



BITTER EJB

中文版

(美) Bruce Tate
Mike Clark 著
Bob Lee
Patrick Linskey

于涵 林琪 等译



MANNING



机械工业出版社
China Machine Press



BITTER EJB

中文版

Bruce Tate
Mike Clark
(美) Bob Lee 著
Patrick Linskey

于涵 林琪 等译



机械工业出版社
China Machine Press

地址：北京市机械工业出版社
电话：(010) 28356294

畅销书作家Bruce Tate在辍笔十余年后于2002年又以畅销书《Bitter Java》再显神威,该书在Amazon上创下了难以打破的畅销记录。在这本书的鼓舞下,作者把讨论主题从初级Java移向了更具挑战性的领域,即Enterprise JavaBeans (EJB)。

本书完全承袭了《Bitter Java》的写作风格,讨论了当今大家争论得如火如荼的话题。作者找出并解释了常见的EJB陷阱,并把这些陷阱提升为“反模式”。这些反模式可以帮助人们避免一些最重要的EJB问题,从持久存储到性能领域都有所涉及。书中穿插的一个个冒险小故事,不仅妙趣横生,引人入胜;而且寓意深远,很好地隐喻了每个反模式主题。书中还讨论了何时应该使用EJB以及何时不应该使用,管理会话状态,实体bean的替代方法,性能调优技术,XDoclet、Ant和JUnit最佳实践,避免各类bean的陷阱等。

古人云:以铜为镜正衣冠;以人为镜明得失。从他人的错误中学习不仅很有意义,而且会有很高的学习效率。

Bruce Tate et al: Bitter EJB (ISBN 1-930110-95-2).

Original English edition published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, Connecticut 06830.

Copyright.© 2003 by Manning Publications Co.

All rights reserved.

Simplified Chinese translation edition published by China Machine Press.

Copyright © 2005 by China Machine Press.

本书中文简体字版由Manning出版公司授权机械工业出版社独家出版。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。

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:01-2005-3049

图书在版编目(CIP)数据

Bitter EJB中文版 / (美)塔特(Tate, B)等著;于涵等译. -北京:机械工业出版社, 2005.9

书名原文: Bitter EJB

ISBN 7-111-17249-3

I. B… II. ①塔… ②于… III. Java语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(2005)第102582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:周琦 刘立卿

北京诚信伟业印刷有限公司印刷·新华书店北京发行所发行

2005年9月第1版第1次印刷

787mm×1020mm 1/16·19印张

印数:0 001-4 000册

定价:38.00元

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

前言^①

我曾经向与我有多年交情的最要好的朋友Mike Oehrtman谈起《Bitter Java》中的冒险故事。实际上，书中起码一半以上的故事中都少不了他的身影。我问他有没有自己的故事可以拿出来，因为当时我们正在筹划另一本Bitter系列的书。他说可以贡献一些徒步旅行的故事。听了这话，我不禁大笑起来，并告诉他如果只是到处走走还称不上我所说的历险。Mike反驳说，“相信我，你会喜欢这个故事的。”

那时我们背着背包在阿拉斯加旅行。一天我们来到Denali国家公园，在与守园人的交谈中，他告诉我们必须对旅行有所计划。这个公园有很好的监视系统，所以即使一路上没有看到其他的游客，守园人也能知道我们身在何处，误差不会超过两英里。我也算是个旅行老手了，所以看地图是不成问题的。我对旅行做了安排，带上地图，乘上一辆公共汽车就出发了，开始了在偏僻地方的旅行。

这辆车上坐满了本地人，司机也是一个土生土长的阿拉斯加人。他的胡子可真长，甚至比坐在我旁边的Beth还有过之而无不及。在旅途中我们看到了数不胜数的野生动物：麋鹿、驼鹿，甚至还有一些野羊。只要你能想到的动物，我们都看到了。转过一个弯后，前面出现一条河，一头熊矗立在前方，这可不是去年我们在Smoky山脉见过的那种小棕熊。这是一头块头极大的灰熊。Beth拍了拍我的肩膀，示意我看车的另一边，我看到到处都是大块头的熊。我以前也不是没有见过大灰熊，但是一下子看到这么多确实是头一遭。车在穿过熊群的时候慢了下来。这样一来，我开始紧张了。我担心这些熊会不会向我们发动攻击，因为车厢里有的是食物。突然司机把车停了下来，我猜想是不是他也很害怕。不过，他只是坐着不动，这个时候我可真的开始冒汗了，怀疑是不是发动机出了故障？

最后，司机转过头来向Beth和我问道：“不下吗？”这时我才恍然大悟。原来我们要在这一站下车。不错，此时此刻，我们必须置身于那么多的熊中间。我真的从来没有这么胆战心惊过。

别人经常问我为什么能如此热衷于这样一种工作，在键盘前一坐就是十几个小时可谓是家常便饭。他们免不了将写作与污迹斑斑的键盘、满是灰尘的显示器和单调乏味的书桌关联起来。

我见过熊，而且见过许许多多的熊。如果你致力于Java开发，我想你也一定遇到过许多你眼中的“熊”。首当其冲的要算是经济了。我是从2000年开始的经济衰退之初着手写作的。我曾写过《Bitter Java》，并在这个领域一片混乱的时候开始从事顾问工作。作为Java行业根基的一些公司地位已经在动摇。Sun遇到了各种挑战，WebGain甚至销声匿迹。实力结构正在发生着变化。在写这本书时，Oracle买下了TopLink，这打破了对象持久存储市场的平衡局面。IBM则买下了

① 参加本书的翻译人员包括：于涵、林琪、张伶、林志秀、王宇、邹红霞、杨春生、阎慧、张建伟、刘亮、王志淋、刘跃邦、孙春娟、丁小峰。

Rational，这也极大地影响着它的发展。

我的客户、读者和学生们还应当了解到另一只“熊”，这就是.NET。不论是对于客户还是Microsoft产品，我认为.NET都是一只货真价实的熊。我需要理解.NET在企业环境中可以做什么，它是否大到可以“吃掉”J2EE？它的“攻击”是否会影响到我的生活？迄今为止这一切还没有发生。不过，这只熊的存在却是毋庸置疑的，它还在一路吞噬着，成长壮大着。PetStore基准测评（在第1章中会谈到的）使我切身感受到威胁的存在。一切都可能瞬息万变，我们只需要确保一点，那就是绝不能小看像Microsoft这样危险的竞争者。

另一个让我真切感受到的“熊”是EJB，这可真是一只又大又肥的“熊”。这个行业为EJB提供了充足的“营养”。IBM、Sun、Oracle、BEA还有其他众多厂商都在大力扶持EJB。EJB是有“牙齿”的：我们最终肯定会看到一些商业EJB应用将大规模增长。而且EJB的精力也很旺盛：应用服务器市场的人气仍然很旺。

我曾经多次远远地观望EJB，然后悄悄走开，担心这只“熊”会转向我，用它的利齿把我撕成碎片。EJB也确实可能会转过来面向你。EJB项目经常会失败，而且失败的频率着实让人震惊。这种失败的可能性也成为本书的一个谈资。如果你仔细研读过成功的模式和最佳实践，我想你可能算是一个不错的开发人员，但是如果你想更上一层楼，那么还必须了解技术的局限所在。勇敢地面对EJB，你一定会有所历练。这个过程会增长你的技艺，打开你的思路，但同时受伤也是在所难免的。幸运的是，还有一种更好的学习方法。《Bitter Java》使我确信，从他人的错误中学习不仅很有意义，而且会有很高的效率。

按前面故事的说法，在本书中，我和合作者们最后都下了车，真实地面对这些熊。并不是说对遇到的所有项目都推荐EJB，绝非如此。我们只是指出EJB有其一席之地，而且在适当的场合下这种“动物”可能会有超群的能力。所以，勇敢地下车吧，探索最大的熊所藏身的阴暗洞穴。我们要对EJB的危险和力量都了然于胸。

Bruce Tate

致 谢

对于我们每个人来说，本书都是一个要求极高的项目，同时也使我们深受启发。如果没有得到众人的大力支持，单凭我们几个平平常常的作者是难以完成这项工作的。我们的朋友、家庭、合作者、同事和出版商都伸出了援手，才使这本书的面世成为可能。我们还从一些从未谋面的人们那里收到了大量反馈，而其中许多人不曾希冀过任何回报。

逐个向大家致谢是不可能的，不过可以肯定的是：正是他们的贡献鞭策着我们尽力前行。

共同致谢

首先，要感谢卓越的Manning Publications出版公司人员。在我们迫切需要时，他们提供了真正的世界一流的编辑支持；而在我们希望潜心写作时，他们又能不加以任何干扰。先要感谢那些一直以来站在幕后推动发展的人们。感谢Helen Trimes从不懈怠的有效推动。尽管《Bitter Java》在Amazon上的畅销记录很难打破，但是他的努力使我们充满信心。还要感谢Mary Piergies，他总能在我们开口之前就洞察到我们想要什么。感谢Chris Hilman为这本书的网页所做的努力和所有那些繁杂的工作。尤其要感谢Lee Fitzpatrick使一切正常运转，感谢Susan Capparelle的有求必应。他们的贡献并不仅仅是编辑、排版、校对或做图。随着对出版业有更多的了解，我们就越发钦佩他们为每一页所倾注的心血。

接下来要感谢帮助我们完成这本书策划的出版团队。感谢Marjan Bace。作为编辑，他和我们可谓是一见如故，不仅成为我们的良师益友，还以他的远见卓识深深地影响着我们。是他的援手使这本书保持了Manning一贯的高质量，这也是我们引以为豪的。感谢Lori Piquet和Adrianne Harun编辑，尽管他们的编辑工作极为严格，但对我们却一直“温柔”相待，着实不易。还要感谢Tony Roberts和Leslie Haines，他们对如何出版精美图书可谓老练。人非圣贤，孰能无过，所以这本书得到了相当苛刻的审校。感谢Thomas Walkama出色的技术审校，还要感谢Elizabeth Martin的审验。Ted Kennedy领导了Manning的审校工作，在此表示感谢。另外对于夜以继日地为Manning审校本书的人一并致以谢意：John Crabtree、Jack Herrington、Adam Maass、Barry Nowak、Dimitri I. Rakitine和Jon Skeet。

还要感谢TheServerSide.com的Ed Roman和Floyd Marinescu，是他们慷慨地在首页上留出空间，提出了审阅这本书的号召。不出所料，我们收到了大量的反馈，所以在此要向热心参与的每一个人表示真诚的感谢：Jerome Abela、Charles Bear、Peter Bonney、Neelan Choksi、Joanne Christian、David Copeland、Taylor Cowan、Scott Dodson、Pierre Fauvel、Rob Hafernik、Don Hanson、Mike Hogan、Konstantin Ignatyev、Erik Jensen、Shailendra Kadre、Oliver Kamps、Rajesh Koilpillai（无论如何，感谢他们所做的努力）、Eugen Kuleshov、Darrell W. Rials、Seb Rose、Ravi Shankar、Curt Smith、Scott Stanchfield、James Strachan、Bernard Trigaux、

Christopher Webster、Matthias Weidmann、Mike Wertheim、Glen Wilcox、Dave Wiltz、David Ziegler和Dick Zetterberg。特别要感谢Matt Gibbs和BORN，他们提出了一些极有价值的反馈。

Mike Clark

最初我只是要写这本书中的一章。后来增加为两章。等我完成时，已经写出了三章，当时还很纳闷冬天怎么那么快就过去了。再回首，我发现这个过程中有许多东西难以言表。在写作过程中，我结识了很多新的朋友和同事，即使多年以后这本书不再出版，我仍希望与他们的友谊保持长存。

在此向Bruce Tate致以最诚挚的谢意，他是这本书的总设计师，从不知疲倦，是他使我有机会圆了这个一生的梦想；还要感谢Thomas Walkama，他审阅了我写的所有内容，通过他的修改让这一部分增色不少，他后来还承担了整本书的技术审校工作；另外感谢Jason Hunter，作为一个值得依赖的良师益友，他不仅以温和的态度纠正了我对会话状态管理的错误认识，还使我的照相技术有所提高；感谢James Duncan Davidson的教导，由此我才能够尝试写作，也是他鼓励我把眼界放远一些；感谢Tyler Jewell无私地贡献了有关有状态会话“野兽”的想法，还慷慨地提供了许多资料；感谢James Strachan在工作的同时，还不辞劳苦地审阅了消息一章的每一页书稿，并做出了卓越的贡献；感谢Glen Wilcox在一次次的午餐时能够忍受我的闲扯；还要感谢Steve Walker在Nebraska参加训练时还能提出有关抖动调优的真知灼见。

特别要感谢我的妈妈和爸爸，是他们的爱和一直以来的教导才使我完成了这项艰巨的工作。原先我还没有清楚地认识到这一点，但是所有这一切确实深深地影响着我。

我把这本书献给Nicole，我的爱人，她总能保证我有充足的时间写作，也总能出现在阳台上为我鼓劲加油。她鼓励我一直写下去，并在适当的时间提醒我该去滑雪了。感谢她无私的爱、永不停止的支持，以及在这马拉松式的过程中无私的奉献。是她带给我最大的快乐。

Bob Lee

感谢Bruce给了我这样一个让人无法相信的机会，使我参与写出了有史以来的第一个作品，他对我的帮助无法度量。还要感谢Mike和Patrick完成了一些颇有争议的主题，使这个项目得以顺利进行。这是一条很长的充满艰辛的道路，不过我可以自信地说：我们确实在极短的时间内探访了许多以前从未涉足的领域。书中凝聚着大家智慧的结晶，这对于企业开发人员（不局限于EJB领域）会大有益处。

在写这本书时，我挖出了在职业生涯中所遭遇的许多失败，诚然，这些例子不能算很贴切，但它们确实成为了这本书的主要基础。在回顾失败的过程中，我也回想起了一些好的方面，特别是那些帮助我摆脱困境的人们（按字母顺序）：Scott Antle、Rick Bayers、Paul Brown、Tim Burns、Dr. Hilton Chen、Bob Cringely、Dan Gluck、Jay Goff、Lori Harvey、Dr. Jim Hayes、Karthik Krishnamoorthy、Pat Niemeyer、Jim Owens、Ed Rich、Tim Saunders、Ken Shealy、Mark Volkmann、Christina Ware、Debbie Wiler、Tim Williams和Jim Zhou。

特别要向我的家人表示感谢。我的父亲Rick在我很小的时候教我把数学看成是一个游戏，

而且从不介意我摆弄（甚至弄坏）他的计算机。我的母亲Nan教我写作，她也是我的自信和动力的源泉。妈妈，你永远在我心中。还要感谢我的Sigma Chi老友们和兄弟Tim，他们都特别相信我的工作能力，无论我身在何处，只要想到我身边这些强劲的臂膀，我就倍感温馨。

Patrick Linskey

我要感谢Steve Kim、Marc Prud' hommeaux和Abe White的帮助和卓越见识。如果没有他们的支持，无论是对EJB规范还是一般意义上的企业开发，我的理解和认识都是远远不够的。还要向Bruce Tate致以真诚的谢意，是他把我纳入本书写作团队，也是他在我写这本书的过程中一直支持、启发和指导我。

如果没有SolarMetric所有人的支持，我也无法完成这本书，我写作的那个时期甚至把日常工作都丢到一边，感谢他们忍受这一切。

Greg Campbell、Eric Lindauer、David Karger和Pat Thomas都以他们各自的方式使我认识到：一件事只要值得去做，就要做好、做对、做完。

最后，要感谢我的父母教我把写作当作一件有趣的事情，还要感谢我的姐姐为这本书提供了许多素材。

Bruce Tate

《Bitter Java》对我来说确实是一件乐事，但这本书则不然，至少最初并非如此。感谢Bob Lee帮助我渡过了最初的阶段，并使组建写作团队的过程不至于太过难熬。我们坚守着他的座右铭：一直写下去。

Patrick Linskey和Mike Clark的加入才使我松了一口气，直至这时这本书才变得有趣起来。我要感谢Patrick，在没有任何写作经验的情况下，他勇敢地承担了3个关键章节的编写任务。他一直都很随和，也很耐心，他的卓越见识对于这个项目是至关重要的。特别要感谢喜欢Frodo Baggins的Mike，他从来没有真正了解前路有多险恶：希望没有把他引入歧途。不过，如果没有他，就没有这本书的诞生。他所付出的时间和精力、他认真的工作态度、他的幽默感，特别是他的为人，所有这些都让我非常钦佩。

感谢Marjan使我有机会做些不同的事情。他对我的第二职业产生了深远的影响。这又是一次成功的合作。感谢Ed Roman和Floyd Marinescu的友谊，还要感谢TheServerSide.com，后来证实这是我们得到反馈的源泉，而且这本书之所以产生很大反响，TheServerSide.com也功不可没。要感谢Jay Zimmerman引我上路，使我有机会完成Bitter系列的书为大家答疑解惑。

另外要感谢Mike Oehrtman提供的那些疯狂的冒险经历，让我不禁为他屏气凝神。我知道他的母亲一直在关注，可能还在为每个新历险忧心忡忡。感谢他公开这些故事。

最后，感谢Maggie无尽的支持、爱和友谊。嫁给一个作家，生活往往不太容易，不过她总是能那么优雅、高贵和可爱，而且我的同事们在她那里总是能得到优厚的款待。有她在我身边，我就很愉快，对她的爱地久天长。

关于本书

为什么以EJB为主角

辍笔十余年之后，Bruce Tate于2002年再度出山，并以畅销书《Bitter Java》重振声威。在《Bitter Java》中，他提出了反模式（antipattern）的概念。按照他的定义，反模式就是软件开发人员经常遭遇的常见程序设计问题。Bruce的目标是针对基本的Java程序设计问题，并希望Java开发人员能够将反模式作为一个重要内容建立相关的概念。尽管《Bitter Java》中对反模式做了充分的讲解，但是这本书的范围仅限于初级Java，考虑到这一点，Bruce马上做出决定：把讨论的主题移向更具挑战性的领域。Enterprise JavaBeans（EJB）正好能够满足要求，在这种技术环境中更容易发现反模式。可以考虑以下因素：

- 尽管EJB能力超群，但是EJB群体一直以来总是在为某个问题争论不休。存在争议的框架势必成为反模式滋生的肥沃土壤。
- Microsoft的.NET作为Java的主要竞争者（推而广之也是EJB的主要竞争对手），正在日益展现其魅力，二者的“争战”可以提供饶有趣味的素材。
- 尽管J2EE在市场上大获成功，但是绝非没有隐患。
- 特别是，J2EE的持久存储框架尽管有所改进，但是仍危机四伏，面对商务实现的重压，往往不堪重负。如果框架本身都存在问题，那么就难免会导致误用。
- 新的J2EE框架发布的速度过于迅猛，这就使我们对其成功喜忧参半。新的工具和技术过多，开发人员误用某种工具（技术）的情况肯定是司空见惯的。

简单地说，EJB在实用中是成功的，而其存在争议也是不可否认的，鉴于这种情况，完全可以将EJB作为主角编写一本不错的Bitter书。在本书中，我们将讨论开发人员采用EJB时遇到的反模式，换句话说，就是开发人员经常落入的陷阱。在此将涉及EJB的诸多不同方面，从事务到持久存储再到消息都有所涵盖。我们还会谈到另外一些重要的主题，如性能和测试等等。

概括地说，本书的写作团队希望奉上两样新品供读者“品尝”：

1) 向读者提供全新的视角。大多数EJB书都会大谈只要采用了某种设计模式或开发方法，使用EJB就会何等容易。我们则反其道而行之，在此会不加掩饰地承认EJB确实很难，如果你不能了解困难所在，很可能会犯大错误。

2) 向读者提供全新的见解。例如，到目前为止，我们确实发现了有些书很好地讨论了如何选择一种会话状态管理方法。但是我们并不认为哪种方法能够通盘适用，也不相信一种方法就能满足你的全部需求。恰恰相反，在此会提供多种实现消息的方法，相应地也引入了许多新的消息反模式。我们还提供了对多种持久存储方法的见解，这在以前的EJB书中是从未有过的。这本书中还谈到了性能调优，如果未能采取适当的方法，这个问题对于EJB开发人员可能相当困难。

这本书不是要抨击EJB有多么不好，在此只是指出如何最大限度地利用EJB框架。这并不说明我们对EJB的每一部分都表示支持和赞同。实际上，读者会从书中了解到，我们对实体bean的评价很低，因为对于解决持久存储问题来说，我们认为还存在更好、更实用的方法。另外，支持EJB也不表示建议人们无论何时何地都使用EJB，以此向别人展示自己的能耐。（Mike把这种做法称为“炫耀式设计”。）读者会看到，我们只是把EJB放到了一个相当受限但很重要的“箱子”中，对EJB的支持仅限于这个箱子范围。读者将了解如何有效地应用EJB去构建分布式、事务性、可扩展的系统来解决实际问题。

本书风格

在现今的计算机图书中可以看到两种趋势。一种趋势是在许多书中提供了大量内容。尽管有时需要探究主题的各个方面并提供详尽的描述，但是如果每一本书都落入这个俗套就未免适得其反了。有些作者不得不大幅扩充内容，以至于过于雕琢细节。这样带来的后果就是，你可能不会再有耐心一页一页地阅读计算机图书了。我们保证不会落入这个陷阱。在此只是要告诉读者希望了解的内容，然后就会知趣地“闭嘴”。

有些出版商对作者补充的非技术性内容（如个人情况）一概不予保留。这当然可以使书免受不公正的攻击或批评，但如此一来，也会导致阅读这些书可能味如嚼蜡。我们尽力让本书更有趣味，但其中并没有充斥作者们无聊的生活琐事。每一章都以一个短小的冒险故事作为开篇。这些冒险故事不仅很有意思，而且通常可以很好地隐喻反模式主题。这些真实的故事都是从作者自己或一些朋友那里收集来的。许多《Bitter Java》的读者都反映说那本书读起来很有意思，其中的故事使它摆脱了许多计算机图书的单调和带给人们的沉闷感。我们会让这些“调味料”尽可能简短，如果读者愿意完全也可以把它们跳过去。请告诉我们你对此的看法。

虽然各章是由不同作者单独撰写的，但是为了避免不一致，我们选出一个人对全书进行了统稿。可以把这本书的见解认为是整个写作团队中所有作者共同经验的体现。我们自己则认为，与其他由多名作者合作产生的书相比，这本书更具有统一性。如果读者在书中读到“我”这样的字眼，例如在小故事中出现的“我”，这里所指的就是这一章的作者。

作为本书的作者，我们的观念一致，都认为自己所做的工作是传道、授业、解惑。在此不想用一些专业术语让读者望而却步，而会用浅显的语言和亲切的第二人称表述，而且句子不会太长，这样读者就能更多地注意内容本身而不是费力地理解词汇。如果用言语不能很好地表达，我们还会辅以图表说明，并尽力始终保证它的趣味性。由于EJB的例子往往都很长，在需要代码来说明的时候，我们只是抽取其中重要的部分在书中给出，其他的代码都可以从www.manning.com/tate2下载。还可以在这个网站上访问Manning出版公司的作者在线论坛（Author Online），通过论坛你可以直接与我们交流。

写作团队

要分析EJB的问题，我们所需要的不仅仅是程序员，而是能够找出反模式并能切中其要害的专家。我们需要的作者不仅要有很高的造诣，还要能够接受《Bitter Java》的写作风格。本书的

写作团队包括:

Bruce Tate, 他是畅销书《Bitter Java》的作者, 这也是Manning出版公司的第一本Bitter系列书。Bruce的顾问生涯长达15年。他在IBM得到了第一份工作, 在那里任职的10年中, 他主要从事数据库和工具开发。在IBM, 他撰写了最初的两本书, 并申请了8项发明专利。离开IBM后, 他在Austin启动并领导了Contextual的Client Services项目。目前他在Texas的Austin工作, 是一个独立顾问, 同时还与Middleware公司和其他客户一道向其客户推广和讲授有效Java设计。

Mike Clark, 他是Clarkware Consulting公司 (www.clarkware.com) 的总裁, 该公司位于科罗拉多州的丹佛市。自1992年以来, 他一直致力于软件开发, 从1997年后, 开始对Java情有独钟。第一次落入EJB陷阱是在1998年, 当时他在开发一个定制EJB容器, 而此时商业J2EE服务器尚未出现。他开发了许多开源工具, 其中包括JUnitPerf, 这是一个用于持续性能测试的JUnit扩展包。另外, Mike会定期把他的遭遇说出来或写出来, 以帮助开发团队更快地构建更好的软件。

Bob Lee, 他在密苏里州的圣路易附近工作, 作为一个独立顾问, 他同时也是一位开源软件的开发者, 并有十多年的软件开发经验。Bob维护了一个Java主题网站 (www.crazybob.org), 欢迎大家访问, 并与Bob一同在艰辛之旅上跋涉。

Patrick Linskey, 他是一个从事Java持久存储业务的公司——SolarMetric的工程副总裁, 该公司位于华盛顿。最近两年他主要在讲授有关Java持久存储的问题, 并致力于构建不同的持久存储方法。作为一个深受欢迎和有很高造诣的演讲人, 他在EJB应用开发和产品开发方面都有很丰富的经验。

本书结构

在本书中, 我们先从简单的基础知识入手, 然后深入介绍核心会话和消息API, 接下来对持久存储展开生动的讨论。本书的最后是有关构建系统和性能调优的点睛之笔。本书分为4部分:

第一部分: 基础知识

第1章 取舍之苦。在这一章中, 我们将简要地介绍本书, 并讨论是否应当使用EJB。在此还会谈到有关EJB的争论, 以及一直以来它对J2EE的发展有何影响。这一章的作者是Bruce Tate。

第2章 代价之苦。第2章将介绍开发EJB所固有的代价。在此会对前面的话题做更为深入的讨论。特别是, 我们会重申为什么不能轻率地选择EJB框架。最后, 这里会第一次审视EJB实体bean, 并指出EJB实体bean作为细粒度服务, 把它包装在粗粒度包装器中就会带来高昂的代价。这一章的作者是Patrick Linskey和Bruce Tate。

第3章 接口之苦。在构建接口时存在一些挡道的陷阱, 这正是本章讨论的内容。这一章中介绍的反模式将作为本书全书的基础。这一章的作者是Bob Lee。

第二部分: 会话和消息

第4章 会话之苦。这一章介绍了无状态会话bean, 这也是EJB体系结构中最活跃的地方。我们将提示如何对无状态会话bean善加利用, 另外还会谈及多种相关的反模式。这一章的作者是Bob Lee。

第5章 会话状态之苦。在这一章中, 讨论的主题将转向有状态会话bean, 这也是最容易遭到误解的分布式通信bean。我们将花大量篇幅分析在什么情况下适用有状态会话bean, 在分析

过程中还将谈到一些有关的反模式。这一章的作者是Mike Clark。

第6章 消息之苦。在第6章中，我们给出了消息驱动bean的定义。由于这种EJB相对较新，在此所述的一些反模式以前还未见于文献。这一章的作者是Mike Clark。

第三部分：EJB持久存储

第7章 实体之苦。我们肯定不是EJB实体bean的狂热支持者，尽管如此，在此仍会深入分析有关实体bean的一些反模式，在你决定使用实体bean时很有可能会落入这样一些陷阱中。我们还会讨论针对这些问题有什么解决之策。这一章的作者是Patrick Linskey。

第8章 选择之苦。这一章与书中其他章节可谓大相径庭。在此我们不建议使用EJB实体bean，而是讨论了另外两种候选方法：基于关系数据库的对象持久存储框架（包括OR映射层和JDO）和带会话外观（session façade）的原来的普通Java对象（plain old Java objects, POJO）（译者注：也有人译纯Java对象）。这一章的作者是Patrick Linskey和Bruce Tate。

第四部分：更宽泛的话题

第9章 调优之苦。在使用EJB时，必须对应用进行性能调优。在此会讨论一些可能适得其反的调优技术。在这一章的最后，我们将把话题转向如何补救，这里会谈到自动测试。这一章的作者是Mike Clark。

第10章 构建之苦。在第10章中，我们分析了对于诸如EJB这样的复杂框架，其日常管理会如何渐渐失控。然后读者会了解到一些补救措施，这些手段很适用于人们的日常构建，这包括利用XDoclet进行代码生成以及利用JUnit进行单元测试。这一章的作者是Bob Lee。

第11章 苦乐参半的未来。前面一直在大谈EJB之“苦”，现在要放松一下了。在此将展示EJB的能力，还勾画出了将来会怎样。这一章的作者是Bruce Tate，并参考了整个写作团队的反馈。

附录A Bitter传说。许多人都可能是初次接触反模式，所以我们决定把《Bitter Java》的第1章纳为本书的附录，其中对反模式做了很好的介绍。

附录B Bitter基础。由于有些读者对EJB没有任何经验，所以在这里对EJB做了一个简要介绍，以便读者从容上路。

参考文献。对于本书中提到的所有书或文章，在此都提供了相关的书目信息，此外还包括另外一些有用的书。

在各章的最后，我们都使用了《Antipatterns: Refactoring Software, Architectures, and Projects in Crisis》一书中建立的模板对反模式做了很有价值的评论。如果读者希望更多地了解这种反模式模板，可以参考该书。

本书读者

本书比《Bitter Java》更进一步，读者的Java水平必须达到中级或高级程度。如果有使用EJB的经验当然最好不过了，不过这不作为一个严格的要求。读者无需对反模式有所了解，我们会帮读者建立有关的基础。

编码约定

如果书中用到了代码，而且出现在一个段落中，代码行将以Courier字体显示，如this()。如

果需要使用更长的代码段，也会用Courier字体显示，不过会分成单独的代码块，并做出相应标注（译者注：这里标注的作用就相当于注释，但是这本书未采用Java代码注释的语法），如下所示：

```
if(we.need(code())) {
    code.show(likeThis);    代码标注
}
```

在此命名采用了camelCaseLikeThis的形式，类名的第一个字母大写，实例变量和方法名的第一个单词则小写。所有代码块都加了可选的大括号。一般来说，我们会尽可能使用Sun的编码约定。

如果书中没有包括问题代码（“痛苦”的来源），本书又会是另一副样子。所以我们希望向读者展示做某些事情会有哪些不当的方法，为此势必要提供不好的代码。不过，为了与正确的代码相区别，我们希望有所警示。正常的代码如下所示：

```
goodCode(looksLikeThis);
```

而问题代码则如下所示：

```
|bitterCode(looksLikeThis);
```

前面的竖线就是在警告你这是不好的代码，让你擦亮眼睛，分清好坏。

我们会信守简洁为美的原则。希望读者能将这本书作为一杯醇香的咖啡细细品尝。

封面插图

本书封面上是一个身着洗浴装扮的土耳其妇女。这个插图选自1799年Madrid首次出版的西班牙民族服饰集锦。这本书的首页上写着：

*Coleccion general de los Trages que usan actualmente todas las Naciones del
Mundo desubierto, dibujados y grabados con la mayor exactitud por R.M.V.A.R.*

Obra muy util y en special para los que tienen la del viajero universal

我们尽力遵从原意，将其直译为：

对于现今世界中多个国家的多种不同服饰，R.M.V.A.R设计并印制了一些惟妙惟肖的图片。这对于周游世界的旅行者来说尤其有用。

这些图片完全是手工绘制的，尽管我们对这些图片的设计者、刻版者和着色者一无所知，但是从图片本身可以清楚地看到他们的工作确实是“惟妙惟肖”。“身着洗浴装扮的土耳其妇女”只是所收录的众多精美图片中的一幅。集锦中收录的各种各样的图片生动地反映了200年前世界各地服饰的鲜明特色。在那个时期，即使两地仅相距数十英里，两地的服饰也可能是迥异的，单从人们的服饰就可以清楚地区分出他们是哪里人。这些图片可以活灵活现地反映出当时存在的隔离感和距离感，其实除了当今这个剧烈动荡的时期外，历史上所有阶段都存在着这种距离感。

从那以后，服饰语言已经发生了很大变化。原先各个地域都有多种多样的服饰，后来这种情况也在逐步消失。现在即使是不同大洲的居民，通常也很难加以区别。如果从好的一方面看，我们也许是在用文化的多样性来换取个人更为多样化的人生，也可以理解为换取一种更加多姿多彩、更有意思的才智人生。

图片中描绘的情景栩栩如生，这种封面不仅让人回首过去，而且我们对Manning出版公司在计算机图书中采用两个世纪前的各色地域风情作为封面的首创性和娱乐性也很满意。

目 录

前言
致谢
关于本书

第一部分 基础知识

第1章 取舍之苦	2
1.1 论战风暴	2
1.2 EJB反模式的历史	3
1.2.1 1998年3月: EJB 1.0	4
1.2.2 1999年11月: EJB 1.1	4
1.2.3 2001年8月: EJB 2.0	5
1.3 案例研究: PetStore基准测评	6
1.4 反模式: 金榔头	8
1.4.1 不明智的选择	8
1.4.2 解决方案: 仔细评估	10
1.5 小结: 前瞻	12
1.6 本章中的反模式	13
第2章 代价之苦	14
2.1 层出不穷的争论	14
2.1.1 EJB的价值	15
2.1.2 适用EJB的应用	15
2.1.3 使用试金石	16
2.1.4 通过测试	17
2.1.5 衡量复杂性	17
2.1.6 估计你的能力	18
2.2 反模式: 大锤打苍蝇	19
2.2.1 增加复杂性	21
2.2.2 解决方案: 简化	22
2.2.3 考虑管理多个文件的代价	24
2.2.4 更细的分析	26
2.3 实体bean应另当别论	28
2.4 实体bean: 两大进步	30
2.4.1 本地接口	30
2.4.2 容器托管关系	31
2.5 实体bean的深入探讨	31
2.6 小结	34
2.7 本章中的反模式	34
第3章 接口之苦	36
3.1 构建一个好接口	36
3.1.1 远程调用性能的下降	37
3.1.2 传引用与传值	37
3.2 设计应用层	39
3.2.1 再看实体bean	40
3.2.2 质疑EJB本地接口	40
3.3 反模式: 本地和远程接口并存	41
3.3.1 混合的接口会扰乱异常管理	42
3.3.2 混合的接口会影响性能	43
3.3.3 小反模式: 到处分布	43
3.3.4 小反模式: 透明分布	43
3.3.5 解决方案: 取得平衡	44
3.3.6 了解如何分布	45
3.4 反模式: 闯入厨房的顾客	45
3.4.1 把用餐者推至门外	46
3.4.2 解决方案: 由服务员疏导顾客	48
3.4.3 使用数据传输对象	49
3.5 反模式: 定制DTO	50
3.6 小结	53
3.7 本章中的反模式	54
 第二部分 会话和消息	
第4章 会话之苦	58
4.1 线程和同步	59
4.1.1 反模式: 纠缠不清的线程	59
4.1.2 解决方案: 标准化	60
4.1.3 处理挂起的线程	61
4.1.4 探寻解决方案	62

4.2 处理异常	64	5.5.2 使用servlet在服务器上存储会话	102
4.2.1 小反模式: 异常实现中存在逻辑	64	5.5.3 使用有状态EJB在服务器上存储会话	103
4.2.2 解决方案: 将逻辑重构移出异常	65	5.5.4 在数据库中存储会话	104
4.2.3 反模式: 吞掉异常	65	5.5.5 再谈购物车	105
4.2.4 解决方案: 一种简单的异常处理策略	66	5.5.6 总体解决方案: 选择适当的工具	106
4.2.5 反模式: 过分受宠的系统异常	67	5.6 小反模式: 将有状态会话bean用作共享数据缓存	107
4.2.6 解决方案: 抛出正确的异常类型	68	5.7 反模式: 会话大杂烩	107
4.3 重复大数据集	69	5.8 小反模式: 会话抖动	109
4.3.1 反模式: 过分耗用数据库连接	71	5.9 小反模式: 腐烂的会话垃圾	110
4.3.2 解决方案: 精确地管理连接	71	5.10 小结: 驯兽	110
4.3.3 反模式: 过于热心的迭代器	72	5.11 本章中的反模式	111
4.3.4 解决方案: 测试, 测试, 再测试	74	第6章 消息之苦	114
4.3.5 缓存结果	74	6.1 JMS简介	114
4.3.6 导出结果	75	6.2 一个早期反模式: 胖消息	115
4.3.7 确定结果的大小	76	6.2.1 没有万能钥匙	117
4.3.8 迭代处理重排数据	76	6.2.2 解决方案1: 让消息节食	117
4.4 高效地互操作	77	6.2.3 解决方案2: 使用引用	118
4.4.1 使用IIOP	78	6.3 小反模式: 瘦消息	119
4.4.2 反模式: 过窄的servlet桥	78	6.4 一个订单处理系统雏形	119
4.4.3 解决方案: 通用servlet桥	79	6.4.1 定义系统	120
4.5 小结	85	6.4.2 设计消息	120
4.6 本章中的反模式	85	6.4.3 选择消息模型	121
第5章 会话状态之苦	89	6.4.4 应对修改	122
5.1 谈谈会话状态	90	6.4.5 构建OrderRequest生产者	123
5.2 核心反模式: 会话包袱	91	6.5 反模式: XML作为银弹	124
5.2.1 状态之累	91	6.6 反模式: 消息拥堵	126
5.2.2 减负	92	6.6.1 持久性的代价	126
5.2.3 解决方案1: 转向无状态	93	6.6.2 持久订购的代价	127
5.2.4 解决方案2: 在适当的时候充分利用会话状态	93	6.6.3 解决方案: 只保存重要的消息	127
5.3 用有状态会话bean管理会话	94	6.7 小反模式: 请求立即应答	128
5.3.1 使用有状态会话bean购物	95	6.8 使用消息驱动bean	129
5.3.2 掀起盖头来	96	6.8.1 建立MDB池	130
5.4 用servlet管理会话	98	6.8.2 构建OrderRequest消费者	130
5.4.1 利用HttpSession力求简单	98	6.9 反模式: 自成一体的消费者	132
5.4.2 使用servlet购物	99	6.9.1 听听测试的意见	133
5.4.3 扩展servlet会话	100	6.9.2 解决方案: 委托给模块化组件	133
5.5 反模式: 会话状态金榔头	101	6.10 反模式: 烫手的山芋	135
5.5.1 在客户上存储会话	102		

6.11 反模式: 慢吞吞的食客	137
6.12 反模式: 监听	138
6.12.1 解决方案: 使用消息选择器	138
6.12.2 声明消息选择器	139
6.12.3 扩展消息选择器	140
6.13 反模式: 事后才考虑性能	140
6.14 小结: 获得消息	142
6.15 本章中的反模式	142

第三部分 EJB持久存储

第7章 实体之苦	150
7.1 理解实体bean反模式	150
7.2 反模式: 裸露的表面	151
7.2.1 网络往返通信会堵塞应用	151
7.2.2 失去事务完整性	152
7.2.3 解决方案: 会话外观	154
7.2.4 使用外观保证事务完整性	156
7.2.5 使用本地接口	156
7.3 反模式: 一口吞下个大面包	156
7.3.1 在BMP和CMP间做出选择	157
7.3.2 解决方案: 尽可能选择CMP	158
7.4 反模式: 应用连接	160
7.4.1 解决方案: 把连接委托给数据库	160
7.4.2 应用连接的常见例子	160
7.5 反模式: 应用过滤器	161
7.6 反模式: 笨重的键	163
7.7 反模式: 旋转门	165
7.7.1 解决方案1: 重构以避免再入	166
7.7.2 解决方案2: 禁用容器的再入检查	167
7.7.3 解决方案3: 游说EJB规范开发小组	167
7.8 小结	167
7.9 本章中的反模式	168
第8章 选择之苦	171
8.1 理解实体bean的替代选择	171
8.2 使用EJB持久存储	173
8.2.1 实现CMP	174
8.2.2 增加DTO和外观	177
8.2.3 包括部署细节	180

8.2.4 用BMP实体bean试试看	182
8.3 用JDBC简化	183
8.3.1 实现一个简单的JDBC模型	184
8.3.2 实现JDBC外观	185
8.3.3 为JDBC部署一个会话外观	187
8.4 使用对象持久存储框架	189
8.4.1 衡量对象持久存储环境	189
8.4.2 理解JDO	190
8.4.3 用JDO实现一个简单的模型	191
8.4.4 实现JDO模型的外观	192
8.4.5 完成方案的部署	193
8.4.6 比较各种选择	195
8.5 反模式: 持久存储问题	195
8.5.1 通用实体bean缺点	196
8.5.2 继承和多态	196
8.5.3 查询语言灵活性通常很重要	198
8.5.4 对于本地实体DTO需要非EJB方案	199
8.5.5 容器范围内的持久存储	200
8.6 解决方案: 不要“继承”持久存储体系结构, 而应适当选择	200
8.7 小结	201
8.8 本章中的反模式	201

第四部分 更广泛的话题

第9章 调优之苦	204
9.1 对成功的度量	205
9.1.1 响应时间	205
9.1.2 吞吐量	206
9.2 反模式: 过早的优化	207
9.2.1 盲目对EJB应用调优	207
9.2.2 解决方案1: 计划, 但(先)别采取行动	208
9.2.3 解决方案2: 编写良构的模块化代码	209
9.3 反模式: 事后才考虑性能	210
9.4 调优点滴	212
9.4.1 让EJB接受测试	212
9.4.2 通过测试	213
9.4.3 指定响应时间作为成功的度量	214

9.4.4 看到调优隧道尽头的一丝光亮	214
9.5 反模式: 抖动调优	215
9.6 小反模式: 手工性能测试	217
9.7 利用JUnitPerf完成自动化性能测试	218
9.7.1 JUnitPerf概览	218
9.7.2 测试响应时间	219
9.7.3 调整代码	220
9.7.4 指定可扩展性作为成功的一个度量	220
9.7.5 测试负载条件下的响应时间	221
9.7.6 使用连接池来提高吞吐量	222
9.7.7 测试吞吐量	224
9.8 性能建模	224
9.9 小反模式: 怯场	226
9.10 小结: 自信地调优	226
9.11 本章中的反模式	227
第10章 构建之苦	231
10.1 用带子绑个大包	232
10.1.1 理解一个示例EJB	232
10.1.2 组织目录结构	234
10.1.3 填写EJB JAR	235
10.1.4 加载类	235
10.2 反模式: 系统加载的应用类	236

10.3 反模式: EJB代码重复	237
10.3.1 解决方案: 自动生成EJB类	237
10.3.2 解决方案: 自动生成清单	239
10.3.3 解决方案: 自动生成EAR描述文件	240
10.4 反模式: 构建权威	241
10.5 反模式: 带着剪刀跑	242
10.6 反模式: 集成地狱	244
10.7 小结	245
10.8 本章中的反模式	245
第11章 苦乐参半的未来	248
11.1 历史定位	248
11.2 下一步规划	250
11.2.1 预计未来	250
11.2.2 修整持久存储	251
11.2.3 修整部署策略	252
11.2.4 考虑经济因素	252
11.3 反模式和下一步的动作	253

附录

附录A Bitter传说	255
附录B Bitter基础	268
参考文献	286