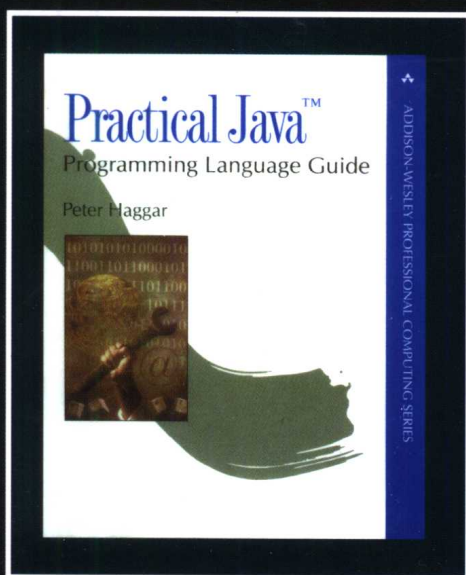


Practical Java  
Programming Language Guide

# Practical Java

## 中文版

[ 美 ] Peter Haggart 著  
侯捷 刘永丹 译



- 出自第一线 Java 编程专家之手
- 68 个改善代码的重要主题
- 所有示例和方案均有详尽解说



开 发 大 师 系 列

**Practical Java**

**Programming Language Guide**

**Practical Java**

**中文版**

[ 美 ] Peter Haggard 著

侯捷 刘永丹 译

中国电力出版社

Practical Java Programming Language Guide (ISBN 0-201-61646-7)

Peter Haggar

Authorized translation from the English language edition, entitled Practical Java Programming Language Guide, published by Addison Wesley Longman, Copyright © 2000.

All rights reserved.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

CHINESE SIMPLIFIED language edition published by China Electric Power Press Copyright © 2003.

本书由美国培生集团授权出版。

北京市版权局著作权合同登记号 图字：01-2000-3084 号

### 图书在版编目 (CIP) 数据

Practical Java 中文版 / (美) 海格尔编著; 侯捷、刘永丹译. —北京: 中国电力出版社, 2001

ISBN 7-5083-0739-9

I. P... II. ①海...②侯...□刘... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 066547 号

责任编辑: 朱恩从

书 名: Practical Java 中文版

编 著: (美) Peter Haggar

翻 译: 侯捷 刘永丹

出版发行: 中国电力出版社

地址: 北京市三里河路6号 邮政编码: 100044

电话: (010) 88515918 传真: (010) 88518169

本书如有印装质量问题, 我社负责退换

印 刷: 北京地矿印刷厂

开 本: 787×1092 1/16 印 张: 19.5 字 数: 446千字

书 号: ISBN 7-5083-0739-9

版 次: 2003年11月北京第一版

印 次: 2003年11月第一次印刷

定 价: 46.00 元

版权所有, 翻印必究

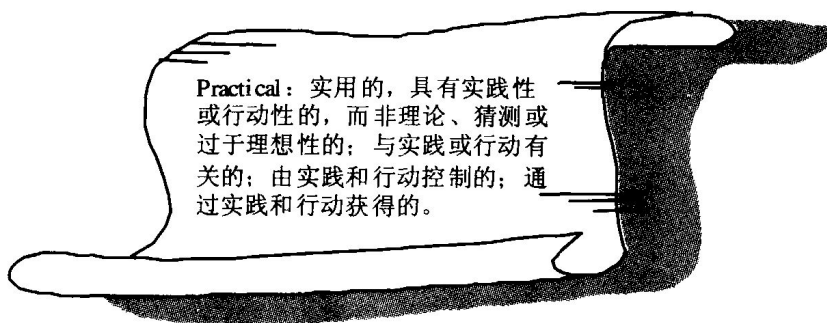
---

教师的影响是永恒的，其价值难以计算

— Henry Adams

……纪念我的父亲

谨以此书献给我的妻子 Tara，以及  
我的孩子们，Lauren，Keely 和 Andrew



**Practical:** 实用的，具有实践性或行动性的，而非理论、猜测或过于理想性的；与实践或行动有关的；由实践和行动控制的；通过实践和行动获得的。

**Practical:** Of, relating to, governed by, or acquired through practice or action, rather than theory, speculation, or ideals.

# 译序

by 侯捷

面对 Java，可从两方面看待，一是语言，一是平台。本书谈的是 Java 语言，以下我所言种种，也是指 Java 语言。

Java 是一门优秀的面向对象编程语言（Object Oriented Programming Language, OOP）。什么是「面向对象」？如何才称得上「优秀」？前者可定量定性，客观；后者往往流于个人感受，主观！所以虽然面向对象语言有着几近一致的条件和门坎<sup>1</sup>（封装、继承、多态……），孰优孰劣却是各人心中一把尺。尽管如此，无人可以否认 Java 语言在 OOP（面向对象编程）上拥有良好的特性和优越的表现。

我所谓良好的 OOP 特性，指的是 Java 提供了许多让程序员得以轻松表达面向对象技术与思维的语言关键词(keywords)如 class、abstract、interface、extends、implements、public、protected、private、final、static、finalize……，又提供条理清晰结构分明的文件组织方式如 package 和 import，又拥有严谨而灵活的动态型别系统(dynamic type system)使得可以提供 RTTI 和 Reflection 机制，并拥有一个优秀、涵盖面广、扩充性强的标准程序库（Java Libraries）。

这些优秀的语言构件（constructs）虽然好用易用，但其技术面、应用面，以及性能考量还有许多隐微细节散布其中，例如 object creation、object initialization、Cloneable、Serializable、Equality、Immutability、Multithreading (Synchronization)、

---

<sup>1</sup> 我常忆起网络论坛上时可与闻的一种怪诞态度。有一派人士主张，OO 是一种思想，一种思考模式，任何语言都能够实现它，因而侈言「C 或 assembly 语言也能 OO」。任何语言各有用途，这是完全正确的；OO 是一种思维，这话也是对的。任何语言都能够实现 OO，这话对某些人也许是对的，但对 99.9999% 的人是错的。以 non-OO 语言实现 OO 思维，非但达成度极低，也非人人能为。Edmund Hillary（埃德蒙·希拉瑞）能达到的高度，你未必达得到——事实上你通常达不到。（注：Edmund Hillary 是第一位登上圣母峰的地球人，1953 年英格兰远征队员。）

Exception Handling……，实在需要 Java 程序员深入认识与理解。

市面上 Java 书籍极多，专注于「编程主题式探讨」并「以独立条款呈现」的书籍比较少。这类书籍面向中高阶读者，不仅选题必须饶富价值、探讨必须极为深刻，各主题最好还独立以利选择阅读，而且最好彼此前后呼应环环相扣，并附良好交叉索引，予读者柳暗花明的强烈冲击。此种「专题条款」式的表现风格，在 Scott Meyers 的《Effective C++》和《More Effective C++》二书面世之后获得许多赞扬，也引来许多追随。

本书和《Effective Java》，对前述重要而基础的技术细微处有着详尽、深刻、实用的介绍、剖析和范例，又以独立条款之姿展现，在内容的扎实度、可读性及易读性上的表现均十分良好。为此，秉持并承继我为 C++ 社区翻译《Effective C++》、《More Effective C++》的态度和机缘，我很开心再次由我负责，将本书和《Effective Java》二书中译本呈献给 Java 社区。

考虑本书读者应已具备 Java 编程基础，对于各种英文术语已有良好的接受度，我在书中保留了许多英文术语，时而中英并陈，包括 class、object、interface、reference、instance、array、vector、stack、heap……，还包括涉及 Java 关键词的一些用语如 private、public、protected、static、abstract……，不胜枚举（下页另有一个扼要说明）。本书努力在字型变化上突显不同类型的术语，以利读者阅读。本书支持网站有一个「术语·英中繁简」对照表，欢迎访问，网址如下。

<http://www.jjhou.com>

本书由刘永丹先生和我合力完成。永丹做前期初译工作，我负责后继的文字修润、技术检阅、大局风貌。永丹技术扎实，文字用心。没有他的协助，本书不可能在这个时间以这样的质量面世。谢谢永丹。

本书每一章起始处都有作者匠心独具收集的一些文摘语录。我们虽勉力译出，恐见识不足，贻笑大方，故均留下原文和出处，庶几不误读者。

侯捷 2003/07/08 于台湾·新竹

[jjhou@jjhou.com](mailto:jjhou@jjhou.com)（电子邮箱）

<http://www.jjhou.com>（繁体）（术语对照表 <http://www.jjhou.com/terms.htm>）

<http://jjhou.csdn.net>（简体）（术语对照表 <http://jjhou.csdn.net/terms.htm>）

p.s. 本书已就英文版截至 2003/07/08 之勘误表修正于纸本。

本书术语翻译与保留之大致原则：

- ※ 广被大众接受之术语，无需额外说明，不在此列。例如继承（inheritance）、封装（encapsulation）及多态（polymorphism）。
- ※ 本书保留与 Java 关键词相关之术语不译，例如 class、interface、private、public、protected、static、final、abstract、synchronized、serializable……。
- ※ 本书保留数据结构名称不译，例如 array、vector、list、map、set、stack、heap……。 “collection” 译为「群集」。
- ※ “class” 及其所衍生之各种名词如 subclass、superclass、immutable class、mutable class、base class 及 derived class 等皆保留不译（时而英中并陈）。“object” 大多数时候译为「对象」，时而保留。“object reference” 保留不译，“reference” 亦不译。
- ※ “type” 译为「型别」。“parameter” 译为「参数」，“argument” 译为「引数」。“delegate”，“delegation” 译为「委托」，“aggregate”，“aggregation” 译为「聚合」。“composition” 译为「复合」。
- ※ 动词 “create” 译为「创建」或「建立」，描述对象之初次诞生。动词 “refer” 译为「指涉」、「指向」或「引用」。动词 “dereference” 译为「提领」。动词 “override” 译为「覆写」。动词 “overload” 译为「重载」。
- ※ 本书将 Java class “methods” 译为函数，因为它等价于其他编程语言之 “function”。若直译为「方法」，行文缺乏术语突出感，恐影响阅读流畅；若不译，过于频繁出现又恐影响版面观感。
- ※ 本书将 Java class “fields” 译为值域，等价于 C++ 语言之 “data member”。
- ※ 本书将 “clone” 译为「克隆」，映照 “copy” 之于「拷贝」。非单纯保留 “clone” 是因为它时常作为动词并频繁出现，而我对术语的保留态度是尽量只考虑名词（偶有形容词）。
- ※ 「static 值域与 instance 值域」、「reference 对象与 value 对象」、「reference 型别与 primitive 型别」等等术语保留部分英文，并使用特殊字型。
- ※ 本书支持网站有一个「术语•英中繁简」对照表，欢迎访问，网址见上页。
- ※ 术语翻译有许多两难之处，祈愿读者体谅；译者勉力求取各方平衡，并尽可能于突兀处中英并陈。

——侯捷



# 译序

by 刘永丹

拿在您手中的，是一本每一位 Java 程序员都应该关注的书。

要想充分驾驭一门语言（譬如英语），「语法」、「语义」和「语用」缺一不可，即使像 Java 这样的计算机语言也是如此。Java 称得上是一门贴心的语言，功能强大却又不失简洁优美，「语法」这道门坎似乎轻易就能跨过去。但「语义」往往隐藏在语法之后，需要仔细体察才行，初学者通常需要找来一本优秀教程仔细研读和领会。「语用」则涉及多种语言特性的综合运用，有的深入语言的实现机理（例如 Java 的运行性能），有的甚至超出语言本身的范畴（例如多线程）。因此「语用」的掌握绝非一朝一夕之功，无法一蹴而就，最好有位资深专家时刻陪伴左右，耳提面命。

并不是每位程序员都有这样的福气，好在有一位资深的 Java 编程专家，Peter Haggart，将其多年的实践经验汇集整理，撰写了您手上的这本书。

本书以小品文形式，帮助程序员理解 Java 语言中最核心的「语义」、「语用」专题，因此本书的定位是已初窥门径的中级程序员。即使您是位 Java 初学者（但一定有随心所欲驾驭 Java 语言的愿望），也请关注这本书，因为您很快就需要它了。

正如书名所喻示，本书围绕 Java 编程中遇到的实际问题展开。可以说书中所列专题正是那些令许多 Java 程序员困惑不已的 FAQ（常见问答集）。作者擅长采用恰如其分的示例来阐释问题，以平实的语言娓娓道出中肯的建议。这些建议往往一语中的，能直接融入编程工作中，可见它们确实是出自作者从实践中获得的真知灼见。

本书的另一大特色在于，它并不因为其实用性而影响讨论的深度和说理的透彻，相反，本书对深奥的论题做了聪明的分解，逐步进行细致入微的探讨，这正是作者的高明之处。作者只是以实用性为界限，摒弃了华而不实、枯燥乏味的理论说教，体现了 **Peter Haggar** 务实的写作风格。

与侯捷先生合译本书是我的荣幸。侯先生对技术严谨认真，一丝不苟，为人却极其谦和，与其交往的经历令我受益良多。还要感谢周筠老师，她不断给与我鼓励和关注，每一次与她交谈都是如沐春风。

刘永丹 于上海

2003/05/19

# 前言

## Preface

让无知尽管信口开河吧，学习自有其价值。

*Let ignorance talk as it will, learning has its value.*

—J. de La Fontaine, *The Use of Knowledge*, Book viii, Fable 19

本书汇集了 Java 编程实践方面的建议、忠告、范例和讨论。本书的组织是一个个独立课程，每个课程谓之实践（PRAXIS，发音 prak-sis），用以讨论特定主题。每个实践按各自独立的方式撰写。你可以从头阅读到尾，也可以挑选某些专题阅读。这种编排风格使你可以在短暂的闲暇中阅读此书。许多实践都少于 5 页，因此你可以在简短的时间内学习它们。

我在这本书中详细分析了某些设计（design）和编程（programming）方面的问题。我挑选主题的依据是编程实践上的有效（effective）和高效（efficient）性质。Java 最被人抱怨的一点是性能（performance），因此我以最大的篇幅讨论这一主题，探索使 Java 代码运行得更富性能的技术。

我撰写本书，希望它能够作为指南，帮助你设计和撰写程序。它可以帮助你更全面地理解 Java，让你撰写出更高效、更健壮和（或许最重要的）更正确的代码。

本书所有信息都适用于各种 Java 编程，并不囿于服务器（server）、客户端（client）或 GUI（图形用户界面）编程。此外，你可以将这些信息运用于 Java 的任一发行版本。

本书风格受到 Scott Meyers 所著的《Effective C++》和《More Effective C++》的影响<sup>1</sup>。我发现他的风格对书籍组织非常有益，因此我决定采用类似的格式。

## 预期读者

本书是为已经掌握 Java 语言基础知识的程序员准备的。我假设读者已经具备 Java 语言和并发编程（concurrent programming）的工作经验，并理解面向对象（object-oriented）的概念和术语。本书适用于「想获得如何高效使用 Java 之实用建议、讨论和范例」的程序员。

无论对 Java 编程老手或新手，本书都为他们提供了 Java 关键领域的信息和讨论。本书提供了充足的新信息，即使经验丰富的程序员也能从考查他们业已熟悉的领域中获得极大收益。例如在某些场合，我以独特的方式讨论问题，帮助程序员以不同的方法思考，或使用与以往不同的角度看待事物。

初入门的程序员也可以从本书获益良多。我提供了讨论和范例，帮助他们消除许多常见的编程错误。同时还澄清了某些常见的 Java 错误观念，并强调了语言特性方面的某些问题。

## 本书组织

本书组织为六大部分。

1. 一般技术——展现 Java 编程的数个基础领域，包括参数传递（parameter passing）、arrays、vectors，以及垃圾回收（garbage collection）。
2. 对象与相等性——研究对象（objects）、基础型别（primitive types），以及如何、为何为一个 class 实现 equals()。

---

<sup>1</sup> Effective C++: 50 Specific Ways to Improve Your Programs and Designs, Second Edition, Scott Meyers, Addison-Wesley, 1998. More Effective C++: 35 New Ways to Improve Your Programs and Designs, Second Edition, Scott Meyers, Addison-Wesley, 1996.

3. 异常处理——提供异常处理技术(exception handling techniques)的细致分析,并告诉你如何在你的代码中高效加入异常处理机制。
4. 性能——展示可用来改善代码性能的诸多技术,并仔细审查 JVM (Java Virtual Machine, Java 虚拟机器)、byte code 和 JITs (Just-in-Time code generators)。
5. 多线程——涵盖线程模型(threading model)的诸多方面,这些问题对于建立健壮、可靠的多线程应用程序极为关键。
6. 类与接口——解释了 interfaces、abstract classes 和 concrete classes,以及何处、何时使用它们。本章还详细讨论了 immutable object (不可变对象)、cloning (克隆) 和 finalization (终结)。

在上述各标题之下,是数量不等的相关专题。往往我会在不止一处讨论特定专题的某个性质。例如我在不同场合讨论了 **synchronized** 关键词的方方面面,每次讨论都涉及 **synchronized** 的不同特性。我提供了数量庞大的交叉参照,你可以由此得知何时阅读特定专题以及何处存在相关信息。

目录之后便是细目。这一部分包含所有实践标题及其页码,并附有每个实践的核心摘要。你可以利用这个细目唤起你对专题的记忆,或用以找出某个专题。

本书附录内附了一份已经证实的技术,可以进一步扩展你的 Java 知识。之后还有一份「进阶读物」清单,列出了关于 Java、一般设计及编程方面的书籍和期刊。

## 三言两语话 PRAXIS (实践)

PRAXIS (实践) 一词,是我搜寻「得以概括本书所做工作」的词汇的结果。1982 年的《*American Heritage Dictionary*》将 PRAXIS 定义为: The practical application or exercise of a branch of learning (实际应用或训练; 学习的一个支脉)。这正是我希望在本书中达到的目标。

最确切的恐怕是《*Webster's New Collegiate Dictionary*》于 1958 年给出的一份定义: Practice, especially of an art, science, or technical occupation; opposite to theory (实际履行, 尤指艺术、科学或技术领域; 与理论遥相对应)。这个定义准确概括了本书精髓。那句“opposite to theory”更是画龙点睛。「理论」本身当然没错,但本书没有为它准备位置。

## 范例代码

正文所列的所有代码，都采用本书写作时可获得之 Java 最新版本加以编译和运行。所有代码都曾经在 Windows NT 4.0 环境下以 Sun Java 2 SDK、Standard Edition, v1.2.1 完成编译和运行。如果你想要得到源码，请在以下网址进行注册：

<http://www.awl.com/cseng/register>

该网页要求你输入一个独一无二的码，此码可在本书末尾标明为「How to Register Your Book」的页面找到。（译注：中文版读者可使用这个码：csng-tbtw-hxqr-xlnz，或自中文版支持网站（见本书封底）之本书专属网页下载。）

## 提供反馈

欢迎读者对本书提供相关反馈信息。任何建议、批评或臭虫报告，都请寄到 [PracticalJava@awl.com](mailto:PracticalJava@awl.com)。

希望本书让你觉得有用、可读，并且具实用价值。

*Peter Haggar  
Research Triangle Park, North Carolina  
November, 1999*

# 致谢

## Acknowledgment

谢谢你，善良的人，我欠你一份情。

*Thank you, good sir, I owe you one.*

— George Colman, *The Poor Gentleman*, Act I, Scene ii

写书不是一个人可以独自完成的。为了让本书做成现在这个样子，许多人付出了他们的时间和精力。审阅书稿可不是件好差事，尤其是那些初期草稿（它们真的是「草稿」）。我真挚地感谢我的审阅人所表现的耐心和热情。

许多人阅读、覆阅、（有时）再覆阅（部分）书稿，为我提供了极具价值和极为深刻的反馈信息。这些反馈包括技术乃至语法等各个层面，对于本书内容和脉络的形成提供了极大帮助。

本书部分草稿由以下人士审阅：Kimberly Bobrow、Joan Boone、Tom Cargill、Bill Field、Dion Gillard、David Hardin、Howard Lee Harkness、Tim Lindholm、George Malek、Jim Mickelson、Devang Patel、Warren Ristow、Susan Elliot Sim 和 Dan Trieschman。

本书完整书稿由以下人士审阅：Larry Collins、Trevor Cushen、Mary Dageforde、Joshua Engel、Elisabeth Freeman、Cay Horstmann、Susanne Hupfer、Brian Kernighan、Bob Love、Judy Oakley、Linda Rochelle、Clayton Sims、Robert Stanger 和 Steve Vinoski。

尽管我幸运地拥有如此出类拔萃的审阅人，但任何可能的错误还是应由我单独负责。

在本书制作过程中，Addison-Wesley 的整个团队一直是可信赖的引导力量。他们自始至终提供无法估量的帮助和鼓励，我为此对他们感激不尽。这个团队包括我的编辑 Mike Hendrickson、Julie DeBaggis、Tracy Russ，以及 Jacquelyn Doucette。

我也要感谢我在 IBM 的经理，Carolyn Ruby 和 John Feller，感谢他们在我写作本书的过程中提供的支持。

特别要感谢 Rosemary Simpson，她用娴熟的技术制作出一流的索引。

我必须感谢 Wayne Kovsky，Colorado Software Summit Java 会议的创办人。正是由于我有机会作为其中的讲师，和与会人士有所交流，从中磨砺了许多想法，受到了启发，才能够承接这个项目。

同样要感谢 Larry、David 和 Mark，他们设法补足了我在高尔夫四人对抗赛中的缺席。在我缺席的数月中，他们表现出极大耐心。值得赞颂的是，他们保留了我对抗组中的位置，我急切盼望回到队伍之中。

就像任何项目一样，总有一些人特别突出。他们的贡献远在他人之上，总是救人于急难。在本书写作期间，他们的词汇表中似乎没有“no”这个词。无论何时，当我寻求帮助、建议或要求他们就我所重写的某一段实践再次复审，他们总是在百忙之中抽出时间提供协助。首先是 Bob Love，他除了提供出色的技术评阅，还提供正确的看法，在我偏离航线时不厌其烦地将我引入正轨。他和 Judy Oakley、Robert Stanger 在实践 11 至实践 15 的制作过程中展现出无法估量的价值。Clayton Sims 在本书写作过程中不但从事审阅，还提供了无穷的热情和鼓励。我要献给 Larry Collins 一份特别的感谢，他为本书的许多方面提供了及时的支持和帮助。他在任何时候都激励我，并且乐于助人，让我感激不已。Bob、Clayton 和 Larray 对他们的耳朵也很慷慨大方——是的，我不止一次地对他们喋喋不休，唠叨个没完。

最重要的，我必须感谢在奋斗过程中忍受最多的另外四个人：我的妻子 Tara 和三个孩子 Lauren、Keely 和 Andrew。

最后，我想向我最大的孩子 Lauren 表达谢意，我撰写本书时她正在学习阅读。我的「故事」（她这样称呼它）终于完成了。



## 个别实践 (PRAXIS) 的致谢

有些人并不知道他们对本书提供了帮助。探究本书内容时, 我阅读了他们的观点。他们给予我某个实践的某些思路, 下面是这些人士的清单以及他们的工作。

实践 8 的灵感来自 Sherman Alpert 的文章 “Primitive Types Considered Harmful”, 刊载于《Java Report》, 1998 年 11 月号。

实践 12 受到 Mark Davis 所写的 “Java Cookbook: Porting C++ to Java” 一文影响。此文可在以下网址找到:

<http://www.ibm.com/java/education/portingc/index.html>

实践 11 至实践 15 深深受到我与 Bob Love 和 Robert Sranger 之间为数众多、篇幅漫长的讨论和电子邮件的影响。

我于《Java Report》, 1999 年 4 月号上发表文章 “Effective Exception Handling in Java”, 之后便收到了来自 Prescott Sanders、Rick Kitts 和 Brian Dellert 的电子邮件。Prescott 和 Rick 提出对 finally 区段处理净化的建议, 被我包含于实践 22 中。Brian 提出关闭 JIT (turning off the JIT) 方面的建议, 被我包含于实践 23 中。

实践 27 受到 Tom Cargill 的文章 “Exception Handling: A False Sense of Security” 的启发, 此文发表于《C++ Report》1994 年 11-12 月号。

实践 33 的灵感来自于 Scott Meyers 的《Effective C++》(第 2 版, Addison-Wesley) 条款 32。

实践 35 从 Paul van Keep 的 “Java Performance” 讲稿得到启发, 此文于 1998 年在科罗拉多州的 Keystone 举办的 Colorado Software Summit Java 会议上发表。

实践 38 受到文章 “Java: Memories Are Made of This” 的影响, 它由 Alex McManus 发表于《Java Report》1998 年 11 月号。

实践 39 的灵感来自 Kevin Clark 在 1998 ChicagoLand Java Users Group 上所给的讲稿。

实践 43 包含缓式评估 (lazy evaluation, 延迟求值) 的讨论和一些范例, 它们以 Scott Meyers 的《More Effective C++》(Addison-Wesley) 条款 17 为基础。