

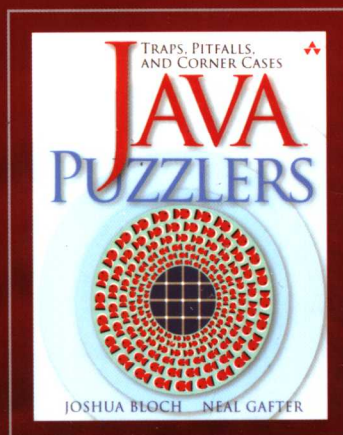
Java Puzzlers

Traps, Pitfalls, and Corner Cases

Java 解惑

[美] Joshua Bloch 著
Neal Gafter 译
陈昊鹏 译

- *Effective Java* 作者又一力作
- 实例讲授 Java 中令人迷惑和不易掌握的知识点
- 寓教于乐，妙趣横生



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Java解惑

Java Puzzlers

Traps, Pitfalls, and Corner Cases

[美] Joshua Bloch 著
Neal Gafter
陈昊鹏 译



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

Java 解惑 / (美) 布洛赫, (美) 加夫特著; 陈昊鹏译. —北京: 人民邮电出版社, 2006.1
(图灵程序设计丛书)

ISBN 7-115-14241-6

I. J... II. ①布...②加...③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 152708 号

内 容 提 要

本书特写了95个有关Java或其类库的陷阱和缺陷的谜题, 其中大多数谜题都采用短程序的形式给出, 这些程序的实际行为与表面上大相径庭。在每个谜题之后都给出了详细的解惑方案, 这些解惑方案超越了对程序行为的简单解释, 向读者展示了如何一劳永逸地避免底层的陷阱与缺陷。

本书趣味十足、寓教于乐, 适合于具备Java知识的学习者和有编程经验的Java程序员。

图灵程序设计丛书

Java 解惑

-
- ◆ 著 [美] Joshua Bloch Neal Gafter
译 陈昊鹏
责任编辑 舒立
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 800×1000 1/16
印张: 18.75
字数: 418千字
印数: 1~5000册

2006年1月第1版
2006年1月北京第1次印刷

著作权合同登记号 图字: 01-2005-5230 号

ISBN 7-115-14241-6/TP · 5118

定价: 39.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

版 权 声 明

Authorized translation from the English language edition, entitled: *JAVA PUZZLERS: TRAPS, PITFALLS, AND CORNER CASES*, 1st Edition, 032133678X by BLOCH, JOSHUA; GAFTER, NEAL; published by Pearson Education, Inc., publishing as Addison-Wesley Professional.

Copyright © 2005 Pearson Education, Inc.

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 Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2006.

本书中文简体字版由Pearson Education Asia Ltd.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

本书封底贴有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。

译 者 序

Java已经成为越来越多的程序员首选的编程语言，很多程序员都是从C++转到了Java。乍一看，Java与C++的理念和语法都很相似，于是很多程序员都认为Java很容易掌握，但是事实并非如此，像本书中所列举的谜题就不是那么容易解决的。因此，如何真正掌握好Java，尤其是掌握好一些似是而非的知识点，就成为了一个重要的课题。

《Java解惑》这本书以轻松诙谐的语言、简单明了的方式和趣味十足的实例向我们介绍了Java编程语言中许多不易被掌握的知识点，其覆盖面几乎涉及Java编程语言的各个角落。本书不仅指出了造成这些谜题的原因，而且深入探讨了解决这些谜题的方案，有时解决方案甚至不止一种，进而进行总结，归纳出一般的规则和警告。这些规则和警告不仅包括给程序员的部分，还包括给Java语言设计者和API编写者的部分。因而，本书是一本循序渐进、由浅入深和总结归纳的书籍，其阅读价值非常高。

在翻译本书的过程中，我们尽量保持了原书的写作风格，在原文过于简练的极个别地方，适当加入了解释性的语句。由于水平有限，书中难免有不足之处，欢迎广大读者指正。

本书由陈昊鹏翻译，章程、李楠在翻译和校稿的过程中给予了很大的帮助。

陈昊鹏

2005年11月

前言

与许多书一样，本书经历了长期的酝酿过程。我们收集Java谜题的时间与我们使用Java这种平台的时间一样长：如果你感兴趣的话，可以告诉你是从1996年中开始至今。在2001年初，我们产生了一个想法：搞一次完全由Java谜题所构成的演讲。我们把这个想法抛给了当时还在Oracle公司的Larry Jacobs，他对这个想法完全买账。

2001年11月在旧金山，我们在Oracle Open World会议上首次作了题为“Java谜题”的演讲。为了增添魅力，我们介绍自己是“Type-it兄弟，Click和Hack”，并且从Tom和Ray Magliozzi主持的Car Talk节目中借用了一大堆的笑话¹。这个演讲被投票选为最佳演讲秀，即使我们不投自己的票结果可能也会如此。由此我们知道找对了路子。

从头到脚穿着蓝领工人那利索的制服，胸前装饰着Java的“咖啡杯”标志，我们在JavaOne 2002上再次利用在Oracle会议上的演讲来鼓吹我们的观点——至少我们的朋友是这么认为的。在接下来的年头里，我们又拿出了另外3个“Java谜题”演讲，并且在数不胜数的会议、公司和大学里宣讲它们，足迹遍及全球许多城市，从奥斯陆到东京。这些演讲几乎得到了普遍的欢迎，几乎没人冲我们扔烂苹果。在*Linux Magazine* 2003年3月刊上，我们发表了一篇完全由Java谜题构成的文章，并且几乎没有收到任何厌恶我们的邮件。本书几乎包含了我们的演讲和文章中的所有谜题，以及许许多多其他的谜题。

尽管本书把注意力放到了Java平台的陷阱和缺陷上，但是我们并不是要以任何方式来诋毁Java。因为热爱Java，我们将近10年的职业生涯都奉献给了它。每一种具有强大能力的平台都会有某些问题，Java与大多数平台相比问题已经少多了。你对问题理解得越透彻，你就越不会受到它们的影响，这正是本书要达到的目的。

本书中的多数谜题都是一些很短的程序，这些程序看起来在“明修栈道”，但实际却“暗渡陈仓”。这就是为什么我们选择视觉幻图来装饰本书的原因，这些幻图看起来是某样事物，但实际上却是另外一件东西。你在努力思考这些程序到底在做什么的时候，去盯着这些幻图好好看看。

毕竟，我们希望本书具有趣味性，真诚地希望你能够尽情享受解惑的乐趣，就像我们尽情享受编写它们的乐趣一样，还希望你能够从中学到很多东西，如我们曾经的那样。

不管怎样，请把你发现的谜题发给我们！如果你有一个你认为应该囊括到本书将来的版本中的谜题，请把它写到一张20美元的账单后面，然后寄给我们，或者发E-mail到puzzlers@javapuzzlers.com。如果采用了你发现的谜题，我们将向你付账。

最后要说的，但不是惟一要说的，就是请不要像我的兄弟那样编写代码。

1. Car Talk是美国NPR电台关于汽车的著名节目，在美国拥有400多万听众。主持人Magliozzi兄弟被称为“Click and Hack, the Tappet Brothers”，名称源自破旧汽车的声音。——编者注

致 谢

我们要感谢Addison-Wesley的整个团队，感谢他们的善举和专业精神。在这个项目的早期，Ann Sellers是我们的编辑。她极富感染力的热情帮助这个项目开了个好头。当Ann离开之后，执行编辑Greg Doench接手了这个项目。Greg是一位令人愉快的编辑和完美的绅士，他眼睛都没眨一下就答应了这个项目的许多要求。Greg的编辑助理是Noreen Regina。我们这本书的项目编辑是Tyrrell Albaugh，营销经理是Stephane Nakib。封面设计是Chuti Prasertsith，加工编辑是Evelyn Pyle。他们在非常紧张的进度安排下都出色地完成了任务。

我们要感谢Google公司管理层的支持。我们的主管Prabha Krishna不断地给我们以信心。我们要感谢Sergey Brin、Larry Page和Eric Schmidt，感谢他们为我们创建了在这个地球上最好的工程环境。

我们要感谢长年累月地给Sun公司发送bug报告的许多Java程序员，特别是那些提交的bug报告被证明为并非真正bug的程序员。这些bug报告也许是谜题资料最丰富的来源：如果正确的行为会误导程序员认为他们发现了一个bug，那么这很可能就是一个陷阱或缺陷。我们以同样的心情感谢Sun公司，感谢她在1996年以非凡的勇气和智慧将完整的Java bug数据库放到网上[Bug]。这个举动是前所未有的，即便是在今天也很少碰到。

“请把你的谜题发送给我们。”我们在每一次演讲结束时都会这么说，而大家的确给我们发来了，从世界各地。特别要感谢Ron Gabor和Mike “madbot” McCloskey，感谢他们所做出的极大贡献。Ron贡献了谜题28、29、30和31，Mike贡献了谜题18、23、40、56和67。我们要感谢Martin Buchholz贡献了谜题81，Armand Dijamco贡献了谜题14，Dominik Gruntz教授（博士）贡献了谜题68和69，Kai Huang贡献了谜题77，Jim Hugunin贡献了谜题45，Tim Huske贡献了谜题41，Peter Kessler贡献了谜题35，Michael Klobe贡献了谜题59，Magnus Lundgren贡献了谜题84，Scott Seligman贡献了谜题22，Peter Stout贡献了谜题39，Michael Tennes贡献了谜题70，Martin Traverso贡献了谜题54。

我们要感谢极富奉献精神的审稿团队，他们在本书还处于粗加工阶段就阅读了每一章：Peter von der Ahé、Pablo Belver、Tracy Bialik、Cindy Bloch、Dan Bloch、Beth Bottos、Joe Bowbeer、Joe Darcy、Bob Evans、Brian Goetz、Tim Halloran、Barry Hayes、Tim Huske、Akiyoshi Kitaoka、Chris Lopez、Mike “madbot” McCloskey、Michael Nielsen、Tim Peierls、Peter Rathmann、Russ Rufer、Steve Schirripa、Yoshiki Shibata、Marshall Spight、Guy Steele、Dean Sutherland、Mark Taylor、Darlene Wallach和Frank Yellin。他们或是发现了缺点，或是给出了改进意见，或是给予了鼓励，或是猛烈地抨击。任何遗留下来的缺点都是我的合著者的失误：)

我们要感谢博客女皇Mary Smaragdis，感谢她在其著名的博客网站上为我们提供了一个舞台 [Mary Blog]。她亲切地让我们在其读者中测试构成谜题43、53、73、87和94的资料，

我们对解惑方案进行判定，Mary则负责发放奖品。根据记录，获奖者有Tom Hawtin、Tom Hawtin（再次获奖）、Bob “Crazybob” Lee、Chris Nokleberg和来自乌克兰敖德萨的神秘人物AT。在Mary博客网站上的这些讨论对上述谜题贡献良多。

我们要感谢对Java谜题长年累月地投以极大热情的许多支持者。SDForum Java SIG的成员是我们每一个演讲在雏形阶段的试听者。JavaOne的程序委员会为这些演讲提供了一个讲台。Yuka Kamiya和Masayoshi Okutsu使“Java 谜题”演讲在日本取得了成功，在那里他们以在讲台上互相争论的形式将演讲变成了真正的游戏。引人注目的是，获得每一次辩论胜利的都是同一个人：日本无可争辩的Java谜题冠军Yasuhiro Endoh。

我们要感谢James Gosling和其他很多优秀的工程师，他们创造了Java平台，并且在年复一年地不断改进它。像本书这样的书只有对坚如磐石的平台才有意义。没有Java，就没有“Java谜题”。

在Google公司、Sun公司和其他地方的大量同行都参与了改进本书质量的技术讨论。在其他人中，Peter von der Ahé、Dan Bloch和Gilad Bracha贡献了非常有用的见识。我们要特别感谢Doug Lea，他是本书中许多想法的大力宣传者，再次感谢Doug不倦地慷慨大方地贡献自己的时间和知识。

我们要感谢日本京都的Ritsumeikan 大学心理学系的Akiyoshi Kitaoka教授，感谢他允许我们使用他的一些视觉幻图来装饰本书。Kitaoka教授的幻图非常简单，但是却令人震撼。语言已经不能对它们进行描述了，所以你应该自己去看一看。他编写的两本书[Kitaoka02, Kitaoka03]可以在日本买到，包含这两本书内容的英译本不久即将面市[Kitaoka05]。同时，你可以访问他的网站：<http://www.ritsumei.ac.jp/~akitaoka/index-e.html>。你不会失望的。

我们要感谢主持Car Talk节目的Tom和Ray Magliozzi，感谢他们为我们提供了借用的笑话，我们还要感谢他们的法律顾问Dewey、Cheetham和Howe，感谢他们没有起诉我们。

我们要感谢Josh的妻子Cindy，她帮助我们使用FrameMaker排版，并帮我们编写了索引，还帮我们编辑了本书，并且还设计了每一章开始处的装饰条。最后我们要说的，但决不是惟一要说的，就是我们的家庭：Cindy、Tim和Matt Bloch，以及Ricki Lee、Sarah和Hannah Gafter，感谢他们对我们的鼓励和包容。

Joshua Bloch

Neal Gafter

加利福尼亚州圣何塞市

2005年5月

目 录

第 1 章 绪论	1
第 2 章 表达式之谜	5
谜题1: 奇数性	5
谜题2: 找零时刻	7
谜题3: 长整除	9
谜题4: 初级问题	11
谜题5: 十六进制的趣事	13
谜题6: 多重转型	15
谜题7: 互换内容	17
谜题8: Dos Equis	19
谜题9: 半斤	21
谜题10: 八两	23
第 3 章 字符之谜	25
谜题11: 最后的笑声	25
谜题12: ABC	27
谜题13: 动物庄园	29
谜题14: 转义字符的溃败	31
谜题15: 令人晕头转向的Hello	33
谜题16: 行打印程序	35
谜题17: 嗯?	37
谜题18: 字符串奶酪	39
谜题19: 漂亮的火花(块注释符)	41
谜题20: 我的类是什么	43
谜题21: 我的类是什么? 镜头2	45
谜题22: URL的愚弄	47
谜题23: 不劳无获	49
第 4 章 循环之谜	53
谜题24: 尽情享受每一个字节	53
谜题25: 无情的增量操作	55

谜题26: 在循环中	57
谜题27: 变幻莫测的i值	59
谜题28: 循环者	61
谜题29: 循环者的新娘	63
谜题30: 循环者的爱子	65
谜题31: 循环者的鬼魂	67
谜题32: 循环者的诅咒	69
谜题33: 循环者遇到了狼人	71
谜题34: 被计数击倒了	73
谜题35: 分分钟	75
第5章 异常之谜	77
谜题36: 优柔寡断	77
谜题37: 极端不可思议	79
谜题38: 不受欢迎的宾客	81
谜题39: 您好, 再见	83
谜题40: 不情愿的构造器	85
谜题41: 域和流	87
谜题42: 异常为循环而抛	89
谜题43: 异常地危险	93
谜题44: 删除类	97
谜题45: 令人疲惫不堪的测验	101
第6章 类之谜	105
谜题46: 令人混淆的构造器案例	105
谜题47: 啊呀! 狸猫变犬子	107
谜题48: 我所得到的都是静态的	109
谜题49: 比生命更大	111
谜题50: 不是你的类型	113
谜题51: 要点何在	115
谜题52: 总和的玩笑	119
谜题53: 做你的事吧	123
谜题54: Null与Void	125
谜题55: 特创论	127

第 7 章 库之谜	131
谜题56: 大问题	131
谜题57: 名字里有什么	133
谜题58: 产生它的散列码	137
谜题59: 差是什么	139
谜题60: 一行以毙之	141
谜题61: 日期游戏	143
谜题62: 名字游戏	145
谜题63: 更多同样的问题	147
谜题64: 按余数编组	149
谜题65: 疑似排序的惊人传奇	152
第 8 章 更多类之谜	157
谜题66: 一件私事	157
谜题67: 对字符串上瘾	161
谜题68: 灰色的阴影	163
谜题69: 黑色的渐隐	165
谜题70: 一揽子交易	167
谜题71: 进口税	169
谜题72: 终极危难	171
谜题73: 隐私在公开	173
谜题74: 同一性的危机	175
谜题75: 头还是尾?	177
名字重用的术语表	180
第 9 章 更多库之谜	183
谜题76: 乒乓	183
谜题77: 乱锁之妖	185
谜题78: 反射的污染	189
谜题79: 狗狗的幸福生活	193
谜题80: 更深层的反射	195
谜题81: 无法识别的字符化	197
谜题82: 啤酒爆炸	199
谜题83: 诵读困难者的一神论	201
谜题84: 戛然而止	203

谜题85: 惰性初始化	205
第 10 章 高级谜题	209
谜题86: 有害的括号垃圾	209
谜题87: 紧张的关系	211
谜题88: 原生类型的处理	213
谜题89: 泛型迷药	217
谜题90: 荒谬痛苦的超类	221
谜题91: 序列杀手	224
谜题92: 双绞线	229
谜题93: 类的战争	231
谜题94: 迷失在混乱中	233
谜题95: 来份甜点	237
附录A 陷阱和缺陷的目录	239
A.1 词汇问题	240
A.2 整数运算	241
A.3 浮点运算	243
A.4 表达式计算	244
A.5 控制流	245
A.6 类初始化	246
A.7 实例的创建与销毁	247
A.8 其他与类和实例相关的主题	248
A.9 名字重用	250
A.10 字符串	251
A.11 I/O	253
A.12 线程	253
A.13 反射	255
A.14 序列化	256
A.15 其他库	257
附录B 书中幻图的注释	259
索引	265
参考文献	281

绪论

本书充满了有关Java编程语言及其核心类库的谜题。任何具有Java应用知识的人都可以理解这些谜题，但是这其中的许多谜题即便是对最富经验的程序员而言，也是一种挑战。因此，如果你不能解决它们，千万不要感到沮丧。根据它们所使用的特性，我们对其进行了松散的分组。但是请不要以为解决某个谜题的技巧是与其所在章的题目相关的，我们保留误导你的权力。

多数谜题都利用了那些会导致bug的行为，这些行为要么违反直觉，要么晦涩难懂，它们被认为是陷阱（trap）、缺陷（pitfall）和冷僻案例（corner case）。每一种平台都有这些问题，但是与其他能力相当的平台相比，Java要少得多。本书的目标就是要用这些谜题寓教于乐，让你避免底层的这些陷阱和缺陷。等学习完这些谜题之后，受代码中这类危险折磨的可能性将会大大降低，而在复审或修订代码时发现它们的可能性将会大大提高。

阅读本书时，在旁边放一台计算机。为了最大限度地发挥这些谜题的效用，还需要一个Java开发环境，例如Sun的JDK[JDK-5.0]，它应该支持5.0版本，因为有些谜题依赖于在这个版本中才引入的特性。可以从www.javapuzzlers.com下载这些谜题的源代码。除非你是极能忍辱负重之人，否则我们建议你在解决谜题之前就去下载这些代码，这比你亲自录入要轻松得多。

大多数谜题都采用了短程序的形式来表示，这些程序看起来是要做某件事，但是实际上做的却是另外一件事。你的任务就是要指出这些程序要做什么。为了最大限度地发挥这些谜题的效用，我们推荐你采用下面的方式：

（1）研究程序并设法在不使用计算机的情况下预测它的行为。如果你看不出个所以然，那就继续看。

(2) 一旦你认为了解了程序要做什么,那么就运行它。它做了你认为它应该做的事了吗?如果没有,你能对所观察到的行为作出解释吗?

(3) 假设有问题,思考可以怎样改正该程序。

(4) 之后,且只能是之后,阅读解惑方案。

有些谜题要求你编写少量的代码。为了最大限度地发挥这些谜题的效用,我们推荐你尝试着在不使用(至少是暂时不使用)计算机的情况下去解决它们,然后在计算机上测试你的解惑方案。如果你的代码不起作用,那么在阅读解惑方案之前,好好琢磨,看看是否可以让它起作用。

与大多数谜题书籍不同,这本书交替介绍谜题和解惑方案。这使读者在阅读本书时不用在谜题和解惑方案之间前后翻找。本书的布局使读者在从谜题转向解惑方案时必须翻页,这样读者就不必担心自己还在努力解决谜题时,一不留神看到了解惑方案,泄漏了天机。

我们鼓励阅读每一个解惑方案,即使你已经成功地解决了谜题。解惑方案所包含的分析,已经远远超过了对程序行为的简单解释,其中还讨论了相关的陷阱和缺陷,并提供了如何避免陷入此类危险的经验教训。与大多数最优实践指南一样,这些教训都不是必须要遵守的规则,但是你应该极少违反它们,即使违反也要有合适的理由。

大多数解惑方案都引用了《Java语言规范第3版》[JLS] (*The Java™ Language Specification, Third Edition*) 的相关章节。这些引用对理解谜题并不是必需的,但是如果你想更深入地钻研谜题底层的语言规则,那么它们就很有用了。与此类似,许多解惑方案都包含了对 *Effective Java™ Programming Language Guide* [EJ] 一书中相关项的引用。如果你想更深入地钻研最优实践,那么这些引用就很有用了。

有些解惑方案还包含了对语言和API设计决策的讨论,这些设计决策导致了在谜题中所展示的危险。这些“对语言设计者的教训”只是思想的食粮,就像其他任何食粮一样,它们应该被有保留地接受。语言设计决策不能彼此孤立地做出,每一种语言都体现了以细微方式相互作用的成千上万的设计决策。一个设计决策对某一种语言是正确的,但是对另外一种语言可能就是错误的。

在这些谜题中的许多陷阱和缺陷都是可以更正的,它们可以通过静态分析(static analysis)被自动探测到,静态分析即在不运行程序的情况下分析程序。有一些很优秀的通过静态分析探测bug的工具可供大家使用,例如Bill Pugh和David Hovemeyer的FindBugs[Hovemeyer04]。某些编译器和IDE工具也可以完成bug探测,例如Jikes和

Eclipse[Jikes, Eclipse]。如果你正在使用的编译器就是其中之一，那么直到你尝试解决某个谜题时才编译该谜题，这一点特别重要：编译器的警告信息可能会泄露解惑方案。

本书的附录是Java平台中的陷阱和缺陷的目录，它提供了一个对谜题中所采用的奇异案例的简明分类法，并且引用了谜题和其他相关的资源。在完成解决谜题的任务之前，不要去看这个附录。先阅读附录会使谜题变得索然无味。应该在完成了所有的谜题之后，才开始浏览这些参考资料。

表达式之谜

在本章所描述的谜题都很简单，它们仅仅涉及表达式的计算。但是切记，不能仅仅因为它们简单，就认为它们很容易解决。

谜题1：奇数性

下面方法的目的是确定其惟一的参数是否为奇数。这个方法可行吗？

```
public static boolean isOdd(int i) {  
    return i % 2 == 1;  
}
```

