

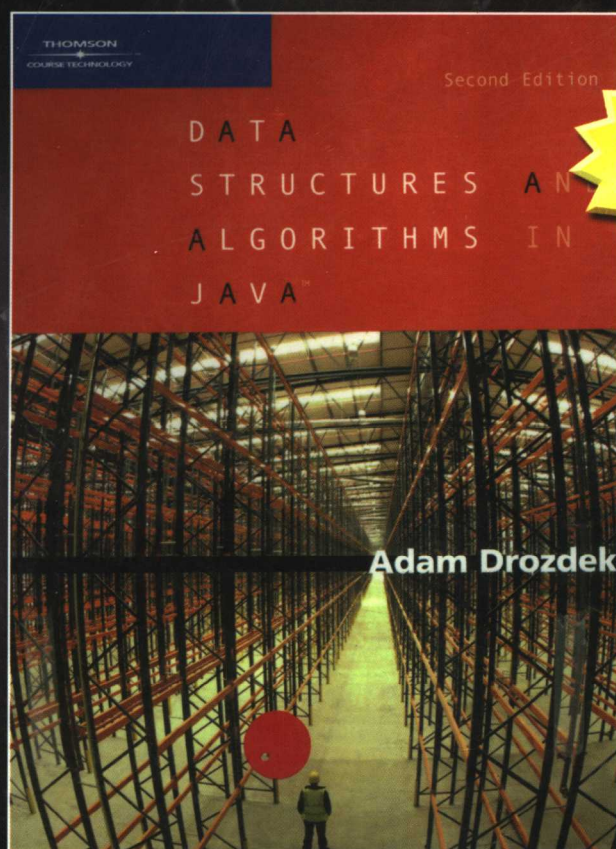
THOMSON

计 算 机 科 学 丛 书

数据结构与算法

Java语言版

(美) Adam Drozdek 著 周翔 译



第2版

Data Structures and Algorithms in Java
Second Edition



机械工业出版社
China Machine Press

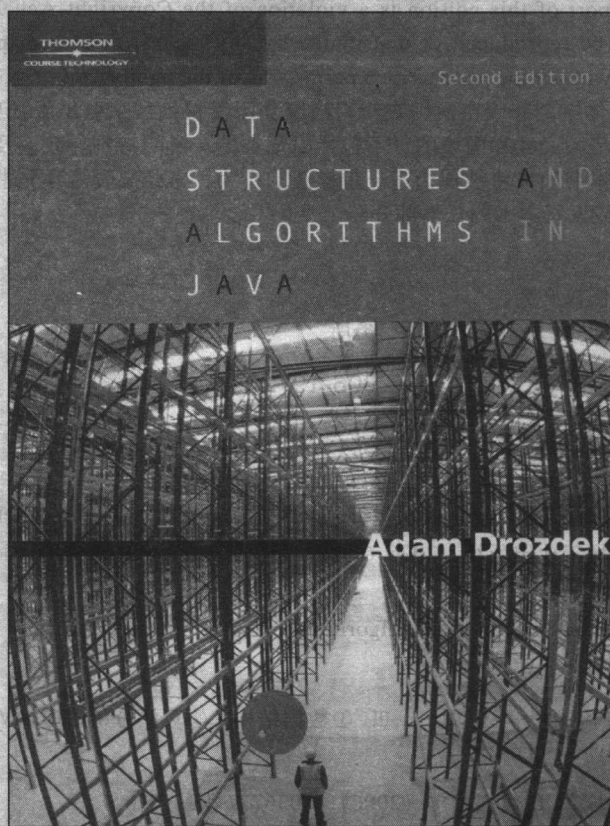
计 算 机 科 学 丛 书

第2版

数据结构与算法

Java语言版

(美) Adam Drozdek 著 周翔 译



Data Structures and Algorithms in Java
Second Edition



机械工业出版社
China Machine Press

本书系统讲解数据结构和算法，并分析了算法的复杂性。本书选择Java语言以面向对象的方式描述数据结构，还特别强调了封装和分解的信息隐藏原理。主要内容包括：面向对象编程的基本原理，判定算法效率的方法，堆栈、队列及其应用，对于多种递归的详细讨论，二叉树、B树、2-4树等的查找和遍历等，分析排序、散列等数据结构的应用，图、NP完整性，数据压缩算法、存储管理技术以及自动机理论和字符串匹配等算法。

本书适合作为高等院校计算机专业的教材，也是计算机算法方面的重要参考书。

Adam Drozdek; Data Structures and Algorithms in Java, Second Edition.

ISBN: 0-534-49252-5

Copyright © 2005 Course Technology, a division of Thomson Learning, Inc.

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd). All rights reserved. 本书原版由汤姆森学习出版集团出版。版权所有，盗印必究。

China Machine Press is authorized by Thomson Learning to publish and distribute exclusively this simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体字翻译版由汤姆森学习出版集团授权机械工业出版社独家出版发行。此版本仅限在中华人民共和国境内（不包括中国香港、澳门特别行政区及中国台湾）销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可，不得以任何方式复制或发行本书的任何部分。

978-981-4195-38-6

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：01-2006-1972

图书在版编目 (CIP) 数据

数据结构与算法：Java语言版（第2版） / (美) 德罗兹德克 (Drozdek, A.) 著；周翔译. —北京：机械工业出版社，2006. 6

(计算机科学丛书)

书名原文：Data Structures and Algorithms in Java, Second Edition

ISBN 7-111-18993-0

I. 数… II. ①德… ②周… III. ①数据结构 ②算法分析 ③JAVA语言—程序设计 IV. ①TP311.12 ②TP312

中国版本图书馆CIP数据核字 (2006) 第036777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：王 玉 刘 晖

北京瑞德印刷有限公司印刷 · 新华书店北京发行所发行

2006年7月第2版第1次印刷

184mm × 260mm · 37印张

定价：59.00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

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相助，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定造的。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：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

随着计算机的普及和硬件的不断升级,软件业的市场不断扩大。当前的用户对软件的期望不仅局限于满足功能方面的需求,同时还要求节约空间、提高效率。因而,对程序设计者提出了更高的要求。编写比传统意义更“好”的软件在一定程度上表现为寻求最佳的算法以及简洁的数据,数据结构的内容正是试图涵盖这两方面的理念。

编程语言自出现以来,经历了结构化编程—面向对象—面向方面的发展过程。目前,编程者广泛采用的仍是面向对象的OOP技术。Java语言作为一种面向对象的编程语言,在诞生之后,因其具有完备的功能和跨平台的特性,逐渐为计算机领域的广大科技工作者所接受。Java不仅可以用来实现计算机用户的各项基本要求,同时还能为当前流行的嵌入式、网络分布式系统服务。由于Java语言在商业领域取得的巨大成就,被很多国外大学选用为计算机专业学生的程序设计语言课程。本书采用Java语言实现算法,取代过去传统的C语言教学模式,突出OOP的封装和分解的信息隐藏原理,以更好地适应当前计算机技术更新进步的需要。

本书2003年出版了第1版,因取材新颖,被很多国内外院校采用为教学参考书。第2版相对于第1版而言,延续了理论结合实际的风格,在介绍理论的同时,配合大量的实际应用案例。用丰富生动的图表诠释枯燥的概念。同时,在章节后面附以示例学习和编程作业,循序渐进、深入浅出地帮助读者理解并灵活应用数据结构理论。

本书由周翔统稿翻译,由隋立恒完成了全书的审校。全书共分13章,其中第1章、第2章介绍了Java的入门知识以及复杂性计算的数学基础,特别对第1版而言,添加了关于NP完整性的讨论;第3~8章介绍了一些基本的数据结构以及操作算法,其中增加了很多经典问题,如图论的邮递员、旅行商、团问题等;第9~12章描述各种数据结构在软件开发中的应用;第13章字符串匹配是第1版中未涉及的领域。字符串作为计算机常用的一种数据,其编码、匹配技术广泛应用在软件开发和网络检测等各项领域。本书结合自动机理论和各种数据结构给出了多种匹配算法;最后的附录则是数学知识方面的补充。对使用者尤其是教学人员而言,第2版内容的更新、补充和整体编排、重构意义重大,作者试图从更深层面、更广泛的角度与读者做有意义的探讨。

作为一名不够资深的译者,我深知计算机行业发展的迅速和自己水平的有限,虽多经揣摩但译文中仍难免存在一些疏误,真诚欢迎热心的读者给予批评指正,以便帮助我们学习提高和今后再版时校正。

在此还要感谢我的家人和朋友在翻译过程中给予的理解和支持,特别要感谢机械工业出版社华章分社的编辑负责、细致的工作,使本书顺利地翻译出版,与读者见面。

周翔

2006年2月

前 言

数据结构既是计算机科学教育的一个重要组成部分，又是其他计算机研究领域的基础。数据结构的很多知识是希望从事软件系统的设计、实现、测试和维护的学生所必须具备的。本书为这类学生提供了必要的知识。

本书主要讲述了数据结构的三个重要方面。第一，重点强调数据结构和算法间的联系，包括分析算法的复杂性。第二，根据当前设计和实现范例，以面向对象的方式描述数据结构。特别是强调了增强封装和分解的信息隐藏原理。最后一个重要部分是数据结构的实现，选择Java作为编程语言。

Java语言作为C和C++的面向对象的后代语言，流行于工业和学术领域内。由于它在Internet上的广泛应用，被认为是一种优秀的编程语言。因为Java一致地使用面向对象的性质以及语言的安全性，所以，用Java来介绍数据结构是很自然的事情。当前，C++是数据结构教学的主选语言。不过，由于Java在应用程序编程中的广泛应用以及它的面向对象特性，使用Java来讲授数据结构和算法课程，即使对于初学者也是很恰当的。

本书可以作为旧版ACM课程中CS2和CS7的课程教材，同时，也符合新版ACM课程中C_A202，C_D202和C_F204大多数课程的要求。

很多章都包括示例学习，用来说明某个算法和数据结构的应用的完整上下文。这些示例学习是从计算机科学的各个领域（比如解释器、符号计算和文件处理等）中挑选出来的，反映了讨论的主题可以应用的广阔范围。

Java代码的简单实例贯穿全书，阐述了数据结构在实践中的重要性。但是，理论分析也是同样重要的。因此，算法描述往往和效率分析结合在一起。

在描述递归时要特别小心，因为即使是高年级学生在处理它时也会出问题。经验说明，只有考虑运行时堆栈，才能最好地诠释递归。不只是在递归的章节，在其他章也有对递归函数的跟踪，用来显示堆栈的变化。例如，如果不解释系统在运行堆栈上所做的工作，那么遍历树的出奇短的方法可能就是个谜。讨论数据结构和算法时如果远离系统，用纯理论的态度未必有效。本书还全面地讨论了数据压缩和存储管理。

本书介绍的重点是数据结构，其他方面的主题讨论是为了帮助读者正确地理解数据结构。算法是从数据结构的角度的考虑的，因此，读者看不到多种不同类型的算法的全面讨论以及对某个算法的完整描述。但如前所述，本书深刻地诠释了递归，还详细描述了算法的复杂性分析。

第1章和第3~8章介绍了不同的数据结构和操作它们的算法。分析了每个算法的效率，并提出了改进算法的建议。

- 第1章讲述面向对象编程的基本原理，介绍动态内存分配和指针的使用以及Java基础知识。
- 第2章描述评定算法效率的方法。
- 第3章介绍链表。

- 第4章介绍堆栈、队列及其应用。
- 第5章包含对递归的详细讨论，讨论多种类型的递归并剖析一个具体的递归调用。
- 第6章讨论二叉树，包括实现、遍历和查找。并且介绍了平衡树。
- 第7章描述更一般的树，如检索树、2-4树和B树。
- 第8章描述图的理论和方法。

第9~12章描述了前几章介绍的数据结构的不同应用，突出了每个主题所涉及的数据结构。

- 第9章详细分析排序，描述了几个基本的和非基本的方法。
- 第10章讨论散列——查找中最重要的领域之一。在描述不同技术的同时，强调数据结构的使用。
- 第11章讨论数据压缩算法和数据结构。
- 第12章描述存储管理的不同技术和数据结构。
- 第13章讨论精确和近似字符串匹配的多种算法。
- 附录A详尽讨论第2章介绍的大 O 符号。
- 附录B给出库克（Cook）定理的证明，并结合示例进行阐述。

每一章包含对某个素材的讨论，并配有相应的图表说明。除第2章之外，其他所有章节都包含示例学习，即使用该章讨论的特性的扩展实例。所有的示例学习都在PC机上用Visual C++编译器，或是在UNIX上用g++编译器测试过（除von Koch雪花问题之外）。每章的后面是各种难度的习题。除了第2章之外，每一章还有编程作业和最近的相关文献的参考书目。

第1~6章（除了第2.9, 3.4, 6.4.3, 6.7和6.8节之外）包含数据结构课程的核心素材，应当顺序学习。余下的六章可以按任意顺序学习。一个学期的课程可以包括第1~6章，第9章，第10.1节和10.2节。全书也可以作为两学期课程的教材。

教学工具

电子版教师手册。教师手册包含了本书所有习题的完整解答。

电子版图文件：提供授课所需的本书中所有图像文件。

源代码。书中的示例源代码可以通过Web站点下载，地址是：<http://www.mathes.duq.edu/drozdek/DSinJava>。

源代码部分，学生也可以从www.course.com下载。除此之外，上述教辅资源仅适用于教师教学使用。需要者请与汤姆森学习出版集团北京代表处联系。（具体联系方式请见本书最后所附的“教学支持服务”说明表。——编辑注）

第2版的变化

相对于第1版，本书增加了一些关于新主题的素材。主要有

- 新添的第13章的模式匹配算法。
- 简要介绍NP完整性（第2.10节），举例讨论NP完整性问题（第8.12节），库克定理的概要（附录B）。
- 关于图的新素材（第8.9.1节，8.10.1.1节，8.10.2.1节，8.11节）。
- 讨论垂直水平树的删除算法（第7.1.7节）。

- 介绍Java文件（第1.3.1~1.3.6节）。

同时还更新了java.util软件包中的方法列表。此外，还有一些小的改动，不再一一赘述。

致谢

衷心感谢下列人士的审阅，他们为本书的改进提供了宝贵意见和建议：James Ball (Indiana State University)、Robin Dawes (Queen's University) 和Julius Dichter (University of Bridgeport)。

作为本书作者，欢迎读者对本书中的疏漏和不足进行指正，我的电子邮箱是：drozdek@duq.edu。

Adam Drozdek

目 录

出版者的话	
专家指导委员会	
译者序	
前言	
第1章 Java语言的面向对象编程	1
1.1 Java入门	1
1.1.1 变量的声明	1
1.1.2 运算符	3
1.1.3 选择语句	3
1.1.4 循环语句	4
1.1.5 异常处理	5
1.2 Java面向对象编程	6
1.2.1 封装	6
1.2.2 抽象数据类型	12
1.2.3 继承	13
1.2.4 多态性	16
1.3 输入和输出	19
1.3.1 输入、输出字节	21
1.3.2 行输入	21
1.3.3 标志输入：单词和数字	22
1.3.4 基本数据类型的输入和输出	22
1.3.5 对象的输入和输出	23
1.3.6 随机存取文件	24
1.4 Java和指针	24
1.5 java.util中的向量	28
1.6 数据结构和面向对象的编程	32
1.7 示例学习：随机存取文件	32
1.8 习题	39
1.9 编程作业	41
参考文献	42
第2章 复杂性分析	44
2.1 计算复杂性和渐近复杂性	44
2.2 大O表示法	44
2.3 大O表示法的性质	46
2.4 Ω 和 Θ 表示法	47
2.5 可能出现的问题	48
2.6 复杂性示例	48
2.7 寻找渐近复杂性：示例	49
2.8 最好的、平均的和最坏的情况	51
2.9 平摊复杂性	53
2.10 NP完整性	56
2.11 习题	58
参考文献	60
第3章 链表	62
3.1 单向链表	62
3.1.1 插入	66
3.1.2 删除	67
3.1.3 查找	70
3.2 双向链表	72
3.3 循环链表	74
3.4 跳转表	75
3.5 自组织表	80
3.6 稀疏表	83
3.7 java.util的链表	85
3.7.1 LinkedList	85
3.7.2 ArrayList	89
3.8 结论	91
3.9 示例学习：图书馆	92
3.10 习题	99
3.11 编程作业	101
参考文献	103
第4章 堆栈和队列	105
4.1 堆栈	105
4.2 队列	111
4.3 优先级队列	117
4.4 示例学习：脱离迷宫	118

4.5 习题	122	6.10 波兰表示法和表达式树	202
4.6 编程作业	124	6.11 示例学习：计算单词频率	206
参考文献	125	6.12 习题	212
第5章 递归	126	6.13 编程作业	214
5.1 递归定义	126	参考文献	217
5.2 方法调用和递归实现	128	第7章 多分树	220
5.3 剖析递归调用	129	7.1 B树家族	220
5.4 尾递归	132	7.1.1 B树	221
5.5 非尾递归	133	7.1.2 B*树	229
5.6 间接递归	137	7.1.3 B*树	230
5.7 嵌套递归	139	7.1.4 前缀B*树	232
5.8 过分递归	139	7.1.5 比特树	233
5.9 回溯	142	7.1.6 R树	235
5.10 小结	147	7.1.7 2-4树	236
5.11 示例学习：递归下降解释器	147	7.1.8 java.util中的树	248
5.12 习题	153	7.2 检索树	257
5.13 编程作业	155	7.3 结论	264
参考文献	157	7.4 示例学习：拼写检查程序	264
第6章 二叉树	158	7.5 习题	273
6.1 树、二叉树和二叉查找树	158	7.6 编程作业	274
6.2 二叉树实现	161	参考文献	277
6.3 搜索二叉查找树	163	第8章 图	279
6.4 树的遍历	164	8.1 图的表示法	280
6.4.1 广度优先遍历	165	8.2 图的遍历	281
6.4.2 深度优先遍历	165	8.3 最短路径	284
6.4.3 无堆栈深度优先遍历	171	8.4 圈检测	291
6.5 插入	175	8.5 生成树	293
6.6 删除	178	8.6 连通性	297
6.6.1 归并删除法	179	8.6.1 无向图的连通性	297
6.6.2 复制删除法	181	8.6.2 有向图的连通性	300
6.7 树的平衡	183	8.7 拓扑排序	302
6.7.1 DSW算法	185	8.8 网络	303
6.7.2 AVL树	187	8.8.1 最大流	303
6.8 自调整树	191	8.8.2 最小代价的最大流量	311
6.8.1 自重构树	192	8.9 匹配	313
6.8.2 伸展树	192	8.9.1 稳定匹配问题	318
6.9 堆	196	8.9.2 分配问题	319
6.9.1 堆作为优先级队列	197	8.9.3 非二部图中的匹配	321
6.9.2 以堆的形式组织数组	199	8.10 欧拉图和哈密顿图	322

8.10.1 欧拉图	322	10.2.2 链	393
8.10.2 哈密顿图	324	10.2.3 桶定址法	394
8.11 图的着色	329	10.3 删除	394
8.12 图论中的NP完整性问题	331	10.4 完全散列函数	395
8.12.1 团问题	331	10.4.1 Cichelli方法	396
8.12.2 3色问题	332	10.4.2 FHCD算法	398
8.12.3 顶点覆盖问题	333	10.5 可扩展文件的散列函数	400
8.12.4 哈密顿回路问题	333	10.5.1 可扩展散列	400
8.13 示例学习：典型代表问题	335	10.5.2 线性散列	402
8.14 习题	336	10.6 java.util中的散列	404
8.15 编程作业	345	10.6.1 HashMap	404
参考文献	346	10.6.2 HashSet	407
第9章 排序	349	10.6.3 Hashtable	410
9.1 基本排序算法	350	10.7 示例学习：桶散列	414
9.1.1 插入排序	350	10.8 习题	421
9.1.2 选择排序	352	10.9 编程作业	422
9.1.3 冒泡排序	353	参考文献	423
9.2 决策树	355	第11章 数据压缩	425
9.3 高效排序算法	357	11.1 数据压缩的条件	425
9.3.1 Shell排序	357	11.2 赫夫曼编码	426
9.3.2 堆排序	360	11.3 顺串长度编码	436
9.3.3 快速排序	363	11.4 Ziv-Lempel编码	437
9.3.4 归并排序	367	11.5 示例学习：结合顺串长度编码的 赫夫曼方法	439
9.3.5 基数排序	370	11.6 习题	448
9.4 java.util中的排序	373	11.7 编程作业	448
9.5 总结	375	参考文献	449
9.6 示例学习：多项式加法	376	第12章 存储管理	451
9.7 习题	383	12.1 顺序适配方法	451
9.8 编程作业	384	12.2 非顺序适配算法	452
参考文献	384	12.3 无用单元收集	459
第10章 散列	387	12.3.1 标记和清除算法	459
10.1 散列函数	387	12.3.2 复制方法	465
10.1.1 除法	387	12.3.3 增量式无用单元收集	466
10.1.2 折叠法	388	12.4 总结	471
10.1.3 平方取中函数	388	12.5 示例学习：内置无用单元收集器	472
10.1.4 提取方法	388	12.6 习题	473
10.1.5 基数变换	388	12.7 编程作业	479
10.2 冲突解决	389	参考文献	481
10.2.1 开放定址法	389		

第13章 字符串匹配	484	13.2.2 k 误配的字符串匹配	524
13.1 精确字符串匹配	484	13.3 示例学习: 最长公共子字符串	526
13.1.1 直接匹配算法	484	13.4 习题	533
13.1.2 Knuth-Morris-Pratt算法	486	13.5 编程作业	535
13.1.3 Boyer-Moore算法	492	参考文献	535
13.1.4 多路查找	500	附录A 大 O 的计算	537
13.1.5 面向位方法	501	A.1 谐波级数	537
13.1.6 词匹配集	504	A.2 函数 $\lg(n!)$ 的近似	537
13.1.7 正则表达式匹配	510	A.3 快速排序平均情况的大 O	538
13.1.8 后缀检索树和树	513	A.4 随机二叉树中的平均路径长度	540
13.1.9 后缀数组	517	A.5 AVL树中的节点数量	541
13.2 近似字符串匹配	518	附录B NP完整性	542
13.2.1 字符串相似度	519	索引	554

第1章 Java语言的面向对象编程

本章向读者介绍Java的基础知识。Java是一个非常好的语言和编程环境，仅在这一章中不可能触及所有和Java相关的问题。这里只介绍用于理解本书中Java代码所必备的知识。熟悉Java语言的读者可以跳过此章。

1.1 Java入门

一个Java程序就是一系列根据预定义语法编写的语句。语句是Java的最小可执行单位，每一条语句都以分号结尾。复合语句，或是程序块，可以利用“{”和“}”将它们圈起来。

1.1.1 变量的声明

使用变量前必须进行声明，这包括指定它的类型和名称。变量名是任意长度的字符串，由字母、数字、下划线和\$组成，而且第一个字符必须是字母、下划线或是\$、不过这里的字母可以是任意大于192的Unicode字符，而不仅仅限于英文的26个字母。局部变量必须被初始化。Java区分大小写，因此变量n和变量N是不同的。

变量的类型可以是8个内置基本类型之一，可以是一个内置或用户自定义的类，也可以是一个数组。以下是内置类型和它们的大小：

类型	大小	取值范围
boolean	1位	true, false
char	16位	Unicode字符
byte	8位	[-128,127]
short	16位	[-32768,32767]
int	32位	[-2147483648,2147483647]
long	64位	[-9223372036854775808,9223372036854775807]
float	32位	[-3.4E38,3.4E38]
double	64位	[-1.7E308,1.7E308]

注意类型的大小是固定的，这对于程序的可移植性是非常重要的。在C/C++中，整型和长整型的大小依赖于系统。Java与C/C++不同，boolean不是一个数值类型，不能对它进行算术运算。但是和C/C++相似的是，字符被看作一个整数（在Java中，看作是无符号整数），因此可以将它们作为操作数进行算术运算。

整数运算采用32位精度（长整数采用64位精度）；所以，对byte和short类型的变量进行运算需要先进行类型转换。例如，语句：

```
byte a,b=1,c=2;
a = b + c;
```

会给出一个编译错误，“等号两边的类型不同，需要使用强制转换将整型（int）转化为字节（byte）类型。”b+c的加法给出一个整型的数值，这必须经过转换后再赋值给字节变量a。为了避免这个问题，赋值语句应当改为：

```
a = (byte) (b + c);
```

一个算术运算引起的溢出（除非是除以0）是不会显示的，编程者必须提防这种情况。对于两个整数，

```
int i = 2147483647, j = i + 1;
```

j的数值是-2147483648。

Java不提供修饰符signed和unsigned，但是它使用了其他的修饰符。

Java和C/C++的一个重要的不同点就是C/C++的字符是8位的，而Java的字符是16位的。通常使用8位只能表示256个不同的字符。为了能表示除英语外其他语言的字符，必须将可用编码集大大地扩展。问题是不仅仅要用可以区分的标志表示字符（例如，波兰字母ń，罗马尼亚字母ț，或是丹麦字母φ）而且还要区分非拉丁字母，如古斯拉夫语、希腊语、日语、中文等等。通过将字符变量扩展为两个字节，可表示的字符数就达到了65 536个。

2

要将指定的Unicode字符赋值给一个字符变量，可以在“u”后面跟上四个十六进制数字，例如：

```
char ch = '\u12ab' ;
```

但是也应当避免使用太高的Unicode编码，因为目前只有少数系统能够显示它们。所以，尽管对ch进行上面的赋值是合法的，但在打印结果时只会显示一个问号。

另一种给字符变量赋直接量值的方法是用单引号把一个字符括起来，

```
ch = 'q' ;
```

以及使用字符转义序列，例如：

```
ch = '\n' ;
```

进行行尾字符的赋值；其他可用的转义字符还有：'\t'（跳格），'\b'（退格），'\r'（回车），'\f'（走纸），'\''（单引号），'\\"（双引号），'\'（反斜杠）。与C/C++不同的是Java没有'\b'（响铃）和'\v'（垂直跳格）。而且，还可以像下面一样使用八进制转义序列'\ddd'：

```
ch = '\123' ; // 十进制中的83，ASCII码中的'S'；
```

其中ddd代表一个八进制数[0, 377]。

整数直接量可以用十进制数中从0~9的数字来表示，

```
int i = 123 ;
```

或是在八进制数的0~7前加0来表示，

```
int j = 0123 ; // 十进制中的83；
```

也可以在十六进制数的0~9和A~F（小写或大写）前加“0x”来表示，

```
int k = 0x123a ; // 十进制中的4666；
```

直接量整数是32位长的，所以，要将它们转化为64位长的数字，必须在数字后加“L”：

```
long p = 0x123aL ;
```

注意应当用大写L而不是小写l，因为后者很容易和数字1混淆。

浮点数小数点的前面和后面是任意的从0~9的数字序列。这个序列可以为空，比如：2.，.2, 1.2。另外，数字后面可以跟着一个字母e以及一系列带符号的数字：4.5e + 6 (= 4.5 × 10⁶ = 4500000.0)，102.055e - 3 = 102.055 × 10⁻³ = .102055)。浮点数直接量默认为64位数值；所以，以下声明和赋值

3

```
float x = 123.45 ;
```


会导致一个编译错误，“声明的类型不兼容。需要将双精度数强制转换为浮点数”，在数字的末尾添加一个修饰符f（或是F）可以消除这个错误，

```
float x = 123.45f ;
```

双精度数的后面可以添加修饰符d或是D，不过这可以省略。

1.1.2 运算符

变量的赋值是通过赋值运算符“=”进行的，一次可以使用一个赋值符，也可以和其他的赋值运算符串在一起使用，例如

```
x = y = z = 1 ;
```

这意味着三个变量都被赋予了相同的值1。当赋值运算符两边变量相同时，Java可以使用简写方式的运算符；例如，

```
x = x + 1;
```

可以缩短为

```
x += 1;
```

Java也可以使用前缀形式或后缀形式的自加和自减运算符，如++n, n++, --n和n--，它们分别是对语句n=n+1和n=n-1的简写，这里的n可以是任意数，包括浮点数。前缀运算符和后缀运算符的不同之处是，对于前缀运算符，变量先进行加法（或减法）然后再执行加法处的其他操作。而对于后缀运算符，自加（或自减）是最后一个被执行的操作；例如，执行下列语句：

```
x = 5;
```

```
y = 6 + ++x;
```

y等于12，而执行以下语句后

```
x = 5;
```

```
y = 6 + x++;
```

y等于11。在这两个例子中，执行完第二条语句后x都等于6。

Java允许对独立的位使用位逻辑运算符进行操作：&（位逻辑与）、|（位逻辑或）、^（位逻辑异或）、<<（左移）、>>（右移）、>>>（补0右移）、~（按位反）。也可以用简写运算符&=、|=、^=、<<=、>>=和>>>=。除了运算符>>>外，其他的运算符在C/C++中也有。运算符>>从右边（最低位）移出指定数目的位，并且如果是正数就移入相同数量的0，如果是负数就移入相同数量的1。例如，下列语句执行后

```
int n = -4;
```

```
int m = n >>1;
```

m的数值为-1，这是因为n中-4的二进制补码是一个32位序列11...1100，右移一位后赋给m的值是11...1110，它是-2的二进制补码。为了给负数也移入0，必须使用>>>运算符，

```
int n = -4;
```

```
int m = n >>>1;
```

这样，n中的11...1100就转化为01...1110赋给m，这代表2147483646（正数的最大上限减1）。

1.1.3 选择语句

第一种选择语句是if语句