、不利影响分析、检测工具或方法、最佳解决方案、优化后代码六个方面进行了剖析。如果本案例涉及优化技巧，那么会对优化技巧从优化类别、实施方法等方面再进行深度解析，让读者更加深刻地理解案例所阐述的相关内容。同时，对案例中重要的技术点也做了提示，以引起读者的注意，加强相关技术点的理解与重视。

本书经典示例代码，不仅是给读者介绍技术与知识的演示，更重要的价值在于成为读者自我提高与演练的绝佳手段。在阅读的过程中一定要亲自动手练习各个案例（建议圈点出劣质代码的瑕疵并做注释，评论出优质代码的亮点），经过“思考→实践→提炼”之后，可更好的转换成自己的技术与知识体系，达到融会贯通的境界。

另外，为了突出说明各代码所要体现的主题，本书所示例代码并没有严格按照一般代码编写规范要求，进行标准的代码注释，只是为降低代码的理解难度，适当地添加了一些必要的注释。为了精简代码，有些案例代码的 import，set/get 部分也进行了省略。

术语解释

1. 成果物：最终要交付给客户的成品资料。
2. 类型编码：简称类型码，是数据字典的一种（如 1 代表男，2 代表女）。
3. 正常系代码：处理正常业务逻辑的代码。
4. 异常系代码：处理异常部分的代码。
5. 饿汉式初始化：指对象在类装载时构建，急切初始化，不调用也创建。
6. 懒汉式初始化：指对象在第一次被使用时构建，延迟初始化，调用时才创建。

符号说明

文中为了标注的方便，用了一些简略符号，总结见表 1。

表 1　本书简略语

符　号	说　明
（C）	利用 CheckStyle 检查是否违反规约
（F）	利用 FindBugs 检查是否违反规约
（P）	利用 PMD 检查是否违反规约
（R）	利用 Review 人工检查进行检测
（EC）	利用 Eclipse 的格式模板或其自身的编译设定进行检测
（EM）	利用 Emma 代码覆盖率工具进行检测

在“检测工具或方法”部分，用符号码来代替内容，而且符号码后面紧跟各个工具检查出的错误名词，以便查询。

致谢

首先，感谢同事尹勋成、周伟鹏、王超、袁宏等分享的工作经验以及对本书写作提出的建议。

其次，要感谢忘年之交贺印普兄长在百忙之中抽出时间为本书作序，以及友人宋海燕对本书进行的阅读体验反馈。

最后，感谢家人的大力支持。作者为了本书及其配套视频早日和读者见面，两年来占用了几乎所有的休息时间进行写作。特别是 2014 年以来，作者为了尽快完成此书，请假半年，专研创作，排除了所有的外界干扰，以期能够早日完成一本优秀的教程。家人的默默支持，哪怕是深夜的一杯热茶，睡前的一声叮嘱，都是作者前进的动力。作者不分日夜地敲击着键盘，推敲着每一个实例代码，斟酌着是否还可以再优化，是否可以更好地把技术展现给读者，连陪伴家人散步都成了一种奢侈，对此家人毫无怨言，一直对作者支持鼓励。待到本书完成时女儿也从 2 岁开始可以叫“爸爸”，到现在 4 岁可以撒娇拉着手要去买冰淇淋了，时光如梭，现在终于可以轻松一下，和女儿分享家庭的快乐了。

读者在阅读过程中如果发现任何疑问，可以与作者通过以下方式联系。

QQ：25846482

微信：itedu-365

E-mail：yantingji@126.com

颜廷吉

2014 年 10 月 18 日于东京

目　录

第1章 代码质量

美好的东西在质不在量。——伊索

俗话说，工欲善其事，必先利其器。高质量的代码就是程序员驰骋疆场的锋利武器之一。在进行优化代码之前，首先让我们认识与了解与代码质量相关的概念。

1.1 什么是代码质量

质量是产品或服务的总体特征与特性，即在使用时能成功满足用户需要的程度。简而言之，质量是满足需求的能力。因此，代码质量也是满足一种需求的能力，这里的需求来源不仅包括客户，也包括各种技术人员（程序员、测试员、维护员等）。

1.2 什么是软件质量

软件质量是满足客户软件需求的能力。高质量的软件产品应该符合用户需求、运行稳定、性能优异、维护简单、文档齐全。

软件质量一般具有以下衡量标准：

1. 可用性，是指软件系统能够正常运行的时间比例，除了维护时间外，一般都要求系统可以提供正常服务。

2. 功能性，是指软件系统能为我们完成所期望工作的能力。

3. 易用性，是衡量用户使用软件产品完成指定任务的难易程度，也就是用户体验，或系统的柔软度与亲和力。

4. 性能，是指软件系统的响应能力，即要经过多长时间才能对某个事件作出响应，或者在某段时间内系统所能处理的事件个数。

5. 可靠性，是指软件系统在错误面前或者使用错误的情况下维持软件系统功能特性的能力。

6. 健壮性，是指在业务处理或者运行环境中，软件系统能够承受的压力或者变更能力。

7. 安全性，是指软件系统向合法用户提供服务的同时，能够阻止非授权用户使用或者拒绝服务的能力。

8. 可维护性，是指体系结构扩充或者应对需求变更的能力。

1.3 代码质量与软件质量

故事三：程序员的需求

有人问程序员还有什么需求？

程序员甲：客户是上帝，我们又不是。

程序员乙：可我曾听一个程序员说，他编写代码有时能体会到上帝造物的感觉（对客户需求的满足），也能体会到别的程序员的心声（对既存代码的理解）……

这个故事告诉我们：

其一，我们研发的软件就是要满足客户的需求，如果代码跟客户需求不对应，不能满足客户的需求，这就对软件的质量造成了影响，即使代码写得漂亮，也不能算是高质量的代码。

其二，客户的需求发生变动或者增加，程序员需要改动代码，“看得懂容易改”就是程序员的需求。这种需求虽然对软件质量没有直接的影响，但是会耗费将来的人力成本。

所以说，代码质量是软件质量的组成部分。开发人员写的代码质量越高，缺陷（以下简称 Bug）就会越少，即使有 Bug 也容易找到；反之代码质量越低，Bug 就会越多。

软件质量不好，迟早有一天会被用户抛弃；同样，代码质量不好，迟早有一天也会被它的需求来源所放弃。最糟糕的状况是，自己被自己的代码抛弃，陷入泥潭无法自拔。软件质量与代码质量的关系如图 1-1 所示。

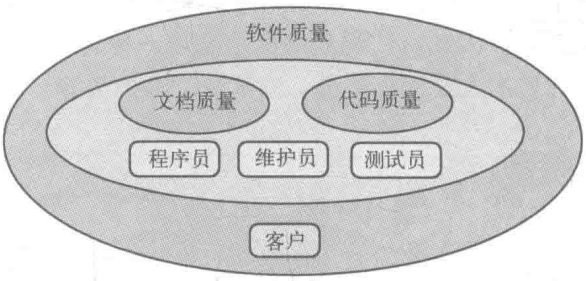

图 1-1　软件质量与代码质量

1.4　代码质量优化理论

高质量代码一般具有以下特性：

1．高可用性：正确、有效、及时地满足客户需求，写出能完成软件功能需求的代码。

2．高可读性：高可读性就是层次清晰又有良好注释的代码。代码是具有个人色彩的，每个人的思想是不同的，写出来的代码不会是完全相同的。繁冗的代码，特别是没有注释的代码，相信大家都不乐意去看。

3．高可测试性：是指软件发现故障并隔离定位其故障的能力，以及在一定的时间或成本的前提条件下，进行测试的能力。

4．高可扩展性：这一点对于有多年工作经验的程序员来说认识会较深，因为客户的需求是随时变化的，所以代码就要预留以后变更需求的空间。

5．高可维护性：软件研发完成，是研发阶段的终止，却是软件运营维护的开始，这需要一个团队长期运作，高可维护性的目的就是要节省运营维护成本。

以上特征都是我们进行质量优化的目标，也是优化之后给我们带来的真实之利，它们之间的关系如图 1-2 所示。可用性可以说是代码优化的最基本要求，只有首先满足了可用性，我们才有资格谈其他特性；如果代码不可读，可维护就无从谈起，因此可读性是可维护性的