Broadview®
www.broadview.com.cn
PEARSON

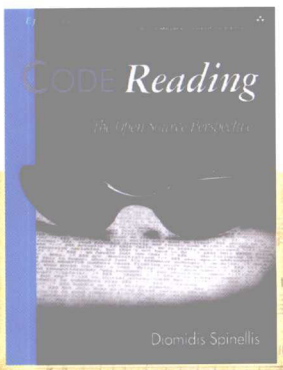
Jolt大奖精选丛书

代码阅读

Code Reading

The Open Source Perspective

[希]Diomidis Spinellis 著
左飞 吴跃 杨宁 译



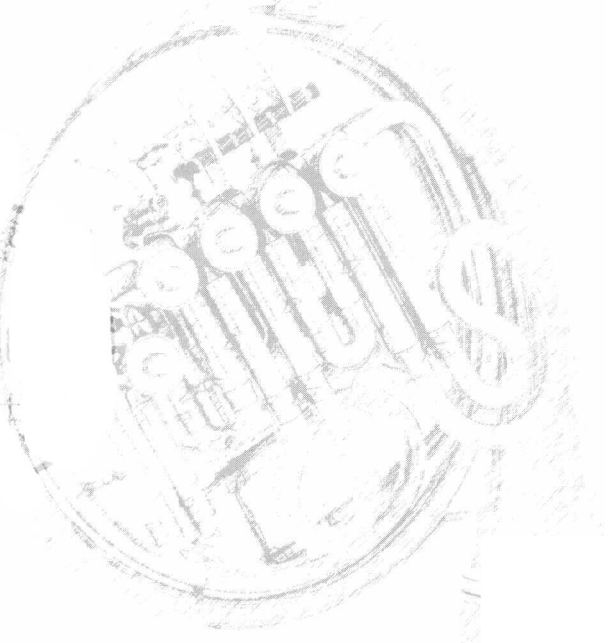
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Jolt 大奖精选丛书

代码阅读

Code Reading
The Open Source Perspective

[希] Diomidis Spinellis 著
左飞 吴跃 杨宁 译



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

内 容 简 介

Jolt 大奖素有“软件业之奥斯卡”的美称，本丛书精选自 Jolt 历届获奖图书，以植根于开发实践中的独到工程思想与杰出方法论为主要甄选方向。作者使用了超过 600 个现实的例子来向你展现如何甄别代码的好坏；如何阅读，应当注意什么，以及如何使用这些知识来改进自己的代码。本书在一些现实中的大型实例基础上，论述了代码阅读的策略，并向读者展示了如何将这些代码阅读和代码理解的技艺运用于实践。

本书荣获 2003 年 Jolt 世界图书大奖，参阅本书对于大专院校相关专业的师生、计算机领域的从业人员或程序设计爱好者都大有裨益。

图书在版编目（CIP）数据

代码阅读/（希）斯宾耐立思（Spinellis,D.）著；左飞，吴跃，杨宁译.—北京：电子工业出版社，2012.8
（Jolt 大奖精选丛书）
书名原文：Code Reading : The Open Source Perspective
ISBN 978-7-121-17481-0

Ⅰ. ①代… Ⅱ. ①斯… ②左… ③吴… ④杨… Ⅲ. ①程序设计－代码 Ⅳ. ①TP311.11

中国版本图书馆 CIP 数据核字(2012)第 143822 号

策划编辑：张春雨　符隆美
责任编辑：白　涛
印　　刷：北京丰源印刷厂
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：787×980　1/16　印张：27　字数：473 千字
印　　次：2012 年 8 月第 1 次印刷
定　　价：79.00 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。

出版说明

经久不息的回荡

今时的读书人，不复有无书可读之苦，却时有品种繁多而无从择优之惑，甚而专业度颇高的技术书领域，亦日趋遭逢乱花迷眼的境地。此时，若得觅权威书评，抑或有公信力的排行榜，可按图索骥，大大增加选中好书的命中率。然而，如此良助，不可多得，纵观中外也唯见一枝独秀——素有“软件业奥斯卡”之美誉的 Jolt 奖！

震撼世界者为谁

在计算设备已经成为企业生产和日常生活之必备工具的今天，专业和大众用户对于软件的功能、性能和用户体验的要求都在不断提高。在这样的背景下，如何能够发挥出软件开发的最高效率和最大效能，已经是摆在每一个从业者面前的重大课题，而这也正是 Jolt 大奖横空出世的初衷及坚持数年的宗旨。

Jolt 大奖历时 20 余年，在图书及软件业知名度极高，广受推崇。奖如其名，为引领计算机科学与工程发展主流，Jolt 坚持将每年的奖项只颁给那些给整个 IT 业界带来震撼结果的图书、工具、产品及理念等，因一流的眼光及超高的专业度而得以闻名遐迩，声名远播。

除图书外，Jolt 针对软件产品设有诸多奖项分类，如配置管理、协作工具、数据库引擎/数据库工具、设计工具/建模、开发环境、企业工具、库/框架、移动开发工具等。但图书历来是 Jolt 大奖中最受瞩目且传播最广的一个奖项分支。Jolt 曾设有通用类图书、技术类图书

等分类，每个分类又设有“卓越奖”（Jolt Award，一般为一个）和“生产力奖”（Productivity Award，一般为 2 或 3 个）。获奖技术图书一经公布，即打上经典烙印，可谓一举“震撼全世界”（赞助商 Jolt 可乐的广告词）。

作为计算机技术图书的厚爱者，我们总在追问——是谁在震撼世界，是谁在照亮明天？Jolt 大奖恰似摆在眼前的橱窗，让我们可以近距离观看潮流在舞蹈，倾听震撼在轰鸣！

朝花夕拾为哪般

Jolt 像是一年一度的承诺，在茫茫书海中为我们淘砺出一批批经得起岁月冲刷的杰作，头顶桂冠的佳作也因而得以一批批引进中国，为国人开阔了眼界，滋补了技术养分。然而，或因技术差距造就的生不逢时、水土不服，或因翻译、制作的不如人意，抑或是疏于宣传等诸多原因，这些经典著作在国内出版后，尽管不乏如获至宝的拥趸，却仍不为诸多人所知，从而与大量本应从中获益的读者擦肩而过。既然这生生错失的遗憾本不该发生，则更不应延续。为此，我们邀国外出版同行、国内技术专家一道，踏上朝花夕拾之路，竭力为广大读者筛选出历久弥新、震撼依旧的 Jolt 图书精品。

Jolt 获奖图书皆由业界专家一致评出，并得到软件从业人员的高度认可，虽然这些书今天读来，不再能看到上世纪史诗时代那般日新月异的理论突破，以及依赖于高深繁复的科学研究所取得的系统化成果，更多是在日复一日的开发实践中总结和提炼出来的工程思想和方法论。重新选材之所以有所弃取，从 Jolt 多年来的评奖规律中可窥端倪——

一万小时真理见

凡是在工程思想领域取得革命性、颠覆性突破的图书，就被归于“震撼”获奖分类。比如，从基于过程的程序设计模型过渡到面向对象的全新模型，就是软件开发思想上的一次带来巨大震撼的革命；再比如，打破传统的瀑布模型而转向持续集成的软件交付模型，这也是一场业界的重大思想转变。像这样的重大思想突破，可以说是数年甚至数十年一遇的，而荣获 Jolt 大奖的图书中更为常见的，则是基于最佳实践的“生产效率”获奖者。获得此类殊荣的图书，都是作者们从平凡的、重复的，甚至用一般人的眼光看来不怎么起眼的日常开发实践中，以独具的慧眼、过人的耐心和大胆的创新，闯开一条不平常道路的心血与经验总结。

这些图书所涉及的主题，都是普通的软件开发人员每天要面对的工作——代码阅读、撰

写测试用例、修复软件问题……但就是这样貌似平淡无奇的工作，是否能每一天、每一个项目都做好，着实拉开了软件开发人员素质的差距，也决定了软件企业开发出来的产品和服务的质量。我们中国有一句古话，叫做熟能生巧；某位著名企业家也说过一句家喻户晓的名言：“把简单的事千百万次地做好，就是不简单的。”这些朴素而实际的真理，同样也是本套丛书最能彰显的所谓程序员精神。它建立在脚踏实地的实践基础之上，也充满了对于自由和创新的向往。

名作可堪比名曲

就不因岁月流逝而褪色来说，与这些 Jolt 名作相媲美者，只有那些百年响彻、震撼古今的经典名曲。希望本丛书带给大家的每部著作，也如百听不厌的乐曲，掩卷良久方余音绕梁，真知存心。仔细想来，软件开发与古典音乐岂非有异曲同工之妙？既是人类心智索问精确科学的探究，亦是寻觅美学享受的追求。工程是艺术的根基，而艺术是工程的极致。衷心地希望各位读者能够认真阅读本丛书的本本珍品，并切实地用于自己的日常工作中，在充分享受大师魅力的同时，为中国的软件事业谱写更多、更震撼的乐章。

电子工业出版社博文视点

二〇一二年春

Preface to the Second Chinese Edition of Code Reading and Code Quality

China is the first country that honors my two *Open Source Perspective* books with a second edition. There are many possible reasons for this, but a particularly appealing one is associated with the heritage of Confucius, who in the Analects extensively stresses the importance of study. In retrospect I realize that when authoring the books *Code Reading* and *Code Quality* I was in fact following his sage advice by prompting my developer colleagues and my students to improve their knowledge and skills by studying software code.

In *Code Reading* I explain how we can read existing code. There are many utilitarian reasons for reading code that others have written: to fix problems, to add features, to scavenge for useful parts, or to review it as part of your organization's quality control procedures. However, the noblest reason for reading code is to learn from it. From existing high-quality code we can learn how a rigorous code style is applied in practice, how to write useful comments, how to format code so that others can understand it, how to select meaningful identifiers, and how to structure complex programs into manageable parts. By studying code we can also learn new algorithms, APIs, and architectures. In short, we can become better programmers.

In *Code Quality* I take a step back to look at the larger picture. When code is assembled into programs it gives rise to so-called emergent properties: reliability, security,

CPU utilization, space consumption, portability, and maintainability. Although these properties may appear abstract and difficult to pin down, it turns out that by studying expertly-written code we can learn many best practices that benefit the code's quality. We can see rigorous error testing, the conservative handling of security credentials, efficient algorithms, nifty data coding schemes, clever abstraction techniques, and essential design patterns applied in practice. In short we can become even better programmers.

I therefore recommend to take time to consciously study existing code and learn from it. For as the Great Philosopher wrote: "To study and not think is a waste. To think and not study is dangerous."

Athens, September 2011

原作者中文版序

中国是首个将我的“开源视角”系列作品再版的国家。这可能有很多原因，而其中一个特别吸引人的原因与孔子的著作有关，他在《论语》中广泛地强调了学习研究的重要性。回顾之前撰写《代码阅读》和《代码质量》[1]的历程，我了解到，实际上，我鼓励了那些从事开发工作的同事和学生借助研究学习软件代码来提升他们自身的知识和技能，这正是我遵循孔子金玉良言的一种方式。

《代码阅读》一书阐述了开发者应当如何阅读已有的代码。关于为何要进行代码阅读，不少人也给出了许多现实中的原因：修正问题、添加特性、寻找有用的片段，或者作为你所在机构质量控制流程的一部分对其进行复查。然而，进行代码阅读最重要的原因其实是从中学习。从已有的高质量代码中，我们可以学习如何将严谨的代码风格应用于实践，如何编写有用的注释，如何编排代码以方便他人读懂，如何选择有意义的标识符，以及如何将复杂的代码组织为可管控的部分。另外，通过研究代码，我们还可以学习到新的算法、API 及架构。简而言之，阅读代码可以帮助我们成为更加优秀的程序员。

《代码质量》一书，退后一步，方为大观。当代码被组装为程序时将会产生所谓的聚现属性[2]：可靠性、安全性、CPU 利用率、空间占用率、可移植性及可维护性。尽管

[1] 《代码阅读》和《代码质量》的英文名分别为 *Code Reading* 和 *Code Quality*。

[2] 聚现属性是一个应用于科学、系统论、哲学甚至艺术学等领域的专业术语。它表示事物的特性是由于系统各部分的协同联合产生的，而并非属于系统中的任何一个部分。换言之，聚现属性是系统中各部分无法独立表现出来的属性。——译者注。

这些属性可能看起来抽象且难以约束，但是诸多成功的经验表明，借助研究专家级的代码，我们可以学习到许多极好的提升代码质量的方法。我们可以从中发现严谨的错误测试、安全证书的保守处理、高效的算法、灵巧的抽象技术及应用于实践中的基本设计模式。简而言之，这将有助于我们成为极其优秀的程序员。

伟大的哲学家孔子曾经提过："学而不思则罔，思而不学则殆"。因此我推荐大家主动花些时间来研究已有的代码并从中予以学习。

Diomidis Spinellis

2011 年 9 月于雅典

推荐序

今年恰逢“十二五”开局之年，在全球软件技术和产业格局孕育重大调整之际，我国软件产业也在工业化、信息化“两化融合”的大背景下迎来了又一个快速发展的新阶段，这其中机遇与挑战并存。当下软件和信息服务业市场的规模不断扩大，物联网作为又一个万亿元级别的产业将产生千亿元级别的服务外包。预计到2020年，全球潜在的服务外包市场需求将达到1.65～1.8万亿美元，大力发展软件业及信息服务业将成为各国抓住新机遇、全面深度参与全球化、提升软件产业技术力量的重要途径。目前我国软件产业规模虽已过万亿元，但在核心技术、基础软件等方面仍有很大发展空间。

高素质人才的储备是推进产业健康快速发展的根本保证。高端软件人才的大量持续涌现，关键在于教育，这其中高校无疑要发挥重要的作用。我们高校软件教育者既要继续贯彻党的教育理念，进一步深化我国高级软件人才培养体系的发展进程；同时又要看到我国与欧美等高水平软件人才教育国家之间的距离，师夷长技，以求在全球化浪潮中谋得一席之地。作为一位优秀的软件教育者，Diomidis Spinellis教授的某些理念无疑是非常值得我们学习和借鉴的。他以人类学习自然语言的认知规律为出发点，独辟新径，强调借助代码阅读来提高编程能力。目前这一思潮也已逐渐由欧美向我国渗透。

代码阅读是每一个软件从业人员经常进行的活动，其重要性对于每一个开发者不言而喻，但人们更多的是在本着修改前人代码而进行此项活动的，换言之，仅仅使用了代码阅读的工作属性，而未见开发其学习属性。其实，代码阅读还帮助人们完成了

“观察—模仿—创造”这样一个过程的初始阶段。南朝刘勰《文心雕龙》里讲“观千剑而后识器”，与之类似，清乾隆年间蘅塘退士还说“熟读唐诗三百首，不会作诗也会吟”，这都是强调了观察对于之后创造的重要性(巧合的是写诗和写代码的观察都是借助阅读来完成的)。

令人遗憾的是，一直以来，许多人都认为阅读代码不是件容易的事情，不仅不容易，很多时候还非常枯燥；即使是自己写的代码，有时隔一段时间再回顾也会不知所云。很多人在自己的编码生涯中都或多或少有过一些阅读代码的经历，有自己的一些方法，但也仅仅是一些个人实践而已，缺乏对整体的把握，经常是只见树木不见森林（很多时候仅仅能看到一小部分树木)。为了在学习的过程中少走一些弯路，业界代码阅读与质量提升方面的开宗明义之作——Diomidis Spinellis 教授所撰写的两部经典之作《代码阅读》和《代码质量》无疑是推荐给每位从业人员的理想读物。这两部曾荣获美国 Jolt 软件开发震撼大奖的作品，影响了一代程序员，是相关领域中的经典名作。

我阅读了两部书的译稿，并非常欣喜地将它们推荐给每一位读者。该书译者和编辑们严谨、认真的工作使得本版最大程度地还原了作者的原意，相信经由他们的辛勤努力，必将能为广大读者献上一道惊艳的佳作。

欧阳修说：“立身以立学为先，立学以读书为本。”衷心希望广大读者借由本书立学解惑，提升自我。

李战怀[1]

于 2011 岁末

[1] 李战怀，西北工业大学教授、博士生导师、计算机学院副院长，兼任中国计算机学会数据库专业委员会副主任和中国计算机学会信息存储委员会常务委员。

译者序

软件产业无疑是本世纪最具广阔前景的新兴产业之一。在该领域，中国拥有数以千亿计的市场规模，更有无数富有才华和热情的年轻从业者。作为一种“无污染、微能耗、高产值且劳动力密集”的产业，软件产业不但能大幅度提高国家整体经济运行效率，而且自身也可以形成庞大规模，拉升国民经济总体水平。进入新世纪以来，中国的软件产业迅猛发展，成果固然可喜，但我们也应清醒地看到中国软件产业同美国、日本、印度等软件强国之间的差距。随着信息技术的发展，软件产业将会成为衡量一个国家综合国力的标志之一。因此，发展和扶持软件产业，是一个国家提高国家竞争力的重要途径，也是参与全球化竞争所必须占领的战略制高点。为鼓励和促进软件产业的健康、可持续发展，国务院先后印发了《鼓励软件产业和集成电路产业发展的若干政策》和《进一步鼓励软件产业和集成电路产业发展若干政策》两项通知，以及《国务院振兴软件产业行动纲要》等纲领性、指导性文件，力求在产业政策、人才培养及其他相关配套等各方面形成合力，共促中国软件产业大发展。

作为推进中国软件产业大发展战略布局中的重要一环，软件人才培养始终是万业之基、重中之重。为此，教育部于 2001 年在全国顶尖高校中组织成立了 35 所国家示范软件学院。国家示范软件学院本着“开拓创新、改革示范、育人为本、质量为先、面向产业、走向世界”的理念，以培养“国际化、工程型”人才为目标，在软件人才培养的大路上披荆斩棘、一路求索，先后为国家输送了十万余名高素质软件人才，摸索出了一整套行之有效的软件人才培养体系，为中国软件产业发展提供了源源不断的动力。

时光荏苒，2011 年，国家示范软件学院成立 10 周年庆祝大会在北京隆重举行。包括国家部委领导、教育界和软件企业界的耕耘者们、外国专家学者在内的业界精英济济一堂，共同庆祝这承载了无数光荣与梦想的历史时刻。作为国家示范软件学院的优秀毕业生代表，我也有幸参加了此次盛会。会上有关领导及专家充分肯定了国家示范软件学院的办学成果，同时寄予中国软件产业和软件教育事业以厚望。

科教可以兴国，中国软件产业的可持续发展，关键在于人才。要培养出更多高质量、国际化、工程型的软件人才，高校与社会同样承担有艰巨而光荣的使命。我们既要继续发扬和巩固已经取得的成果，推进中国软件人才培养的科学化体系建设，同时也应看到中国同世界上其他软件强国在软件教育方面的差距，取人之长、以补己短。作为当今世界上的头号科技强国和世界软件产业的领跑者，美国在软件人才培养方面同样有许多可贵的经验和成果值得我们借鉴。其中，关于软件技术的许多真知灼见已经凝聚成了一本本经久不衰的科技专著。作为一名技术作家、译者，我们一直希望同中国的 IT 出版业一道将这些承载着先进思想的著作介绍给国人，若能以这种方式为中国软件产业和软件教育事业略尽绵薄之力，吾辈也必将倍感欢欣。

由国际知名的 Addison-Wesley 出版社推出的“高效软件开发系列”丛书为现代软件开发的方方面面提供了专业的建议和意见。收录在该系列中的书籍本本都是技术方面声名卓著的佳作，这些书籍的作者在创作时煞费苦心，力求作品篇幅适中、易于阅读，同时保证作品的价值能够历久弥新，而不会随时光的流逝而渐显黯淡。系列中的每一本都描述了一项软件开发的核心话题，这些内容可能是业界专家们在开发中始终秉承的，也可能是需要本书的读者们引以为戒的，而之所以这样做的目的只有一个，就是要创造出最杰出的软件。Diomidis Spinellis 教授的两部著作《代码质量》和《代码阅读》均收录在此系列中。在 IT 产业蓬勃发展的今日，电子工业出版社顺应时代发展和广大读者希冀，隆重推出了“Jolt 大奖精选丛书”，《代码阅读》和《代码质量》再次被收入其中。经典之作《代码质量》得以拨云开雾，同广大中国读者见面。

作者的这两本书都可谓是在业界独树一帜的经典之作。在学习如何编写代码之前，应当首先学习如何阅读代码，因为众所周知，学习其他语言方法都是先学阅读，再学写作，而且在当前，大多数开发人员的主要任务是修改已存在的代码，而不是开发代码。即使是主要从事开发的人员，也难免出于各种各样的目的去阅读他人的代码，其中一件重要的事情就是优化代码，提升代码质量。如果说作者的《代码阅读》一书将阅读代码

的技巧传授给了读者，那么《代码质量》则为后续的提高和优化提供了指导。《代码质量》一书重点讨论了代码的非功能特性，深入讲述代码如何满足重要的非功能性需求，如可靠性、安全性、可移植性和可维护性，以及时间效率和空间效率。本书从 Apache Web 应用服务器、BSD/UNIX 操作系统和 HSQLDB 数据库等开源项目中攫取数百个小例子，并以实例为基准点，辅以理论分析，从实用的角度讲述每个专业软件开发人员能立即运用的概念和技术。我们相信，每一名软件从业者都会从中有所收获、有所提高。作为译者，我们真诚地将该书推荐给国内的读者，希望借由本书，可以提高广大软件从业人员的编码质量，打开中国软件人才培养的新思路，若能如此，我心足矣。

怀揣着这样的心情，我们始终秉持着一丝不苟的态度，力求把经典之作原汁原味地带给中国读者。而一本专业技术领域的译作得以成功问世，其中之波折也是在所难免。幸得多位业内专家不吝指导，使得我们的工作倍感鼓舞。在本书翻译之初，原书作者 Diomidis Spinellis 教授即给我们提出了许多宝贵的建议，他的指导给予我们极大的帮助。此外，西北工业大学博士生导师、计算机学院副院长李战怀教授审阅了译稿，并欣然为本书作序推荐，感激之情，溢于言表。参与本书翻译校对工作的还有杨宁、丁玮、张曼、王团团、胡宗正、王延青，对于他们严肃认真的工作态度，谨表示由衷敬佩。

最高品质的图书始终是作译者永恒的追求。但有一千个读者，就有一千个哈姆雷特。因此，我们也真诚地希望本书的读者能够把阅读中的所想所得与我们分享，能够把书中的纰漏毫无保留地向我们指出，从而使得本书能够日臻完善，以利来者。欢迎访问笔者的技术博客 http://baimafujinji.blog.51cto.com，或发邮件至 heisbaixingzhi@163.com。

左飞

2011 年秋

原书序言

我们是程序员。我们的工作（在许多情况下，还有我们的兴趣）是通过编写代码的方式来促使某些事情成为现实。我们不会用大量的图表、详尽的项目进度表，或者四英尺高的一堆设计文档来满足用户的需求。所有这些都是意愿，它们表达出了我们真正希望实现的东西。我们通过编写代码的方式来交付用户的需求：代码才是实实在在的。

这也正是我们所得到的教导，它看似很合理。我们的工作是编写代码，所以我们需要学习如何去写代码。大学里的课程教我们编程。训练课程告诉我们如何通过编码的方式来使用新的库和应用程序接口。这是这个行业中最大的悲剧之一。

因为学习编写伟大代码的方式是阅读代码，阅读大量的代码。这些代码可能是高品质的，也可能是低品质的。可能是汇编语言代码，抑或是 Haskell[1]代码。可以是千里之外的陌生人所编写的代码，也可以是我们自己上周编写的代码。因为，如果不这样，我们势必会不断重复别人已经完成的工作，重复过往已经发生过的成功和失误。

我很好奇能有多少伟大的小说家从未读过其他人的著作，能有多少伟大的画家从来没有研究过他人的画作，能有多少技术娴熟的外科医生从未观摩过其他同事如何动手术，又能有多少波音 767 的机长不是首先在副驾驶的位置上观看别人如何操控飞机的。

1 Haskell 是一种纯函数式编程语言，名字源于美国数学家 Haskell Brooks Curry，他在数学逻辑方面的工作使得函数式编程语言有了广泛的基础。

然而，我们却期望程序员能够做到这些。“本周的任务是编写……”。我们告诉开发者语法和结构，然后便希望他们能够编写出相当于伟大小说的软件。

具有讽刺意味的是，对于阅读代码而言，从来没有比现在更好的时间了。多亏了开源社区所作出的巨大贡献，高达数吉字节的源代码就在网络上，只待我们去阅读。选定任何一种语言，你都可以找到源代码。选择任何一个问题领域，也都存在着大量的源代码。选择任何一个级别，从微码到高级别的商业功能，你都能够找到大量的源代码。

阅读代码是件妙趣横生的事。我喜欢阅读他人的代码。我阅读代码是为了学习技巧，分析陷阱。有时，我会遇到尽管很小，但却十分珍贵的至宝。我仍然记得，当我在 PDP-11 汇编程序里无意中找到一个二进制转八进制实例程序时所收获的喜悦，那段程序没有使用循环计数，在一个紧凑的循环中输出了一个 6 位八进制数。

有时我阅读代码是为了寻找一段故事，就像在长途飞行之前，你会在机场选择一本书一样。我期望从巧妙的构思和意外的匀称中收获愉悦。Jame Clark 的 gpic 程序（是其 GNU groff 包的一部分）就是此类代码中的一个绝佳例子。他用简明优雅的结构实现了明显极为复杂的功能（一种说明性的、设备无关的绘图语言）。对此，我深受鼓舞，并想着如何同样整洁地构造自己的代码。

有时，我还会批判地阅读代码。此时，阅读速度会放慢。阅读时，我会不断地向自己提问，比如：“为什么要这样写？”或者“何种背景使得作者做出了这种选择？”我这样做常常是因为，我要复查代码来解决存在的问题。我寻找能够给予我指示的模式与线索。如果我看到，作者在代码中的某处没有对共享的数据结构进行锁定，我就会怀疑同样的问题是否在别处也存在，接下来，就会想是否就是这个错误引起了我正在处理的问题。我还使用我发现的不一致性，进一步验证我的理解；我常常发现，那些本来认为可能存在问题的地方，经过进一步仔细分析后，变成了完美的好代码。从中，我也学到了一些东西。

实际上，代码阅读是消除程序中存在问题的最有效方式之一。Robert Glass，本书的审稿人之一，说过：“通过正确地使用（代码）审查，软件产品中 90%以上的错误能够在测试之前得以消除。”就在这篇文章中，他引用的调查研究表明，“将注意力放到代码上的检查人员比将注意力放到过程上的要多发现 90%的错误。”有趣的是，在阅读本书引用的代码片段时，我就碰到了几个 bug 和令人疑惑的编码实践。存在这些问题的代