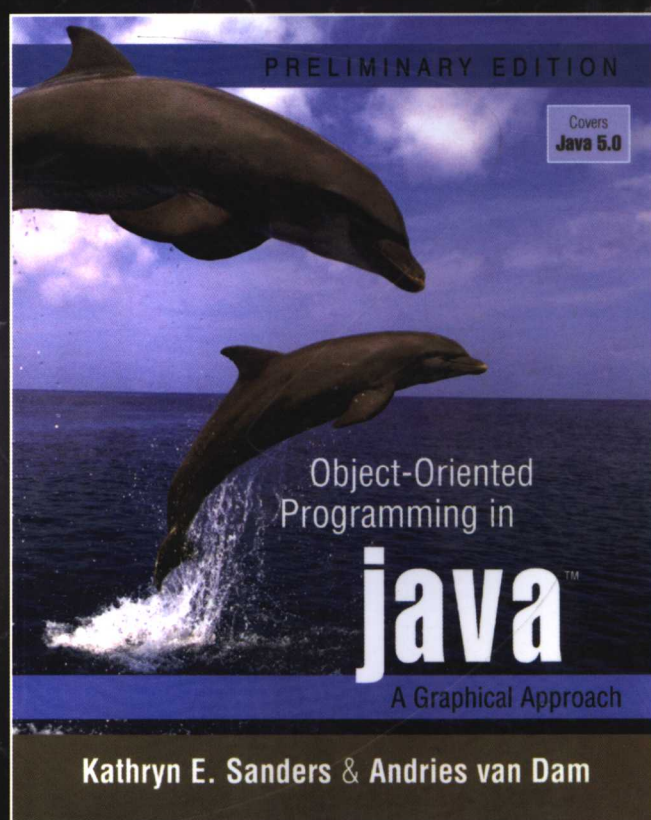


Java面向对象程序设计 图形化方法

(美) Kathryn E. Sanders Andries van Dam 著 李强 等译



Object-Oriented Programming in Java
A Graphical Approach



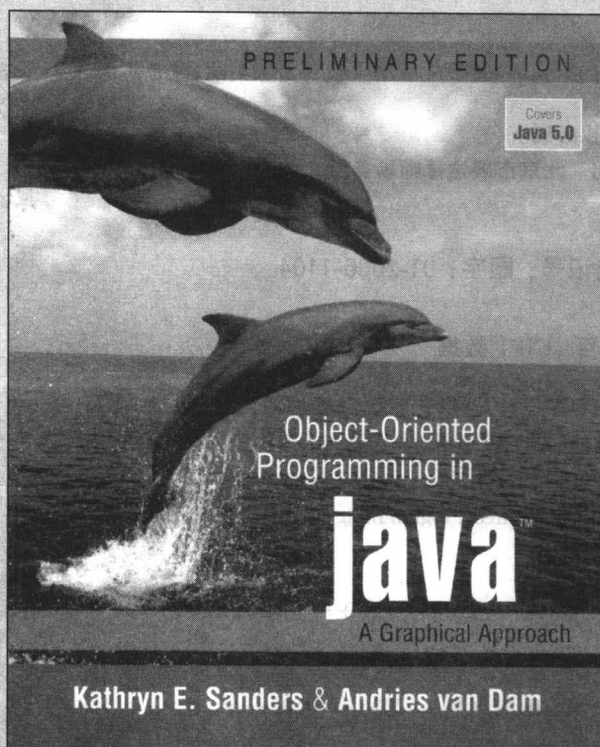
机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

Java面向对象程序设计 图形化方法

TP312
2029D

(美) Kathryn E. Sanders Andries van Dam 著 李强 等译



Object-Oriented Programming in Java
A Graphical Approach



机械工业出版社
China Machine Press

本书介绍 Java 及其标准 Swing 图形库的面向对象编程方法,强调了面向对象设计和编程的原则。本书在开始部分就介绍了所有的面向对象编程方法,从封装、继承到接口以及多态。本书使用很多可执行的实例讲解模块化,以及其他良好的编程习惯。书中设计了很多程序实例和练习题,利用交互图形可视化的方法来激发学生的学习兴趣。

本书适合作为高等院校相关专业的程序课程教材或参考书,也可供软件开发人员参考。

Simplified Chinese edition copyright © 2006 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Object-Oriented Programming in Java: A Graphical Approach* (ISBN 0-321-24574-1) by Kathryn E. Sanders, Andries van Dam, Copyright © 2006.

All rights reserved.

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

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:01-2006-1104

图书在版编目(CIP)数据

Java 面向对象程序设计:图形化方法/(美)桑德斯(Sanders, K. E.), (美)达姆(Dam, A. D.) 著;李强等译. - 北京:机械工业出版社, 2006.7

(计算机科学丛书)

书名原文:Object-Oriented Programming in Java A Graphical Approach

ISBN 7-111-19161-7

I. 面… II. ①桑… ②达… ③王… ④新… ⑤李… III. JAVA 语言-程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 050593 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:郑 荃

北京京北制版印刷厂印刷·新华书店北京发行所发行

2006 年 7 月第 1 版第 1 次印刷

184mm×260mm·27.5 印张

定价:56.00 元(附光盘)

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

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，不仅擘划了研究的范畴，还揭橥了学术的源变，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，有的还专程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,我们的联系方法如下:

电子邮件: hzjsj@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
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

译者序

还记得几年前刚刚走出校园时候的一次面试经历。负责面试的技术主管出了一道 Java 编程题，我很容易就完成了。在和他谈话的时候，他又提出一个问题：“面向对象的 3 个主要特征是什么？”我毫不犹豫地就说出了答案：“封装、继承和多态。”接下来，他又问道：“什么是封装、继承和多态呢？”这时候我急得满头大汗，就是答不上来，最终仓皇败下阵来，错失一次良机。

其实，今天的很多大学生在学习 OO 编程的时候（不管是 Java 还是 C++），几乎都面临着和我当年相同的窘境：也许能够编写程序解决一定的问题，但是对“封装、继承和多态”等完整的 OO 思想知之甚少，更谈不上有什么理解了。这样的窘境，和 OO 编程的教学方式有着密不可分的关系。我们的学习总是过多关注编程技术细节，而忽略 OOP 的思想精髓；或者说，总是先学编程技术，后学 OOP 的思想精髓，造成编程实践和思想内涵的严重脱节。现在市面上很多 Java 编程图书，包括很多教材，仍然保持这种传统的学习方法。

本书“先教授完整的面向对象思想”的方法为我们学习面向对象编程带来了一些新颖的启发。本书的作者在大学多年从事教学实践工作，在美国 Brown 大学、Rhode Island 学院等大学进行了多年的“先教授完整的面向对象思想”的试验，取得了较好的效果和较大成功。更难能可贵的是，在“先教授完整的面向对象思想”的过程中，作者并不是干巴巴地进行理论讲解，而是通过生动的例子来说明封装、继承和多态。例如通过药囊、房间的电路系统的例子来说明封装，通过动物分类的例子来说明继承，通过超级英雄和图书馆的例子来说明多态。在每章介绍完这些基本概念之后，就在“方法”部分介绍 Java 中相关 OOP 概念的实现方法，随后趁热打铁，在“学以致用”部分通过精心设计的、生动活泼的图形化程序实例，帮助读者通过实践进一步加深对这些概念的理解，起到先介绍，后理解，再巩固的作用。

“先教授完整的面向对象思想”是本书的精华所在，然而却不是本书唯一的精华。正如本书的名字所表明的，“图形化方法”是本教材的另一个显著特点。如果说“先教授完整的面向对象思想”堪称本书的首创，“图形化方法”只不过是顺应潮流。为什么这么说呢？因为市面上已经不乏通过“图形化方法”来讲授 OOP 的图书，其中也有一些教材。在 OO 程序员的眼中，图形也是对象。在作者的精心构思下，一个个图形化程序实例充分展示了对象的内在变化和强大功能，生动地再现了封装、继承和多态等概念的应用。可以说，对象概念和技术与图形化程序结合在一起，相得益彰，产生了非常好的学习效果。

本书还有什么值得期待的地方呢？除了对表达式、条件语句、循环、递归、数据结构、I/O 等必备的 Java 编程知识进行详细系统地介绍外，本书还用专门的一章介绍了“设计模式”。这一章对于学生了解、认识设计模式，进而在自己的程序中应用模式打下了很好的基础。这对于读者今后的学习经历和职业生涯都是大有裨益的。

从知识内容的全面性来讲，本书对于 Java 网络编程、多线程等话题都没有涉及，这似乎略有些缺憾。但是作为教材，作者的这种选择就容易理解了，因为这些内容完全可以让学生在课后通过查阅相关的资料进行自学和扩展。总的来说，本书是一本具有很强的特色和实用性的教材。如果你希望自己的学生不再面临前面提到的那种尴尬，如果你希望在掌握编程技能之外对 OO 的精髓有更深刻的理解，如果你希望轻松而愉快地学习 Java 面向对象编程并牢固地掌握这些知识技能，那么，本书是你再合适不过的选择。

本书由李强负责主要的翻译工作人员。感谢机械工业出版社华章分社的编辑为本书出版工作付出的辛勤劳动。参与本书翻译的人员还有：关志兴、王建勇、毛立涛、闫柳青、姜巧生、沈海峰、谢扣林、乔义峰、刘查强、王义强、刘国际、杨传辉、王建华、汪明军、朱兆涛、毛付安、张男、杜兴平、刘志飞。由于译者水平有限，请读者不吝赐教。译文如有任何不妥之处，欢迎通过 reejohn@sohu.com 和我们交流探讨，以期共同进步。

前言

本书读者对象

本书介绍面向对象建模、问题解决和使用 Java 设计程序。目标读者是第一次学习编程的学生。试图学习 Java 的有经验的程序员，也可能会发现本书的教学方法很有趣，尽管他们可能会略过某些入门内容。

很多编程入门的图书都推崇 Java 的功能。在编写本书之前，我们看过很多这类图书，但没有一本是我们所期望的那种。本书有 5 个显著的特点，这 5 个特点是其他同类图书所不能同时满足的。

1. 首先介绍面向对象编程。
2. 寓教于乐。
3. 实用有效。
4. 使学生能够适应快速变化的领域。
5. 详略得当。

首先介绍面向对象编程

本书最独特的一点就是首先介绍面向对象编程的所有内容。面向对象语言有 3 个定义性的特征：封装、继承和多态。真正的面向对象编程要求程序员把这 3 点都用到。

现在，很多 Java 图书都是“先教对象”。也就是说，先教学生如何定义类和实例化对象，而把继承和多态推迟到学期末才教授：很早就接触对象，但面向对象编程却姗姗来迟。

这会引来一种争论：用“先教对象”方法教授的学生仅仅学会了良好的、结构化的、过程化的编程。毕竟，“先教对象”和我们教学生使用 Pascal 编程时的抽象数据类型的老思路没什么太大差异。尽管这种方式很容易教学，但是它的缺点是，在后面的课程中（也许是在第二学期），当教师介绍继承和多态思想的时候，学生很难从过程化的思路中转换过来。

另一种方法是先教对象和继承，但推迟教授多态。我们尝试过这种方法，但是发现没有了多态，良好的设计是不可能实现的。我们或者被迫去构造不需要多态的虚构的实例，或者使用一些迫使学生手工编码（使用条件分支）完成本该由编译器做的实例。为了推迟教授多态，我们牺牲了设计。

由于面向对象设计对我们的方法至关重要，因此我们决定在讲完对象和继承后立即介绍多态。但我们有两方面的担心：其一，多态的概念是否太难？其二，较早地教授多态是否会涉及更深的细节，而这些细节是否在以后讲更合适？

在采用这种方法近 10 年的教学实践中，我们看到学生认识到了多态的可管理性，并且较早地教授多态并不需要我们介绍太多的额外细节。较早地教授多态甚至达到了我们所期待的所有正面效果。学生可以使用多态进行很多实践，他们不需要改变使用条件来进行替代的习惯，并且我们可以在学期中较早地开始讨论面向对象设计。

寓教于乐——图形化方法

我们相信编程是件愉快的事。当我们学习编程的时候，仅是让计算机显示“Hello World!”

就能使我们兴奋，而基于文本或基于算术的程序足以吸引我们沿着成为计算机科学家的道路走下去。

今天的学生所生活的环境不太一样。大多数学生已经对计算机很熟悉。几乎所有的人都用过 Web、E-mail 和电脑游戏。他们对图形化界面很熟悉，但是很少有人看到过计算机的命令行界面。基于算术和基于文本的程序已经不能像多年前那样，给他们带来兴奋和神秘感了。

为了给学生带来我们曾拥有的那种对计算机编程的兴奋感觉，在本书的第一部分中，我们使用图形来设计所有的实例和作业。学生可以编写简单的动画、GUI 和游戏。基于数字和文本的计算将在本书后面的部分涉及。

实用有效

虽然这些实例很有趣，但学生毕竟是在学习 Java 编程，这是当今世界上最广泛使用的编程语言之一。在本教程的最后，我们所使用的类库是专业程序员所使用的标准库。

编写图形化程序不仅是为了获得乐趣，它也是一项很有用的技能，因为市场上的几乎每一个软件包都有图形化的界面。实际上，既然学生将要编写的大多数程序都是使用图形界面的交互式程序，那么学习使用简单的图形化对象就应该是他们掌握编程的一部分。在一种像 Java 这样的面向对象语言中，图形化小部件及 UI 小部件和对象之间自然就有映射关系。

使学生能够适应快速变化的领域

计算机科学教师所面临的最大挑战就是要使学生适应快速变化的领域。即便我们能够教会学生使用当今数以百计的最流行的工具和软件包的每个细节，仍然是不够的，5 年后，现在认为是“必须掌握的”软件列表到那时也会变化，学生必须从头开始。因此，和其他领域的教师相比，我们必须教会学生如何学习，并且给他们一个基本的知识结构，这个结构要适合他们将要学习的新东西。

因此，本书并不只是介绍 Java 语言，还介绍了用 Java 实现面向对象建模和设计。我们强调程序是模型并且面向对象程序是一组协同工作的对象（每个对象通常都可以拆解为协同工作的对象组的一个子系统）的模型。我们提供了一个设计面向对象模型的流程，在书中给出了大量实例和有一定难度的作业。

本书是针对美国 CC2001 (*Computing Curricula* 2001) 中所述的先讲对象顺序的两门课程的第一门课，或者三门课程的前两门课而量身定做。本书包含有关数据结构和递归的章节，以及简要介绍查找算法、排序算法和算法分析的章节，另外，本书还有单独的一章介绍设计模式。希望严格遵从 CC2001 的教师可以在本书的基础上再添加一些有关计算历史和计算所包含风险的内容。

详略得当

对于任何教授 Java 的人来说，最大的挑战之一就是当学生需要掌握一些细节的时候能够告诉他们足够多的细节，且又恰到好处。Java 是一种庞大而复杂的语言。教师必须避免让学生看到过多的树木，而导致他们看不到森林。例如，让他们学习太多的语法而不能理解概念。希望学生从一开始就能编写有趣的程序是很难的，而希望学生编写图形化程序更是几乎不可能的。

为了让学生把注意力集中到关键概念上而不是语法上，我们提供了一个名为 *Wheels* 的定制的图形库。*Wheels* 是 Java 自己的图形库 (AWT 和 Swing) 的一个非常简单的外包类 (wrapper)，它使得教师可以介绍图形而又不必去解释 AWT 和 Swing 的所有复杂的知识点。

Wheels 很容易学习, 它只包含 6 个接口和 14 个类。学生只需要知道 6 个类: Frame、Ellipse、Line、Rectangle、RoundedRectangle 和 ConversationBubble。学生可以使用这些类在窗口中创建由二维图形组成的窗口和图片。另外, ConversationBubble 类使他们能够创建卡通动画。如果需要, 图形可以被点击, 但是为了简单起见, 这个功能最初是隐藏的。Wheels 的所有代码都包含在本书的随书光盘中。

由于我们希望学生在学期末能够使用“真正的”Java, 因此我们只在本书的前 6 章用到 Wheels。之后用两章专门介绍 Java 中图形编程, 在这两章中, 我们给学生一个性能良好的 AWT 和 Swing 的子集, 并且简单描述了事件处理。因此, 学生可以直接使用 AWT 和 Swing 库。

章节顺序

在 Brown 大学, 我们在一个 14 周的学期中讲授了第 0 章到第 16 章的内容, 差不多是整本书的内容。在 RIC, 我们用相同的时间讲授了第 0 章到第 11 章以及第 13 章的内容。也可用其他方法组织章节。无论怎样都应包含第 0 章到第 5 章的所有内容。第 7 章(图形)和第 8 章(GUI)及第 6 章(算术和条件)可以根据这些章的内容来选择。也可以先讲后面章节再讲第 7 章和第 8 章(或者干脆略掉这两章), 但是这需要在后面的章节中用自己编写的“学以致用”部分的程序。

本书特色

组织结构: 为了区分独立于语言的内容和特定于 Java 的内容, 我们必须协调一致地组织各个章节。“概念”部分主要涵盖了对于多种语言都很有用的信息, “方法”部分则主要介绍特定于 Java 的细节。

关键概念: 贯穿本书的“关键概念”为学生指出每一章中最重要的概念。学生应密切注意“关键概念”周围的内容。这部分也可以充当一个帮助学生复习备考的工具。

“学以致用”部分: 理解 Java 概念的最好的方法就是参与程序开发的过程。每章后面都有“学以致用”部分, 将该章的概念和技术应用到一个可运行的程序中, 这部分适合在课堂试验中探讨。这些程序的源代码都在本书的随书光盘中提供。

练习题: 每章末尾都有几种类型的练习。“自我检测题”是针对该章中内容的简单的实际问题。“知识应用题”和“程序调试题”是需要稍加思索的纸笔练习题。“修改程序题”要求修改一个给定的程序(通常是同一章中的“学以致用”部分的程序)。如果你在最近的试验中要求学生做过“学以致用”部分, 那么最好用相应的“修改程序题”来巩固知识。“编写程序题”要求从头编写一个程序。

补充资料

如果你购买了本书, 会得到本书的随书光盘, 其中包括:

- 本书中“学以致用”部分的源代码文件。
- Wheels 库。
- J2SE 5.0。

下面的教辅资料有密码保护, 仅供采用本书作为教材的教师使用。需要者请先填写书后的“教学支持说明”, 邮寄、传真或发邮件到培生教育出版集团北京办事处(联系方式见“教学支持说明”), 经审核通过之后获得用户名及密码, 之后再登录 www.aw.com/computing 下载下面的教辅资料。

教辅资源包括:

- PowerPoint 幻灯片电子教案。
- 习题答案。
- 试题库。

致谢

没有我们的学生、学校和家庭的支持，本书不可能诞生。我们要特别感谢 Brook Conner 提出了基本的先讲对象的方法；感谢 Brook Connor 和 Robert Duvall 编写了关于面向对象编程章节的初稿；感谢很多年级的大学生助教对于 Wheels 不同版本所做的工作（尤其是 Matt Chotin 和 John Goodwin）；感谢布朗大学 CS015 级的学生、RIC CS201 级的学生和 UConn 大学 CSE133 级的学生耐心地使用这些章节的粗糙初稿；感谢我们在布朗大学和 RIC 以前的学生对于初稿带有批评眼光的阅读、评论，以及在线上和线下参与漫长的讨论，尤其感谢 Michael Black、Matt Chotin、David Eustis、David Goldberg、John Goodwin、Bernie Gordon、Erik Holder、Katherine (Casey) Jones、Christopher Maloney、Aaron Myers、Shirin Oskooi、Leah Pearlman、Devon Penney、Janete Perez、James Piechota、Jason Sobel、Christopher (Zeke) Swepson、Dana Tenneson、Susan Warren、Stacy Wong 和 A.J. Young。

特别感谢 Teresa McRann 编写了大多数调试练习；感谢 Zeke Swepson 为 Blob 示例程序所做的工作；感谢 Charles Berube 为最初的 FoodService-Professional 实例所做的工作；感谢 Marc Rene 在最初阶段针对软件测试所做的讨论；感谢 Uppsala 大学的 Anna Eckerdal 在最终编辑阶段的热情和鼓励；感谢康涅狄格大学的 Robert McCartney 和 Ruth Simons 以及 RIC 的 Ann Moskol 在他们的课程中提供本书章节的多个版本的测试，并且提出了宝贵的反馈意见；感谢布朗大学的 Tom Doepner 阅读和评论了第 18 章；感谢布朗大学的 Shriram Krishnamurthi 和 Steve Reiss，以及康涅狄格大学的 Laurent Michel 对编程语言和设计问题进行了很有帮助的讨论。感谢 RIC 数学和计算机科学系主任 Helen Salzberg，以及 RIC 艺术与科学学院院长 Richard Weiner 对我们的鼓励和支持。

感谢外界的审阅者和他们很有帮助的评论。他们是 SUNY Buffalo 的 Carl Alphonse、科罗拉多州立大学的 Elizabeth Boese、俄亥俄州迈阿密大学的 Fazli Can、Brandeis 大学的 Jacques Cohen、北佛罗里达大学的 Sherif Elfayoumy、罗彻斯特科技学院的 Henry Etlinger、华盛顿大学的 Stan Kwasny、东北大学的 Michael Lipton、中佛罗里达大学的 Mark Llewellyn、玛利亚学院的 Mauricio Manengoni、Clemson 大学的 John McGregor、Austin 德克萨斯大学的 Shyamal Mitra、威尔明顿的北卡罗莱纳大学的 Sridhar Narayan、马凯特大学的 Richard Povinelli、Austin 德克萨斯大学的 Roger Priebe、华盛顿大学的 Stuart Regis、Lane 社区学院的 Jerry Ross、北爱荷华大学的 Ben Schafer、佛罗里达大学的 Dave Small、新墨西哥州立大学的 Esther Steiner、蒙哥马利大学的 Gary Thai、乔治华盛顿大学的 Kent Vidrine 和伍斯特州立大学的 Karl Wurst。

感谢 Trina Avery 对本书进行的精心编辑，感谢 Rosemary Simpson 对理想索引的要求，以及她在有限的篇幅中给出的我们必须接受的一种实现方法，感谢 Lauren Clarke 沉着的组织工作，感谢 Addison-Wesley 为了本书而工作的整个团队，尤其是 Michael Hirsch、Susan Hartman、Patty Mahtani、Joyce Wells 和 Marilyn Lloyd。

感谢国家科学基金会的支持，他们通过 DUE-0410546 确保我们能够开发和改编那些和本书方法相关的资料。

最后，感谢 Robert 和 Debbie。我们对你们充满感激。

目 录

出版者的话	
专家指导委员会	
译者序	
前言	
第 0 章 背景知识	1
0.1 简介	1
0.2 计算机硬件	1
0.3 计算机软件	3
0.4 编程语言和 Java	3
0.5 执行程序	5
0.6 编程的软件工具	5
0.7 Java 中的编译器如何工作	6
0.8 编写程序的过程	7
0.9 程序不工作怎么办	9
小结	10
自我检测题	10
练习题	10
第 1 章 对象	13
1.1 简介	13
1.2 概念	14
1.2.1 什么是模型	14
1.2.2 如何构建模型	15
1.2.3 如何拥有相同类型的不同 对象: 类和实例	16
1.2.4 管理复杂性	17
1.2.5 是什么造就一个好的软件模型	18
1.3 方法	19
1.3.1 橡皮图章: 声明和定义一个类	20
1.3.2 对属性建模: 实例变量	21
1.3.3 对能力建模 I: 构造方法	23
1.3.4 为属性赋值: 赋值语句	24
1.3.5 对能力建模 II: 更多有关 方法的知识	25
1.3.6 UML 图	27
1.4 学以致用: 对象——学习 Java 的 起点	28
1.4.1 起步	29
1.4.2 背景: 图形库 Wheels	29
1.4.3 分析	31
1.4.4 设计	31
1.4.5 实现	31
1.4.6 执行的步骤	32
1.4.7 测试	32
1.4.8 继续	32
小结	33
自我检测题	33
练习题	34
编程项目	37
最新风格惯例	37
第 2 章 带有参数的方法	39
2.1 简介	39
2.2 概念	40
2.2.1 参数	40
2.2.2 返回值	41
2.3 方法	42
2.3.1 参数	42
2.3.2 带有参数的构造方法	43
2.3.3 方法调用的调试技巧	45
2.3.4 参数、局部变量和实例变量	45
2.3.5 返回类型	48
2.3.6 方法签名和重载	49
2.4 学以致用: 对象——会说话的太阳	49
2.4.1 背景: 关于 Wheels 的更多知识	50
2.4.2 分析	51
2.4.3 设计	52
2.4.4 实现: 版本 1	52
2.4.5 导入、包和库	52
2.4.6 版本 2: 添加构造方法体	53
2.4.7 Java 的内建颜色	54
2.4.8 测试版本 2	54
2.4.9 把图形放到 Frame 中的不同 位置	54
2.4.10 实现: 版本 3	55
2.4.11 测试版本 3	56

2.5 学以致用: 复合对象——雪人	56	编程项目	88
2.5.1 分析	56	第 4 章 接口	91
2.5.2 设计	57	4.1 简介	91
2.5.3 实现: 复用 SunCartoon	58	4.2 概念	91
2.5.4 测试: 程序的第一段	58	4.3 方法	92
2.5.5 实现: Hat	58	4.3.1 定义和实现接口	92
2.5.6 测试 Hat: 部分 1	59	4.3.2 实现多个接口的类	93
2.5.7 设置复合对象的位置: Hat	60	4.3.3 扩展接口	95
2.5.8 测试 Hat(续)	61	4.3.4 设计选择: 接口 VS. abstract 类	96
2.5.9 组装起来: 实现 Snowman	62	4.3.5 设计选择: 使用多少个接口	97
2.5.10 测试 Snowman: 部分 1	62	4.3.6 设计选择: 哪个能力放入到 哪个接口中	97
2.5.11 实现 Snowman(续)	62	4.4 学以致用: 接口——一个会动的 太阳	97
2.5.12 测试 Snowman(续)	63	4.4.1 编写一个接口	98
小结	63	4.4.2 编写一个实现了接口的类	99
自我检测题	64	4.4.3 实现 mouseDragged 方法	99
练习题	64	4.4.4 实现 Frame: MovableSunApp	100
编程项目	66	4.5 学以致用: 接口和复合图形—— 一顶可移动的帽子	101
最新风格惯例	68	4.5.1 起步: 接口和 MovableHatApp	102
第 3 章 继承	69	4.5.2 创建知道其包含者的一个部分: HatPart 类	102
3.1 简介	69	4.5.3 创建一个像单元一样移动的复合 对象: Hat 类	103
3.2 概念	70	4.6 秘诀	104
3.3 方法	71	4.6.1 定义类	105
3.3.1 简单继承	71	4.6.2 初始化实例变量	106
3.3.2 覆盖继承的方法	73	4.6.3 带有默认值的实例变量	106
3.3.3 实现 abstract 方法	74	4.6.4 accessor 和 mutator	107
3.3.4 继承属性: 伪继承	76	4.6.5 知道包含者的部分	108
3.3.5 方法解析	78	小结	108
3.4 学以致用: 继承——Blobs	79	自我检测题	109
3.4.1 背景: Wheels 中的鼠标方法	80	练习题	109
3.4.2 分析	80	编程项目	110
3.4.3 设计	81	第 5 章 多态	111
3.4.4 实现: 带有一个 Blob 的 BlobApp	82	5.1 简介	111
3.4.5 测试(部分 1)	83	5.2 概念	112
3.4.6 实现: 带有一个 WinkingBlob 的 BlobApp	83	5.3 方法	114
3.4.7 测试(部分 2)	84	5.3.1 继承多态	114
3.4.8 实现: 带有 WinkingBlob 和 TalkativeBlob 的 BlobApp	84	5.3.2 接口多态	115
3.4.9 最终测试	85	5.3.3 确定调用哪个方法	116
小结	85	5.4 学以致用: 多态——画板	117
自我检测题	85		
练习题	86		

5.4.1 分析	118	7.3.6 绘制图形和 Graphics 对象	156
5.4.2 设计	118	7.4 学以致用: 事件和动画——	
5.4.3 实现	119	弹跳的球	158
5.4.4 测试	122	7.4.1 Java 事件模型	159
小结	123	7.4.2 使用事件: 让球开始弹跳	163
自我检测题	123	7.4.3 旋转图形	166
练习题	124	7.5 学以致用: 事件和复合图形——	
编程项目	127	鱼缸	166
第 6 章 算术表达式和条件语句	129	7.6 学以致用: 设计——构建自己	
6.1 简介	129	的 Wheels	173
6.2 概念	129	7.6.1 Java 的内建图形类	173
6.2.1 表示数字	130	7.6.2 新的 ColorShape 类	174
6.2.2 算术运算符	130	小结	178
6.2.3 条件语句和布尔逻辑	131	自我检测题	178
6.2.4 关系运算符	133	练习题	178
6.2.5 对象相等	133	编程项目	179
6.2.6 从一个 yes-no 选择中选取	134	第 8 章 图形用户界面和事件处理	183
6.2.7 从两种选择中选取	134	8.1 简介	183
6.2.8 从多种选择中选取	134	8.2 概念	184
6.2.9 何时使用条件语句	135	8.2.1 设计原则	184
6.3 方法	136	8.2.2 设计工具	187
6.3.1 Java 中的数字	136	8.3 方法	188
6.3.2 算术运算符	137	8.3.1 在屏幕上排列组件: 布局	189
6.3.3 自增运算和自减运算	138	8.3.2 组件	192
6.3.4 布尔运算符	139	8.3.3 让程序能够响应输入: 事件处理	
6.3.5 关系运算符	139	和 SmartComponent	195
6.3.6 单个选择	140	8.4 学以致用: GUI——一个用户控制的	
6.4 学以致用: 算术表达式和条件语句——		弹跳球	200
雏菊, 雏菊, 告诉我答案	141	8.4.1 部分 1: 布局	201
小结	145	8.4.2 ControlPanel	202
自我检测题	145	8.4.3 QuitButton	203
练习题	146	8.4.4 添加带有 BouncingBall 的	
编程项目	148	BallPanel, 添加颜色按钮	
第 7 章 二维图形	149	并使它们工作	204
7.1 简介	149	8.4.5 添加滑块	207
7.2 概念	150	小结	209
7.3 学以致用: 窗口和图形——修改后		自我检测题	210
的 FirstApp	150	练习题	210
7.3.1 JFrame	151	编程项目	212
7.3.2 JPanel	152	第 9 章 设计模式	215
7.3.3 布局	153	9.1 简介	215
7.3.4 把一个椭圆形放入到 JPanel	154	9.2 概念	216
7.3.5 创建一个“聪明的”椭圆形	154	9.3 方法	217

9.4 学以致用: Holder 模式——绘图	
程序的一部分	217
9.4.1 对当前颜色建模: Holder 模式 ..	218
9.4.2 ColorHolderApp	219
9.4.3 ControlPanel 类和按钮类	220
9.4.4 ColorShapePanel	221
9.5 学以致用: 代理模式——赛车	222
9.6 学以致用: 复合模式——太空	
外星人	233
小结	236
自我检测题	236
练习题	237
编程项目	237
第 10 章 高级算术表达式和条件	
语句	239
10.1 简介	239
10.2 概念	239
10.2.1 对实数建模	239
10.2.2 浮点数的运算	240
10.2.3 强制类型转换	240
10.2.4 对多重选择建模	240
10.3 方法	241
10.3.1 Java 中的浮点数	241
10.3.2 浮点数的算术运算和强制	
类型转换	243
10.3.3 Java 的 Math 类: static 变量	
和方法	245
10.3.4 随机数	245
10.3.5 常量: 如何、何处以及何时	
定义它们	246
10.3.6 对多重选择建模: 嵌套	
条件语句	249
10.3.7 短路	250
10.3.8 switch 语句	251
10.4 学以致用: 浮点数、嵌套条件语句	
和随机数——ColorBounceApp	253
小结	257
自我检测题	257
练习题	258
编程项目	259
最新编程惯例	259
第 11 章 循环	261
11.1 简介	261
11.2 概念	262
11.3 方法	262
11.3.1 Vector	262
11.3.2 无限循环: while 循环	265
11.3.3 Iterator	267
11.3.4 无限循环: do-while 循环	268
11.3.5 有限循环: for 循环	268
11.3.6 循环中的 break 和 continue	270
11.3.7 在循环中选择	271
11.3.8 编写循环的秘诀	271
11.3.9 调试循环	273
11.3.10 控制流程	275
11.4 学以致用: 循环——ButterflyApp ..	276
小结	280
自我检测题	281
练习题	281
编程项目	283
第 12 章 递归	285
12.1 简介	285
12.2 概念	286
12.3 方法	286
12.3.1 带有一个递归调用的例子:	
阶乘	287
12.3.2 带有多个递归调用的例子:	
汉诺塔	289
12.3.3 调试技巧	290
12.3.4 在迭代和递归之间选择	291
12.4 学以致用: 递归——SpiralApp	293
12.5 学以致用: 递归——TreeApp	297
小结	300
自我检测题	300
练习题	300
编程项目	301
第 13 章 数组、Vector 和 ArrayList	303
13.1 简介	303
13.2 概念	303
13.3 方法	304
13.3.1 声明和初始化数组、ArrayList	
和 Vector	304
13.3.2 访问和使用元素	305
13.3.3 作为参数传递	306
13.3.4 多维结构	306
13.3.5 调试技巧: 常见的数组错误 ..	308

13.3.6 比较数组、Vector 和 ArrayList	308	编程项目	358
13.3.7 主线程揭秘	309	第 16 章 排序和查找	359
13.4 学以致用: 数组——ClickApp	310	16.1 简介	359
小结	313	16.2 概念	360
自我检测题	313	16.2.1 排序	362
练习题	314	16.2.2 冒泡排序	362
编程项目	315	16.2.3 插入排序	365
第 14 章 数据结构	317	16.2.4 选择排序	367
14.1 简介	317	16.2.5 合并排序	367
14.2 概念	318	16.2.6 递推关系	369
14.2.1 堆栈	318	16.2.7 查找: 概述	371
14.2.2 队列	319	16.2.8 查找: 数组	371
14.2.3 链表	320	16.2.9 查找: 链表	372
14.2.4 字典	320	16.2.10 查找: 二叉树	372
14.2.5 使用链表实现 ADT	321	16.2.11 查找: n 叉树和混合树	373
14.2.6 实例图	321	16.2.12 查找: 哈希表	374
14.3 方法	323	小结	376
14.3.1 Node 类	323	自我检测题	376
14.3.2 StackADT 接口	324	练习题	377
14.3.3 Stack 类	324	编程项目	378
14.3.4 QueueADT 接口	326	第 17 章 字符串和文本 I/O	379
14.3.5 Queue 类	326	17.1 简介	379
14.3.6 ListADT 接口	328	17.2 概念	379
14.3.7 实现 List 和迭代器设计模式	328	17.3 方法: String	380
14.3.8 DictionaryADT 接口	329	17.3.1 组合 String	381
14.3.9 Dictionary 类	332	17.3.2 提取 String 的一部分	381
14.4 学以致用: 数据结构——测试		17.3.3 比较 String	382
程序	337	17.3.4 反转 String	383
小结	339	17.3.5 String 类概览	384
自我检测题	339	17.4 学以致用: String——GUI	
练习题	340	文本 I/O	384
编程项目	341	17.4.1 简单文本 I/O 程序的一个	
第 15 章 树	343	模板 GUI	385
15.1 简介	343	17.4.2 一个简单应用程序:	
15.2 概念	344	EchoApp	388
15.2.1 使用树建模	344	17.4.3 另一个应用程序:	
15.2.2 使用树高效实现字典操作	345	StringMethodsApp	389
15.2.3 中序、先序和后序	348	17.4.4 第三个应用程序:	
15.3 方法: 二叉查找树	350	EqualTestApp	390
15.4 学以致用: Tree——测试程序	357	小结	392
小结	357	自我检测题	392
自我检测题	358	练习题	392
练习题	358	编程项目	393