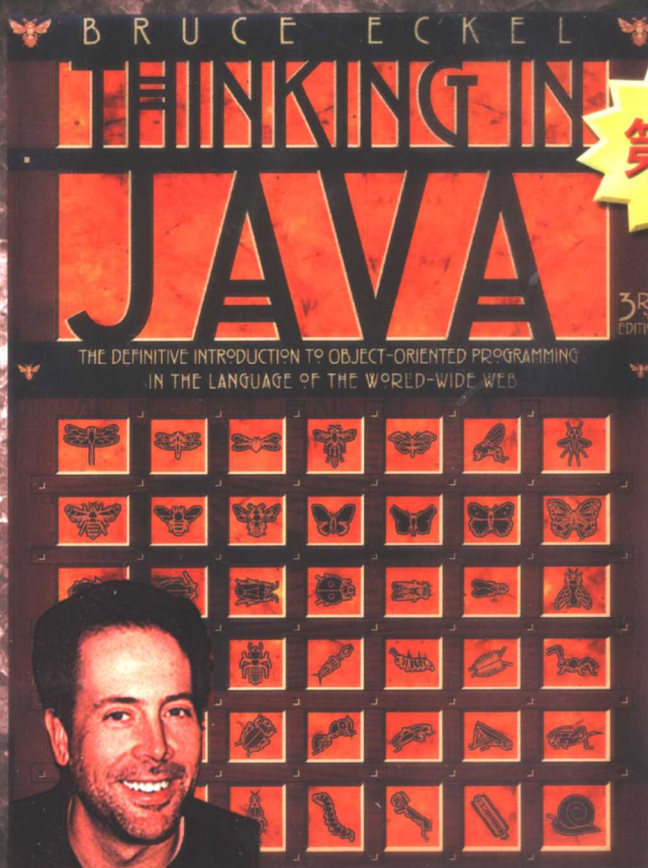


PEARSON  
Prentice  
Hall

计 算 机 科 学 从 书


# Java 编程思想

(美) Bruce Eckel 著 陈昊鹏 饶若楠 等译



第3版

Thinking in Java  
Third Edition

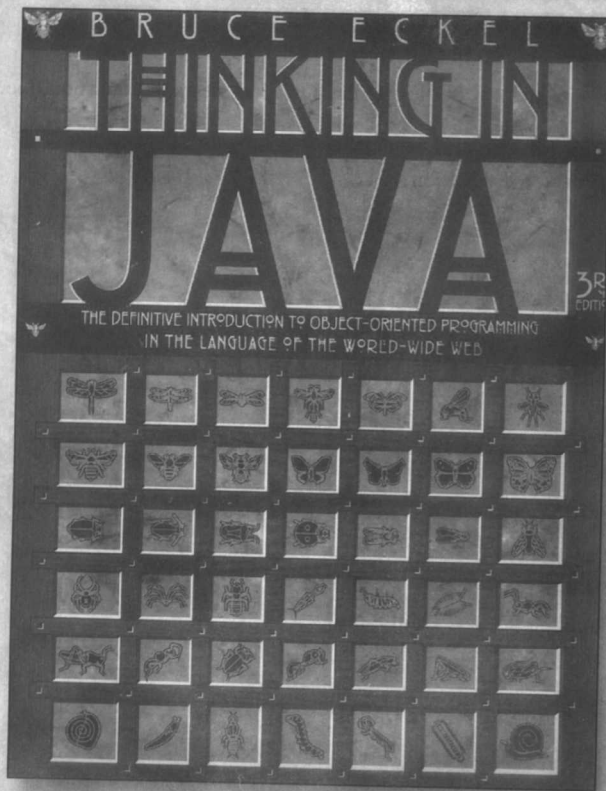
 机械工业出版社  
China Machine Press

计 算 机 科 学 丛 书

第3版

# Java 编程思想

(美) Bruce Eckel 著 陈昊鹏 饶若楠 等译



**Thinking in Java**  
Third Edition

 机械工业出版社  
China Machine Press

从本书获得的各项大奖以及来自世界各地的读者评论中,不难看出这是一本经典之作。本书作者有多年的教学经验,对C、C++、Java语言都有独到、深入的理解。因此他非常了解如何教授Java语言这门课程,也非常明白语言教学中的难点及人们的困惑。作者以通俗易懂及小而直接的示例解释了一个个晦涩抽象的概念,精心选取“对读者理解Java语言来说最为重要”的部分编写成书。同时又在随书光盘中提供了大量参考材料——这也是本书绝对物超所值的地方。

随书光盘没有包含本书的源代码(可以从支持网站www.MindView.net免费下载),而是提供了大量作者讲授的讨论课内容及本书的前2版内容。

本书内容丰富——从Java的基础语法到最高级特性,适合各层次的Java程序员阅读,同时也是高等院校讲授面向对象程序设计语言及Java语言的绝佳教材。

Simplified Chinese edition copyright © 2005 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Thinking in Java, Third Edition* by Bruce Eckel, Copyright © 2003.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall PTR.

本书封面贴有Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2003-1994

### 图书在版编目(CIP)数据

Java编程思想:第3版 / (美)埃克尔(Eckel, B.)著;陈昊鹏等译. -北京:机械工业出版社, 2005.5

(计算机科学丛书)

书名原文: *Thinking in Java, Third Edition*

ISBN 7-111-16220-X

I. J… II. ①埃… ②陈… III. Java语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(2005)第018953号

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:刘立卿

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2005年5月第3版第1次印刷

787mm × 1092mm 1/16 · 49.75印张

印数:0001-5 000册

定价:95.00元(附光盘)

凡购本书,如有缺页、倒页、脱页,由本社发行部调换  
本社购书热线:(010) 68326294

## 出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭开了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

电子邮件: [hzedu@hzbook.com](mailto:hzedu@hzbook.com)

联系电话: (010) 68995264

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037

# 专家指导委员会

(按姓氏笔画顺序)

尤晋元  
石教英  
张立昂  
邵维忠  
周立柱  
范明  
袁崇义  
谢希仁

王珊  
吕建  
李伟琴  
陆丽娜  
周克定  
郑国梁  
高传善  
裘宗燕

冯博琴  
孙玉芳  
李师贤  
陆鑫达  
周傲英  
施伯乐  
梅宏  
戴葵

史忠植  
吴世忠  
李建中  
陈向群  
孟小峰  
钟玉琢  
程旭

史美林  
吴时霖  
杨冬青  
周伯生  
岳丽华  
唐世渭  
程时端

## 读者评论

这本书比我见过的所有Java书都要好得多。循序渐进……非常完整，并搭配恰到好处的范例，睿智而不呆板的解说……这使本书的品质比别的书“超出了一个数量级”。与其他Java书相比，我发现本书考虑非常周全、前后一致、理性坦诚、文笔流畅、用词准确。恕我直言，这是一本学习Java的理想书籍。

Anatoly Vorobey, 以色列海法Technion大学

在我所见过的程序设计指南中（无论何种语言），这绝对是最好的一本。

Joakim Ziegler, FIX系统管理员

感谢您这本精彩的、令人愉快的Java书。

Dr. Gavin Pillay, 登记员, 南非爱德华八世医院

再次感谢您这本杰出的书。作为一名不用C语言的程序员，我曾经感到（学习Java）步履维艰，但是您的书让我一目了然。能够一开始就理解底层的概念和原理，而不是通过反复试验来自己建立概念模型，真是太棒了。我希望能在今后的将来参加您的讨论课。

Randall R. Hawley, 自动化工程师, Eli Lilly公司

我见过的计算机著作中，这是最好的一本。

Tom Holland

这是我读过的编程语言书中最棒的一本……有关Java的书中最棒的一本。

Ravindra Pai, Oracle 公司, SUNOS 产品线部门

我见过的最好的Java书！您做了一项了不起的工作。您的深度令人赞叹，出版的时候，我一定会购买一本。我从1996年10月就开始学习Java，其间也读过好几本这方面的书，但我觉得您这本才是“必读书”。最近几个月，我一直集中精力于一个完全用Java开发的产品。您的书帮我夯实了某些不牢固的知识点，并拓展了我的知识面。我甚至在面试签约者时引用书中的内容，作为参考的依据。通过问一些我从书中学到的知识，来判断他们对Java的理解程度（例如，数组与Vector的区别）。您的书真是伟大！

Steve Wilkinson, 资深专家, MCI 电信公司

伟大的书。迄今为止我见过的最佳Java书籍。

Jeff Sinclair, 软件工程师, Kestral 计算技术公司

感谢您的《Thinking in Java》。早就应该有人把仅仅介绍语言的教程编写成富有思想、分析透彻的入门指南，而不是局限于“某个公司”的语言。我阅读过许多这方面的书，但只有您和Patrick Winston的作品给我印象深刻。我已经向客户推荐这本书。再次感谢您。

Richard Brooks, Java 咨询顾问, 达拉斯Sun专业服务部门

Bruce, 您的书真是太棒了！您的讲解清晰明确。通过这本迷人的书，我获得了大量Java知

识。练习题也同样令人着迷，它们对巩固各章阐述的知识起到了很好的效果。我期待您的更多作品。对您的这本著作致以谢意。阅读了《Thinking in Java》之后，我的代码质量大有改善。为此我要感激您，我相信，维护我的代码的程序员同样也会感激您。

Yvonne Watkins, Discover 技术公司

其他书籍只涵盖Java的WHAT（探讨语法和相关程序库），或者只包含Java的HOW（实际的程序范例）。《Thinking in Java》则是我知道的书籍中唯一对Java的WHY做出讲解的一本。为什么要这样设计，为什么它会那样运作，为什么有时候会发生问题，为什么它在某些方面比C++好而某些方面不会。虽然它在教授程序语言的WHAT和HOW方面也很成功，但《Thinking in Java》更是爱钻研者的首选Java书籍。

Robert S. Stephenson

感谢您写了一本伟大的书。我越看越喜欢。我的学生也很喜欢。

Chuck Iverson

我要赞美您在《Thinking in Java》一书上的表现。正是有了您这样的人，才使得因特网充满前景，而我想感谢您的付出与努力。真是感激不尽。

Patrick Barrell, Network Officer Mamco, QAF Mfg. Inc.

我真的非常感激您的热情与您的作品。我下载了你的在线书籍的每一个修订版本，我正在深入钻研语言，并探索那些以前从来不敢碰的内容（对于C#、C++、Python和Ruby也有作用）。我至少还有15本Java书（为了工作，我还要掌握JavaScript和PHP语言），并且订阅了《Dr. Dobbs》、《JavaPro》、《JDJ》、《JavaWorld》等杂志。在深入钻研Java（包括Java企业版）之后，我对您的书更加尊敬。它的确是一本有思想的书。我订阅了您的邮件列表，并希望有一天，我所探讨和解决的问题能被您扩展到解题指导中（我将购买解题指导！）。同时，非常感激。

Joshua Long, www.starbuxman.com

市面上的Java书籍，大多比较适合初学者。它们大多数也就只具备基础内容，范例也大同小异。在我见过的富有思想性并讲解高级主题的书籍中，您的是最好的。快点出版吧！……鉴于《Thinking in Java》带给我的深刻印象，我也购买了《Thinking in C++》。

George Laframboise, LightWorx 技术咨询公司

关于您的《Thinking in C++》（我工作的时候，它总在书架上占据最显眼的位置），我曾经写信告诉过您我对它的喜爱。现在，我通过您的电子书仔细钻研Java，我还得说“我喜欢！”本书内容广博，讲解详细，阅读起来不像是无味的教科书。您的书中涵盖了Java开发工作中最重要、却很少被提及的概念——“原理”。

Sean Brady

我同时用Java和C++进行开发，您的这两本书是我的救星。如果我被某个问题难住了，我知道可以靠您的书来：a) 清楚地解释原因；b) 找到符合我所遇问题的具体例子。我还没找到另一本能令我如此反复热情推荐的作者（如果有人愿意听我推荐的话）。

Josh Asbury, A^3 软件咨询公司, 辛辛那提, OH

您的例子不仅清楚，而且容易理解。Java中的许多重要细节您都考虑到了，这些内容在编排较差的Java文档中很难找到。您假设程序员已经具有了基本知识，这就节约了读者的时间。

Kai Engert, 德国Innovative 软件公司



我是《Thinking in C++》一书的忠实书迷，我已经将它推荐给了我的同事们。当我读完您的Java书籍电子版时，我觉得，您总能保持高水准的写作水平。感谢您！

Peter R. Neuwald

写得非常好的Java书……我认为您在此书上取得了非常出色的成就。作为芝加哥地区Java兴趣小组的领导人，我已经多次在我们最近的聚会中赞扬您的这本书和您的网站。我想将《Thinking in Java》作为我们每月聚会讨论的主要内容。这样我们可以在聚会中对书中的章节进行复习和讨论。

Mark Ertes

顺便提一下，《Thinking in Java 2nd Edition》俄语版依旧畅销。阅读此书已经与学习Java成为同义词，真是太好了。

Ivan Porty (《Thinking In Java 2nd Edition》俄语版的译者及出版商)

对于您的辛勤工作，我由衷感激。您的书是佳作，我将这本书推荐给我们这儿的使用者和博士班学生。

Hugues Leroy // Irisa-Inria Rennes France,  
Head of Scientific Computing and Industrial Tranfert

虽然我只读了约40页的《Thinking in Java》，却已经发现本书是我所见过的讲述最为清晰、编排最为合理的程序设计书籍……作为一名作者，我可能会有些挑剔。我已经订购了《Thinking in C++》，迫不及待地想钻研一番。对于程序设计，我还算是新手，因此事事都得学习。这不过是一篇向您的绝佳作品致谢的简短书信。在痛苦地浏览大多数语言艰涩、内容散乱的计算机书籍（包括那些有着极佳口碑的书籍）后，我对计算机书籍的阅读热情一度消退。不过，现在我又重拾信心。

Glenn Becker, Educational Theatre Association

感谢您提供了这么一本精彩的书。当我遇到那些令人困惑的Java和C++问题时，这本书对我最终理解问题提供了极大帮助。阅读您的书令人如沐春风。

Felix Bizaoui, Twin Oaks Industries, Louisa, Va.

对于这部优秀的作品，我必须向您道贺。鉴于阅读《Thinking in C++》的经验，我决定读一读《Thinking in Java》，而事实证明它的确未让人失望。

Jaco van der Merwe, 南非DataFusion系统公司软件专家

本书无疑是我所见过的最佳的Java书籍之一。

E.F. Pritchard, 英国剑桥动画系统公司高级软件工程师

您的书使那些我曾经读过或草草翻过的Java书显得更加无用、该骂。

Brett g Porter, Art & Logic公司高级程序员

我阅读您这本书已经一两个星期了。与以前我曾读过的Java书籍比较，您的书似乎更能给我一个绝佳的开始。我已经把此书推荐给我的朋友，他们对此书也评价甚高。对于您写出的这本著作，请接受我的恭喜。

Rama Krishna Bhupathi, 加州圣何塞TCSI公司软件工程师

只是很想告诉您，您这本书是多么杰出的作品。我已将它作为公司内部Java工作的主要参考书。我发现目录的安排恰如其分，可以很快找到需要的章节。能够看到这么一本既不拿API炒冷饭，也不把程序员当傻瓜的书，真是太棒了。

Grant Sayer, 澳大利亚Ceedata 系统私人有限公司, Java组件组长

哇! 这是一本可读性强、极富深度的Java书籍。市面上已经有太多质量低劣的Java书籍。其中虽然也有少数不错的,但在看过您的大作之后,我认为它当然是最好的。

John Root, 伦敦社会安全局Web开发人员

我才刚开始阅读《Thinking in Java》。我想它一定相当不错,因为我很喜欢《Thinking in C++》(我以一名熟悉C++、同时希望提升自身能力的程序员身份来阅读这本书)。尽管我不太熟悉Java,但本书想必能令我满意。您是一位伟大的作家。

Kevin K. Lewis, ObjectSpace公司技术专家

我想这是一本了不起的书。我所有的Java知识都学自这本书。感谢您让大家可以从Internet上免费取得这本书。如果没有您的付出,我至今恐怕仍然对Java一无所知。本书最棒的一点,莫过于它同时也说明了Java不好的一面,而不像那些商业宣传资料。您的表现真是优秀。

Frederik Fix, 比利时

我始终热中读您的著作。几年以前,当我开始学习C++时,是《C++ Inside & Out》带领我进入C++的迷人世界。那本书帮助我得到了更好的机会。现在,为了更进一步钻研知识,我兴起了学习Java的念头,我无意中又碰见了《Thinking in Java》。毫无疑问,我认为自己不再需要其他书籍。它是那么的令人难以置信。阅读此书的过程,就像重新发掘自我一样。我学习Java至今只有一个月,现在对Java的体会日益加深,这一切都不得不由衷感谢您。

Anand Kumar S., 印度Computervision公司软件工程师

您的书作为综合性的导论,是如此出色。

Peter Robinson, 剑桥大学计算机实验室

在帮助我学习Java的书籍中,这一本显然是最好的。我只是想让您知道,我觉得自己能够读到这本书是多么幸运。谢谢!

Chuck Peterson, IVIS 国际公司Internet产品线产品组长

了不起的一本书。自从我开始学习Java,这已经是第三本了。目前我大概阅读了三分之二,并打算把它读完。我能够找到这本书,是因为这本书被用于Lucent 技术公司的某些内部课程,而且有个朋友告诉我这本书可以在网络上找到。很棒的作品。

Jerry Nowlin, Lucent 技术公司MTS部门

在我所读过的六本Java书籍中,您的《Thinking in Java》显然最好,也最清晰易懂。

Michael Van Waas博士, TMR Associates公司总裁

感谢您的《Thinking in Java》。您的作品真是精彩!更不必说它可以免费从网络下载了!作为一名学生,我觉得您的书籍是无价之宝(我也有一本《C++ Inside & Out》,它同样是一本伟大的C++书籍),因为您的书不仅教导我应该怎么做,也教导我这么做的原因所在,这一点对

C++或Java学习者建立起坚实基础非常重要。我有许多和我一样喜爱程序设计的朋友，我也对他们提起您的书。他们觉得真是太棒了！再次感谢您！顺道一提，我是印度尼西亚人，就住在“爪哇”（Java）。

Ray Frederick Djajadinata, 印度尼西亚雅加达Trisakti 大学学生

单是将作品免费放在网络上这种气度，就令我震惊不已。我想，我应该让您知道，对于您的工作，我是多么感激与尊敬。

Shane LeBouthillier, 加拿大 Alberta大学计算机工程系学生

我得告诉您，每个月我都在期待您的专栏。作为面向对象程序设计领域的新手，我要感谢您花在那些基础主题上的时间和思考。我已经下载了您的这本书，而且我一定会在本书出版的时候购买一本。感谢您对我的帮助。

Dan Cashmer, B. C. Ziegler公司

能够完成这么了不起的作品，恭喜您。开始，我偶然发现了《Thinking in Java》的PDF版本。甚至在我读完之前，我又跑到书店找到了《Thinking in C++》。我已经在计算机领域工作了八年多，做过顾问、软件工程师、教师/教练，最近则从事自由职业。所以我觉得自己也算是见多识广了（注意，不是“无所不知”，而只是“见多识广”）。不过，这些书使得我的女朋友称我为“呆子”。我并不反对，只不过我发现自己已经远远超过这个阶段。我发现自己如此喜爱这两本书，我以前接触过或购买过的其他计算机书籍，都无法与之相比。这两本书都有极佳的写作风格，对于每个新主题都有很好的介绍，书中充满了睿智的见解。干得好。

Simon Goland, simonsez@smartt.com, Simon Says 咨询公司

我得说，您的《Thinking in Java》真是了不起。它正是我要找的那种书。尤其那些讨论优秀与拙劣的Java软件设计的章节，完全就是我要的。

Dirk Duehr, 德国贝塔斯曼集团 Lexikon 公司

感谢您写了两本著作：《Thinking in C++》和《Thinking in Java》。在面向对象程序设计的学习过程中，您带给我巨大帮助。

Donald Lawson, DCL Enterprises公司

感谢您花时间来撰写这么一本很有用的Java书籍。如果是教学让您明白了某些事情的话，到如今您一定极为满意自己的成就。

Dominic Turner, GEAC Support

我曾读过的最棒的Java书籍——我真的读过不少。

Jean-Yves MENGANT, 法国巴黎 NAT-SYSTEM公司首席系统架构师

《Thinking in Java》涵盖全面，讲解清晰。本书极易阅读，而且程序代码也是如此。

Ron Chan博士, 匹兹堡 Expert Choice公司

您的书真好。我读过许多程序设计书籍，但是这本书中您对程序设计的深刻见解依然深深触动了我。

Ningjian Wang, Vanguard 集团信息系统工程师

《Thinking in Java》是一本既优秀，又容易阅读的书籍。我向所有的学生推荐它。

Dr. Paul Gorman, 新西兰 Otago 大学计算机科学系

依靠您的书，我现在已经理解了面向对象程序设计的含义……我相信，Java比Perl更直接，甚至更容易。

Torsten Römer, Orange 丹麦公司

您打破了“天下没有白吃的午餐”这句谚语。不是那种施舍性质的午餐，而是连美食家都觉得美味的午餐。他们都会为此感激您。

Jose Suriol, Scylax 公司

感谢有机会看到这本书成为一部杰作！在这个主题上，本书绝对是我所读过的最佳书籍。

Jeff Lapchinsky, Net Results 技术公司程序员

您的书简明扼要，容易理解，而且读起来充满乐趣。

Keith Ritchie, KL 集团公司Java研发组

确实是我所读过的最好的Java书籍！

Daniel Eng

生平所见最好的Java书籍！

Rich Hoffarth, West 集团高级架构师

感谢您带来了如此精彩的一本好书。通读各个章节带给我极大的乐趣。

Fred Trimble, Actium 公司

您一定掌握了艺术的精髓，使我们得以循序渐进地成功掌握细节知识。您也让学习过程变得非常简单，同时令人愉快。感谢您这本真正精彩的指南。

Rajesh Rau, 软件顾问

《Thinking in Java》撼动了整个免费世界！

Miko O'Sullivan, Idocs 公司总裁

# 关于《Thinking in C++》

最好的书! 1995年《Software Development》杂志Jolt大奖得主!

“本书成就非凡。您应该在书架上也摆一本。其中讨论输入、输出流的章节,在我所见过的有关此主题的论著中,它是表述最全面、也最容易理解的。”

Al Stevens,《Dr. Dobb's Journal》的特约编辑

“对于如何重新认识面向对象程序的构造,Eckel的这本书是唯一能做出如此清晰解释的书籍。同时,它也是透彻讲解C++的优秀教程。”

Andrew Binstock,《Unix Review》的编辑

“Bruce对C++的洞察力,不断令我感到惊讶。《Thinking in C++》则是他迄今为止所有绝妙想法的最佳合集。有关C++的种种难题,如果您需要清楚的解答,请买下这本杰作。”

Gary Entsminger,《The Tao of Objects》的作者

《Thinking in C++》耐心而系统地对C++种种特性的使用时机与方式进行了探讨。包括:内联函数、引用、操作符重载、继承、动态对象。也包括了许多高级主题,比如模板、异常、多重继承的恰当用法。对这些交织在一起,最后形成了Eckel对对象和程序设计的独特看法。它是每个C++开发者书架上的必备好书。如果您正以C++从事严肃的开发工作,那么《Thinking in C++》是您的必备书籍之一。

Richard Hale Shaw,《PC Magazine》的特约编辑

# 题 献

献给那些正在创造  
伟大的新一代计算机语言的人们

# 译者序

《Java编程思想》已经成为广大Java程序员和爱好者们手中必备的宝典，它在读者心目中的地位已经无可替代。其作者Bruce Eckel也已经成为了众多Java程序员顶礼膜拜的大师。随着Java技术的发展以及对Java认识的不断深入，Bruce Eckel推出了《Java编程思想》第3版。在这里我们应该向他致敬，他不断精益求精、锐意进取的精神正是我们努力学习的榜样。

随着软件开发技术，尤其是Web应用开发技术的发展，Java越来越受到人们的青睐。越来越多的企业都选择使用Java及其相关技术（例如J2EE技术）来构建它们的应用系统。但是，掌握好Java语言并不是一件可以轻松完成的任务，如何真正掌握Java语言，从而编写出健壮的、高效的、灵活的程序，是Java程序员们面临的重大挑战。

《Java编程思想》就是一本能够让Java程序员轻松面对这一挑战，并最终取得胜利的经典书籍。本书深入浅出、循序渐进地把我们领入Java的世界，让我们在不知不觉中学会了用Java的思想去考虑问题、解决问题。本书不仅适合Java的初学者，对于有经验的Java程序员来说，每次阅读本书也总能有新的体会，这正是本书的魅力所在。

本书的第2版由侯捷先生翻译，他已经把翻译原著这项工作做到了极致。我们在翻译过程中总是诚惶诚恐，一方面害怕曲解了原著的意思，另一方面也害怕破坏了第2版译著已经做出的让广大中国读者认可本书的种种努力。

我们在翻译本书的过程中力求忠于原著，对于书中出现的大量的专业术语尽量遵循标准的译法，并在有可能引起歧义之处著上了英文原文，以方便读者的对照理解。

全书的翻译由陈昊鹏和饶若楠合作完成，薛翔、郭嘉和方小丽也参与了全书的翻译工作。由于我们水平有限，书中出现错误与不妥之处在所难免，恳请读者批评指正。

陈昊鹏

上海交通大学软件学院

# 序 言

我的兄弟托德正准备从硬件领域转移到程序设计领域。我曾向他建议，下一场技术革命将是基因工程。

我们将使用微生物来制造食品、燃料和塑料；这些微生物不仅能清除污染，还能让我们以比现在少得多的代价，就能主宰整个物质世界。我曾以为，相比之下计算机革命将显得微不足道。

后来我意识到自己犯了科幻小说家常犯的错误：纠缠在技术细节上（当然，这在科幻小说里司空见惯）。有经验的作家都知道，故事的重点不是技术，而在于人。基因工程的确将对我们的生活造成深远的影响，但它未必就会使计算机革命（或者至少是信息革命）黯然失色，因为正是计算机技术推动了基因工程的发展。信息指的是人与人之间的沟通。当然，汽车、鞋子，包括基因疗法，这些都很重要，但它们最终都只是表面现象。人类如何与世界相处才是问题的本质。这个相处的过程就是沟通。

本书恰好就是一个例子。很多人认为我很大胆，或者有点疯狂，因为我把所有资料都放在网络上。“还有谁会购买呢？”他们这样问。如果我的性格保守谨慎一些，我就不会这么做。但是我真的不想再用传统方式来编写一本新的计算机书籍。我不知道这么做会有什么后果，但结果表明，这是我在写书经历中做过的最明智的一件事。

首先，人们开始把修正意见反馈给我。这是个令人惊叹的过程，因为人们仔细检查每个角落、每个缝隙，找出技术上和文字上的种种问题，让我得以改正所有错误，而这些错误单凭我个人是很难觉察到的。人们对这种做法还有些顾虑，他们常常说“我并无冒犯之意……”，然后扔给我一大堆错误。无疑我自己从未察觉到这些错误。我很喜欢这种集体参与的过程，这也使这本书更加特别。这些反馈很有价值，所以我使用“BackTalk”系统创建了一些应用，来对这些建议进行收集和分类。

但是，很快我就开始听到“嗯，很好。把书做成电子版放到网络上是个好主意，可是我希望购买印刷出版并装订成册的书籍”。我曾经努力过，让每个人都能很容易地打印出美观的书籍，但这样仍然无法满足人们对印刷书籍的需求。大多数人不习惯在计算机屏幕上阅读整本书，也不喜欢总是带着一捆捆的纸，所以即使这些打印出来的书籍再美观，也吸引不了他们（而且激光打印机的碳粉并不便宜）。看来即使是计算机革命，也难以抢走出版商的生意。不过，有个学生提出这也许会在未来成为一种出版模式：先在网络上出版书籍，当它引起足够的关注以后，才考虑制作纸质版本。目前，绝大多数书籍都不赚钱，这种新方式或许可以给整个出版业带来更多的利润。

这本书也以另一种方式带给我一些启迪。开始，我认为Java“只不过是另一种程序设计语言”——从许多方面看的确如此。但随着时间流逝，以及对Java学习的深入，我才开始明白，这个编程语言的目的，与我所见过的其他语言根本不同。

程序设计就是对复杂度的管理，复杂度包括：待解问题的复杂度和所依赖的底层机器的复杂度。正是因为这种复杂度的存在，导致了大多数项目的失败。并且据我所知，还没有哪个程



序设计语言全力将主要设计目标放在“征服程序开发与维护过程中的种种复杂度”上<sup>①</sup>。当然，许多编程语言设计时也确实考虑到了复杂度问题，但它总是与被视为更本质的问题混在一起。当然，那些也都是让程序员感到困扰的问题。比如，C++必须向上兼容于C（为使C程序员容易过渡），并具有执行效率高的优点。这两点都很有用，并且帮助C++取得了成功。不过，两者也引入了额外的复杂度，使得某些项目无法完成。（当然，你可以归咎于开发或管理人员，但如果有某种语言可以帮助我们找到错误，何乐而不为呢？）Visual Basic (VB) 是另一个例子，它被局限于Basic语言（它并不把可扩展能力作为设计目标），这就使得所有堆积于VB 之上的扩充功能，都造成了可怕且难以维护的语法。Perl 也向上兼容于Awk、Sed、Grep以及其他Unix工具，这就导致了被诟病为“能写不能读”的程序代码（也就是说，一段时间之后，你就无法看懂这些代码）。另一方面，C++、VB、Perl、Smalltalk 之类的编程语言，在设计上也都为复杂度问题付出了相当大的努力，因而在解决特定类型问题的时候非常成功。

当我开始理解Java 之后，印象最深的莫过于在Java上体现出的Sun公司的设计目标：为程序员降低复杂度。就好像说：“我们关注的是降低编写健壮代码的困难程度和需要的时间。”以前，编写健壮代码将导致性能降低（尽管有许多这样的承诺——Java总有一天能够执行得足够快），不过Java的确大大缩短了开发时间；相比同等功能的C++ 程序，它只需一半甚至更少的开发时间。只凭这一点，就足以省下大量的时间与金钱。不过，Java 并不仅仅如此。它又不断引入了一些日益重要的复杂任务，比如多线程和网络编程。并且通过语言本身的特性和程序库，使得这些工作变得轻而易举。最后，Java还着眼于某些有着极高复杂度的问题：跨平台程序、动态改变程序代码，甚至安全问题，它们都属于复杂度问题的重要议题。所以尽管存在已知的效率问题，Java 带来的许诺仍极其诱人：它能使程序员的生产率大大提高。

根据我的观察，Web是Java影响最远的地方之一。网络程序设计总是非常困难，但Java使之得以简化（Java的设计者仍在努力使之变得更简单）。网络程序设计所关注的，就是研究如何让我们用比使用电话更有效率、更廉价的方式进行沟通（单是电子邮件就已经使许多业务领域发生了革命性的变化）。当我们能更有效地进行沟通时，奇妙的事情就开始发生，这也许要比基因工程可能创造的前景更让人感到神奇。

编写程序、团队开发、设计用户界面（让程序可以和用户交互）、跨平台执行、轻松编写跨因特网的通信程序等，Java给我们带来的所有这一切扩展了人与人之间的通信带宽。我认为，通信革命的成果也许不应以海量数据的传输来衡量；我们将体会到真正的革命，因为我们能更容易地和他人沟通——可以是一对一的形式，可以是分组形式，也可以是全球通信的形式。有人曾说过，随着足够多的人之间产生了足够多的相互联系，下一次革命将会是一种全球化思维的形成。Java可能是、也可能不是这场革命的导火索，但至少这种可能性是存在的，这使我觉得，教授这门语言是一件非常有意义的事情。

### 第3版序

这一版的主要目的和大量工作都用在了与Java 1.4版保持同步上。不过，本书的主旨也更加清楚：使大多数读者通过本书牢牢抓住语言的基本概念，以便深入学习更复杂的主题。因为Java还在不断地演化，所以重新定义“基础知识”的涵义就很有必要，再说本书也不应过于面

<sup>①</sup> 在本书第2版我收回了这句话：我认为Python语言非常接近这个目标。请参考[www.Python.org](http://www.Python.org)。