HZ BOOKS 华章IT

EFFECTIVE 系列丛书

Pearson

Java之父James Gosling鼎力推荐、Jolt获奖作品全新升级，针对Java 7、8、9全面更新，Java程序员必备参考书

包含大量完整的示例代码和透彻的技术分析，通过90条经验法则，探索新的设计模式和语言习惯用法，帮助读者更加有效地使用Java编程语言及其基本类库

Effective Java

Third Edition

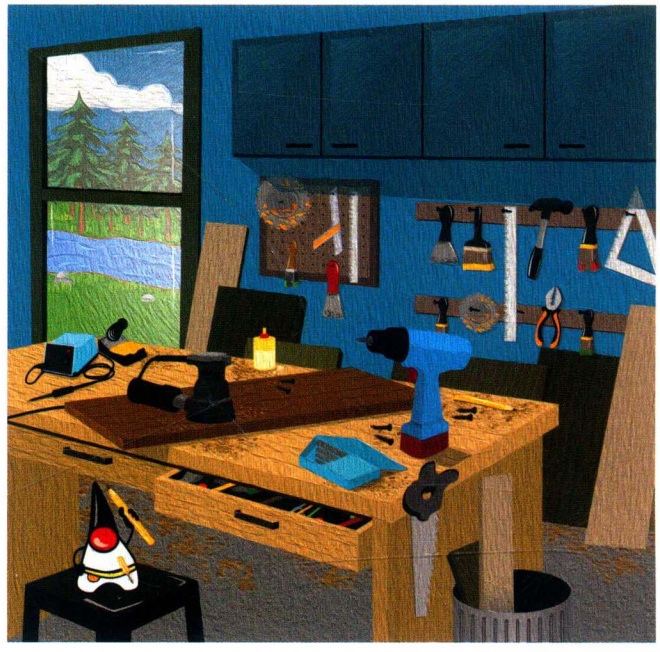
Effective Java

中文版

（原书第3版）

[美] 约书亚·布洛克（Joshua Bloch） 著

俞黎敏 译



机械工业出版社
China Machine Press

EFFECTIVE
系列丛书

Effective Java
Third Edition

Effective Java

中文版

（原书第3版）

[美] 约书亚·布洛克（Joshua Bloch） 著
俞黎敏 译

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Effective Java 中文版（原书第 3 版）/（美）约书亚·布洛克（Joshua Bloch）著；俞黎敏译 . —北京：机械工业出版社，2018.10
（Effective 系列丛书）
书名原文：Effective Java, Third Edition

ISBN 978-7-111-61272-8

I. E… II. ①约… ②俞… III. JAVA 语言－程序设计 IV. TP312.8

中国版本图书馆 CIP 数据核字（2018）第 247876 号

本书版权登记号：图字 01-2018-2524

Effective Java 中文版（原书第 3 版）

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：陈佳媛　　责任校对：殷 虹
印　刷：北京市兆成印刷有限责任公司　　版　次：2019 年 1 月第 1 版第 1 次印刷
开　本：186mm×240mm 1/16　　印　张：19.5
书　号：ISBN 978-7-111-61272-8　　定　价：119.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426 88361066　　投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259　　读者信箱：hzit@hzbook.com

华章IT
HZBOOKS | Information Technology

Forword 推 荐 序

如果有一个同事这样对你说："我的配偶今天晚上在家里制造了一顿不同寻常的晚餐，你愿意来参加吗？"（Spouse of me this night today manufactures the unusual meal in a home. You will join?）这时候你脑子里可能会想到三件事情：第一，满脑子的疑惑；第二，英语肯定不是这位同事的母语；第三，同事是在邀请你参加他的家庭晚宴。

如果你曾经学习过第二种语言，并且尝试过在课堂之外使用这种语言，就该知道有三件事情是必须掌握的：这门语言的结构是怎么样的（语法），如何命名你想谈论的事物（词汇），以及如何以惯用和高效的方式来表达日常事物（用法）。在课堂上大多只涉及前面两点，当你使出浑身解数想让对方明白你的意思时，却常常发现母语人士或当地人对你的表述忍俊不禁。

程序设计语言也是如此。你需要理解语言的核心：它是面向算法的，还是面向函数的或者是面向对象的？你需要知道词汇表：标准类库提供了哪些数据结构、操作和功能？你还需要熟悉如何用习惯和高效的方式来构建代码。关于程序设计语言的书籍通常只涉及前两点，或者只是蜻蜓点水般地介绍一下用法。也许是因为前两点比较容易编写。语法和词汇是语言本身固有的特性，用法则反映了使用这门语言的群体特征。

例如，Java 程序设计语言是一门支持单继承的面向对象程序设计语言，在每个方法的内部，它也支持命令式的（面向语句的）编码风格。Java 类库提供了对图形显示、网络、分布式计算和安全性的支持。但是，如何把这门语言以最佳的方式运用到实践中呢？

还有一点：程序与口语中的句子以及大多数书籍和杂志都不同，它会随着时间的推移而发生变化。仅仅编写出能够有效地工作并且能够被别人理解的代码往往是不够的，我们还必须把代码组织成易于修改的形式。针对某个任务 T 可能会有 10 种不同的编码方法，而在这 10 种方法中，可能有 7 种方法是笨拙、低效或者难以理解的。而在剩下的 3 种编码方法中，哪一种会是最接近任务 T 的下一年度发行版本的代码呢？

目前有大量的书籍可以供你学习 Java 程序设计语言的语法，包括《The Java Programming Language》(作者是 Arnold、Gosling 和 Holmes)，以及《The Java Language Specification》(作者是 Gosling、Joy 和 Bracha)。同样，介绍 Java 程序设计语言相关的类库和 API 的书籍也不少。

本书将解决你的第三种需求：习惯和高效的用法。作者 Joshua Bloch 在 Sun 公司多年来一直从事 Java 编程语言的扩展、实现和使用的工作；他还大量地阅读了其他人的代码，包括我的代码。他在本书中提出了许多很好的建议，系统地把这些建议组织起来，旨在告诉读者如何更好地构建代码，以便它们能够更好地工作，也便于其他人能够理解这些代码，将来对代码进行修改和改善的时候不至于那么头疼。甚至，你的程序也会因此而变得更加令人愉悦、更加优美和雅致。

Guy L. Steele Jr.

马萨诸塞州，伯灵顿

2001 年 4 月

The Translator's Words 译 者 序

Java 从 1997 年诞生到日趋完善，经过了 20 多年不断的发展壮大，已经拥有了近千万开发人员。如何编写出更清晰、更正确、更健壮且更易于重用的代码，是大家所追求的目标。本书是经典 Jolt 获奖作品《 Effective Java 》的第 3 版，对上一版内容进行了彻底的更新，涵盖了自 2001 年第 1 版之后所引入的 Java SE 5 和 Java SE 6 的新特性，以及 2008 年第 2 版之后所引入的 Java SE 7 和 Java SE 8 以及 Java SE 9 的新特性。作者探索了新的设计模式和语言习惯用法，介绍了如何充分利用从泛型到枚举、从注解到自动装箱的各种特性，帮助读者更加有效地使用 Java 编程语言及其基本类库：`java.lang`、`java.util` 和 `java.io`，以及子包，如 `java.util.concurrent` 和 `java.util.function` 等。本书的作者 Joshua Bloch 曾经是 Sun 公司的杰出工程师和 Google 公司的首席 Java 架构师，带领团队设计和实现过无数的 Java 平台特性，包括 JDK 5.0 语言增强版和获奖的 Java Collections Framework。在本书中，他为我们带来了 90 条程序员必备的经验法则：针对你每天都会遇到的编程问题提出了最有效、最实用的解决方案。

书中的每一章都包含几个“条目”，以简洁的形式呈现，自成独立的短文，它们提出了具体的建议、对于 Java 平台精妙之处的独到见解，并提供优秀的代码范例。每个条目的综合描述和解释都阐明了应该怎么做、不应该怎么做，以及为什么。通过阅读贯穿全书的透彻的技术剖析与完整的示例代码，认真理解并加以实践，必定会从中受益匪浅。书中介绍的示例代码清晰易懂，也可以作为日常工作的参考指南。

读者对象

本书不是针对初学者的，读者至少需要熟悉 Java 程序设计语言。如果你连 `equals()`、`toString()`、`hashCode()` 都还不了解的话，建议先去看些优秀的 Java 入门书籍，之后再来阅读本书。如果你在 Java 开发方面已经有一定的经验，想更加深入地了解 Java 编程语

言，成为一名更优秀、更高效的 Java 开发人员，那么，建议你用心研读本书。

内容形式

本书分为 12 章共 90 个条目，涵盖了 Java 5.0 / 6.0 / 7.0 / 8.0 / 9.0 的种种技术要点。与第 2 版相比，本书删除了“C 语言结构的替代”一章，增加了 Java 7 及之后所引入的新特性：Lambda 表达式、Stream、Optional 类、接口中的默认方法、`try-with-resources`、`@SafeVarargs` 注解、Module 模块化 。数量上从 78 个条目发展到了 90 个，不仅增加了 12 个条目，并对原来的所有资料都进行了全面的修改，删去了一些已经过时的条目。但是，各章之间并没有严格的前后顺序关系，你可以随意选择感兴趣的章节进行阅读。当然，如果你想马上知道第 3 版究竟有哪些变化，可以参阅附录。

本书重点讲述了 Java 5 所引入的全新的泛型、枚举、注解、自动装箱、`for-each` 循环、可变参数、并发机制，还包括对象、类、类库、方法和序列化这些经典主题的全新技术与最佳实践，以及如何避免 Java 编程语言中常被误解的细微之处：陷阱和缺陷，并重点关注了 Java 语言本身和最基本的类库（`java.lang`、`java.util`）和一些扩展（`java.util.concurrent` 和 `java.io` 等）。

主要章节简介

第 1 章为引言。

第 2 章阐述何时以及如何创建对象，何时以及如何避免创建对象，如何确保它们能够适时地销毁，以及如何管理对象销毁之前必须进行的各种清除动作。

第 3 章阐述对于所有对象都通用的方法，你会从中获知对 `equals`、`hashCode`、`toString`、`clone`、`finalize` 以及 `Comparable.compareTo` 方法相当深入的分析，从而避免今后在这些问题上再次犯错。

第 4 章阐述作为 Java 程序设计语言的核心以及 Java 语言的基本抽象单元（类和接口）在使用上的一些指导原则，帮助你更好地利用这些元素，设计出更加有用、健壮和灵活的类与接口。

第 5 章和第 6 章中分别阐述在 Java 1.5 发行版本中新增加的泛型（Generic）以及枚举（Enum）和注解（Annotation）的最佳实践，教你如何最大限度地享有这些优势，并使整个过程尽可能地简单化。

第 7 章专门讨论在 Java 8 中新增的函数接口（Functional Interface）、Lambda 表达式和方法引用（Method Reference），使创建函数对象（Function Object）变得更加容易。接着探讨为

处理数据元素的序列提供了类库级别支持的 Stream API，以及如何最佳地利用这些机制。

第 8 章讨论方法设计的几个方面：如何处理参数和返回值，如何设计方法签名，如何为方法编写文档，从而使方法设计在可用性、健壮性和灵活性上有进一步的提升。

第 9 章主要讨论 Java 语言的具体细节，讨论了局部变量的处理、控制结构、类库的使用、各种数据类型的用法，以及两种不是由语言本身提供的机制（Reflection 和 Native Method，反射机制和本地方法）的用法，并讨论了优化和命名惯例。

第 10 章阐述如何充分发挥异常的优点来提高程序的可读性、可靠性和可维护性，以及减少异常使用不当所带来的负面影响，并提供了一些关于有效使用异常的指导原则。

第 11 章阐述如何帮助你编写出清晰、正确、文档组织良好的并发程序，比如如何避免过度同步，优先采用 Executor Framework、并发集合（Concurrent Collection）、同步器（Synchronizer），以及是否需要依赖于线程调度器等。

第 12 章阐述序列化方面的技术，并且有一项值得特别提及的特性，就是序列化代理（Serialization Proxy）模式，它可以帮助你避免对象序列化的许多缺陷。

举个例子，就序列化技术来讲，HTTP 会话状态为什么可以缓存？ RMI 的异常为什么可以从服务器端传递到客户端？ GUI 组件为什么可以被发送、保存和恢复呢？是因为它们实现了 Serializable 接口吗？如果超类没有提供可访问的无参构造器，它的子类可以被序列化吗？当一个实例采用默认的序列化形式，并且给某些域标记为 transient，那么当实例反序列化回来后，这些标记为 transient 域的值各是些什么呢？这些问题如果你现在不能马上回答，或者不能确定，也没有关系，请仔细阅读本书，你将会对它们有更深入与透彻的理解。

技术范围

虽然本书所讨论的是更深层次的 Java 开发技术，讲述的内容深入，涉及面又相当广泛，但是它并没有涉及图形用户界面编程、企业级 API 以及移动设备方面的技术，不过在一些条目中会不时地讨论到其他相关的类库。

这是一本分享经验与指引你少走弯路的经典著作，针对如何编写高效、设计优良的程序提出了最实用、最权威的指导方针，是 Java 开发人员案头上的一本不可或缺的参考书。

本书由我组织进行翻译，并负责本书所有章节的全面审校。参与翻译和审校的还有：杨春花、荣浩、邱庆举、万国辉、陆志平、姜法有、王琳、林仪明、凌家亮、李勇、刘传飞、王建旭、程旭文、罗兴、翟育明、杨征和陈建都。

虽然我们在翻译过程中竭力追求信、达、雅，但限于自身水平，也许仍有不足，还望各位读者不吝指正。关于本书的翻译和翻译时采用的术语以及相关的技术讨论大家可以访问我

的博客 http://YuLimin.ItEye.com，也可以发邮件到 YuLimin @ 163.com 与我交流。

在这里，我要感谢在翻译过程中一起讨论并帮助我的朋友们，他们是：满江红开放技术研究组织创始人曹晓钢，Spring 中文站创始人杨戈（Yanger），SpringSide 创始人肖桦（江南白衣）和来自宝岛台湾的李日贵（jini）、林康司（koji）、林信良（caterpillar），在此再次深表感谢。

最后，感谢华章公司的两位编辑陈佳媛与关敏，她们耐心、细致地审校了全书，使本书得到了极大的改善。赞！

快乐分享，实践出真知，最后，祝大家能够像我一样在阅读中享受本书带来的乐趣！

Read a bit and take it out, then come back read some more.

俞黎敏

2018 年 10 月 24 日于港珠澳大桥开通之时

Preface 前　言

第 3 版前言

1997 年，Java 才面世不久，James Gosling（Java 之父）称之为“超级简单的蓝领语言”[Gosling97]。几乎与此同时，Bjarne Stroustrup（C++ 之父）则将 C++ 称为“多范式语言”（multi-paradigm language），因为“它与那些只支持单一编程方法的程序语言有着天壤之别”[Stroustrup95]。Stroustrup 曾发出过这样的警告：

> 正如大多数刚面世的语言一样，Java 的相对简单性，很可能一部分是出于错觉，一部分是因为其尚不完整而导致的结果。随着时间的推移，Java 在规模和复杂度方面都会显著增长。到那时，其规模可能呈双倍甚至三倍增长，并产生大量依赖于实现的扩展或者类库 [Stroustrup]。

现在，二十年过去了，坦白地说，Gosling 和 Stroustrup 说的都没有错。Java 现在果然是既庞大又复杂，许多东西都带有多个抽象，从并发执行，到迭代，再到日期和时间的表示法。

随着 Java 平台的发展，我的热情有所降温，但我依然钟爱 Java。考虑到 Java 日益增加的规模和复杂度，对于最前沿的最佳实践指导的需求成了重中之重。在本书中，我将不遗余力地为读者提供这样的指导。希望这一版能够在坚持前两个版本的精神的前提下，继续满足读者的最新需求。

简单即美，但要做到大道至简却实属不易。

Joshua Bloch

San Jose, California

2017 年 11 月

附：近年来，我在业界的最佳实践方面花费了大量的精力。自 20 世纪 50 年代诞生这个行业以来，我们已经自由地重新实现了彼此的 API。这个实践对于计算机技术的快速成功至关重要。我始终积极地致力于维护这种自由［ CompSci17 ］，并且鼓励你们也加入到这个行列中来。我们的专业要想持续健康地发展，确保重新实现各自 API 的权利显得尤为重要。

第 2 版前言

自从我于 2001 年写了本书的第 1 版之后，Java 平台又发生了很多变化，是该出第 2 版的时候了。Java 5 中最为重要的变化是增加了泛型、枚举类型、注解、自动装箱和 `for-each` 循环。其次是增加了新的并发类库：`java.util.concurrent`。我和 Gilad Bracha 一起，有幸带领团队设计了最新的语言特性。我还有幸参加了设计和开发并发类库的团队，这个团队由 Doug Lea 领导。

Java 平台中另一个大的变化在于广泛采用了现代的 IDE（Integrated Development Environment），例如 Eclipse、IntelliJ IDEA 和 NetBeans，以及静态分析工具的 IDE，如 FindBugs。虽然我还未参与这部分工作，但已经从中受益匪浅，并且很清楚它们对 Java 开发体验所带来的影响。

2004 年，我离开 Sun 公司到了 Google 公司工作，但在过去的 4 年中，我仍然继续参与 Java 平台的开发，在 Google 公司和 JCP（Java Community Process）的大力帮助下，继续并发和集合 API 的开发。我还有幸利用 Java 平台去开发供 Google 内部使用的类库。现在我了解了作为一名用户的感受。

我在 2001 年编写第 1 版的时候，主要目的是与读者分享我的经验，便于让大家避免我所走过的弯路，使大家更容易成功。新版仍然大量采用来自 Java 平台类库的真实范例。

第 1 版所带来的反应远远超出了我最大的预期。我在收集所有新的资料以使本书保持最新时，尽可能地保持了资料的真实。毫无疑问，本书的篇幅肯定会增加，从 57 个条目发展到了 78 个。我不仅增加了 23 个条目，并且修改了原来的所有资料，并删去了一些已经过时的条目。在附录中，你可以看到本书中的内容与第 1 版的内容的对照情况。

在第 1 版的前言中我说过：Java 程序设计语言和它的类库非常有益于代码质量与效率的提高，并且使得用 Java 进行编码成为一种乐趣。Java 5 和 Java 6 发行版中的变化是好事，这也使 Java 平台日趋完善。现在这个平台比 2001 年的要大得多，也复杂得多，但是一旦掌握了使用新特性的模式和习惯用法，它们就会使你的程序变得更完美，使你的工作变得更轻

松。我希望第 2 版能够体现出我对 Java 平台持续的热情，并将这种热情传递给你，帮助你更加高效和愉快地使用 Java 平台及其新的特性。

Joshua Bloch
San Jose, California
2008 年 4 月

第 1 版前言

1996 年，我打点行囊，西行来到了当时的 JavaSoft，因为我很清楚那里将会出现奇迹。在这 5 年间，我是 Java 平台库的架构师。我设计、实现和维护过许多类库，同时也担任其他一些库的技术顾问。随着 Java 平台的成熟和壮大，主持这些类库的设计工作是一个人一生中难得的机会。毫不夸张地说，我有幸与一些当代最杰出的软件工程师一起工作。在这个过程中，我学到了许多关于 Java 程序设计语言的知识——它能够做什么，不能够做什么，以及如何最有效地使用这门语言及其类库。

本书是我的一次尝试，希望与你分享我的经验，你可以因此而吸取我的经验，避免重复我的失败。本书中我借用了 Scott Meyers 的《 Effective C++ 》一书的格式，该书中包含 50 个条目，每个条目给出一条用于改进程序和设计的规则。我觉得这种格式非常有效，希望你也有这样的感觉。

在许多例子中，我冒昧地使用了 Java 平台库中的真实例子来说明相应的条目。在介绍那些做得不是很完美的工作时，我尽量使用我自己编写的代码，但是偶尔我也会使用其他同事的代码。尽管我尽力做得更好一点，但是如果我真的冒犯了他人，我先在这里致以最诚挚的歉意。引用反面例子是出于协作的精神，而不是要羞辱例子中的做法，我希望大家都能够从我们过去的错误经历中得到启发。

尽管本书并不只是针对可重用组件开发人员的，但是过去 20 多年来我编写此类组件的经历一定会影响这本书。我很自然地会按照可导出 API（ Application Programming Interface）的方式来思考问题，而且我建议你也这样做。即使你并没有开发可重用的组件，这样的思考方法也将有助于你提升软件的质量。进一步来说，毫无意识地编写可重用组件的情形并不少见：你编写了一些很有用的代码，然后在同伴之间共享，不久之后你就有了很多用户。这时候，你就不能随心所欲地改变 API 了，并且如果你刚开始编写软件的时候在设计 API 上付出了较多的努力，那么这时你就会非常庆幸了。

我把焦点放在 API 的设计上，这对于那些热衷于新兴的轻量级软件开发方法学（比如

Extreme Programming，即“极限编程”，简称 XP）的读者来说，也许会显得有点不太自然。这些方法学强调编写最简单的、能够工作的程序。如果你正在使用此类的某种程序设计方法，那么你会发现，把焦点放在 API 设计上对于“重构”（refactoring）过程是多么有益。重构的基本目标是改进系统结构，以及避免代码重复。如果系统的组件没有设计良好的 API，要达到这样的目标则是不可能的。

没有一门语言是完美的，但是有些语言非常优秀。我认为 Java 程序设计语言及其类库非常有益于提高代码质量和工作效率，并使得编码工作成为一种乐趣。我希望本书能够抓住我的热情并传递给你，帮助你更有效地利用 Java 语言，使工作变得更加愉快。

Joshua Bloch

Cupertino, California

2001 年 4 月

Acknowledgements 致　谢

第 3 版致谢

我要感谢本书前两版的读者给予本书如此热情的好评，感谢他们将书中的理念铭记在心，感谢他们让我知道该书给他们及其工作带来了怎样积极的影响。感谢许多教授在教学中采用了本书，感谢许多开发团队应用了本书。

我要感谢 Addison-Wesley 和 Pearson 的整个团队，感谢他们的诚恳、专业、耐心，以及极端压力之下所体现出来的从容。编辑 Greg Doench 自始至终保持镇定自若：他是一名优秀的编辑，同时也是一位完美的绅士。我担心在这个项目结束时，他会为此增添不少银丝，为此我深表歉意。产品经理 Julie Nahil 和项目编辑 Dana Wilson 具备了应该具备的一切：勤奋、敏捷、训练有素，且待人和气。文字编辑 Kim Wimpsett 一丝不苟，富有鉴赏能力。

我有幸再一次得到了所能想到的最佳审校团队的支持，真诚地感谢他们中的每一位。核心团队负责审校每一个章节，他们包括：Cindy Bloch、Brian Kernighan、Kevin Bourrillion、Joe Bowbeer、William Chargin、Joe Darcy、Brian Goetz、Tim Halloran、Stuart Marks、Tim Peierls 以及 Yoshiki Shibata。其他审校人员包括：Marcus Biel、Dan Bloch、Beth Bottos、Martin Buchholz、Michael Diamond、Charlie Garrod、Tom Hawtin、Doug Lea、Aleksey Shipilëv、Lou Wasserman 以及 Peter Weinberger。这些审校人员再次提出了大量的建议，使本书得到了极大的改善，也让我避免了诸多尴尬局面。

我要特别感谢 William Chargin、Doug Lea 和 Tim Peierls，他们成了书中许多理念的倡导者，并为本书毫不吝惜地奉献了他们的时间和学识。

最后，我要感谢妻子 Cindy Bloch，她鼓励我写作，阅读了初稿中的每个条目，为我编写索引，帮我打理在完成全书工作时难免发生的一切事务，在我写作的时候一直对我十分宽容。

第 2 版致谢

我要感谢本书第 1 版的读者给予本书如此热情的好评，感谢他们将书中的理念铭记在心，感谢他们让我知道本书给他们以及他们的工作带来了怎样积极的影响。感谢许多教授在教学中采用了本书，感谢许多开发团队应用了本书。

我要感谢 Addison-Wesley 的整个团队，感谢他们的诚恳、专业、耐心，以及压力之下所体现出来的从容。编辑 Greg Doench 自始至终保持镇定自若：他是一名优秀的编辑，同时也是一位完美的绅士。产品经理 Julie Nahil 具备了产品经理应该具备的一切：勤奋、敏捷、训练有素，且待人和气。文字编辑 Barbara Wood 一丝不苟，富有鉴赏能力。

我有幸再一次得到了所能想到的最佳审校团队的支持，我真诚地感谢他们中的每一位。核心团队负责审校每一个章节，他们包括：Lexi Baugher、Cindy Bloch、Beth Bottos、Joe Bowbeer、Brian Goetz、Tim Halloran、Brian Kernighan、Rob Konigsberg、Tim Peierls、Bill Pugh、Yoshiki Shibata、Peter Stout、Peter Weinberger 以及 Frank Yellin。其他审校人员包括：Pablo Bellver、Dan Bloch、Dan Bornstein、Kevin Bourrillion、Martin Buchholz、Joe Darcy、Neal Gafter、Laurence Gonsalves、Aaron Greenhouse、Barry Hayes、Peter Jones、Angelika Langer、Doug Lea、Bob Lee、Jeremy Manson、Tom May、Mike McCloskey、Andriy Tereshchenko 以及 Paul Tyma。这些审校人员再次提出了大量的建议，使本书得到了极大的改善，也让我避免了诸多尴尬。剩下的任何错误都是我自己的责任。

我要特别感谢 Doug Lea 和 Tim Peierls，他们成了书中许多理念的倡导者。Doug 和 Tim 为本书毫不吝惜地奉献了他们的时间和学识。

我要感谢我在 Google 公司的经理 Prabha Krishna，感谢她持续不断的支持和鼓励。最后，我要感谢我的妻子 Cindy Bloch，她鼓励我写作，阅读了初稿中的每个条目，用 Framemaker 帮我排版，为我编写索引，在我写作的时候一直对我十分宽容。

第 1 版致谢

我要感谢 Patrick Chan 建议我写这本书，并将这样的想法传达给此系列图书的主编 Lisa Friendly 和此系列图书的技术编辑 Tim Lindholm；以及 Addison-Wesley 出版社的执行编辑 Mike Hendrickson。感谢 Lisa、Tim 和 Mike 对我的鼓励，以及认定我终有一天可以完成此书而保持的耐心和坚定的信念。

感谢 James Gosling 及其原始团队给予我这么好的写作题材。也感谢众多追随 James 足迹的 Java 平台工程师。特别要感谢我在 Sun 公司的 Java Platform Tools 和 Libraries Group 的同事们，感谢他们的见解、鼓励和支持。这个团队由 Andrew Bennett、Joe Darcy、Neal Gafter、

Iris Garcia、Konstantin Kladko、Ian Little、Mike McCloskey 和 Mark Reinhold 组成。前一个团队的成员还包括 Zhenghua Li、Bill Maddox 和 Naveen Sanjeeva。

我要感谢我的经理 Andrew Bennett 和我的领导 Larry Abrahams 对这个写作计划的热情支持。我还要感谢 Java Software 工程副总裁 Rich Green，他提供的环境让工程师能够以创造性的方式去自由思考并发表成果。

我拥有一个你所能想象的最佳审校团队，我要把我最真诚的感谢献给他们：Andrew Bennett、Cindy Bloch、Dan Bloch、Beth Bottos、Joe Bowbeer、Gilad Bracha、Mary Campione、Joe Darcy、David Eckhardt、Joe Fialli、Lisa Friendly、James Gosling、Peter Haggar、David Holmes、Brian Kernighan、Konstantin Kladko、Doug Lea、Zhenghua Li、Tim Lindholm、Mike McCloskey、Tim Peierls、Mark Reinhold、Ken Russell、Bill Shannon、Peter Stout 和 Phil Wadler，以及两位未署名的审校者。他们提出了很多建议，使本书获得了极大的改善，也让我避免了诸多尴尬局面。剩下的任何错误都是我自己的责任。

许多同事，包括 Sun 公司上上下下的员工，都参与了本书的技术讨论，从而提升了本书的质量。其中 Ben Gomes、Steffen Grarup、Peter Kessler、Richard Roda、John Rose 和 David Stoutamire 贡献了极有用的深刻见解。我还要特别感谢 Doug Lea，他在本书许多构想中与我产生共鸣。他也无私地将其时间和知识都毫无保留地分享给了我。

感谢 Julie Dinicola、Jacqui Doucette、Mike Hendrickson、Heather Olszyk、Tracy Russ，以及 Addison-Wesley 出版社的整个支持团队，感谢他们的支持和专业。即使处于一种几乎不可能的紧张进度中，他们还是始终保持友善和通融。

感谢 Guy Steele 为我写序。我很荣幸能有他参与这个项目。

最后，要感谢我的妻子 Cindy Bloch，感谢她鼓励并偶尔催促我撰写本书，感谢她阅读本书刚出炉的每一个条目，感谢她用 Framemaker 帮助我排版，还编写本书的索引，并在我写作期间始终对我十分宽容。