

世界著名计算机教材精选

PEARSON
Prentice
Hall

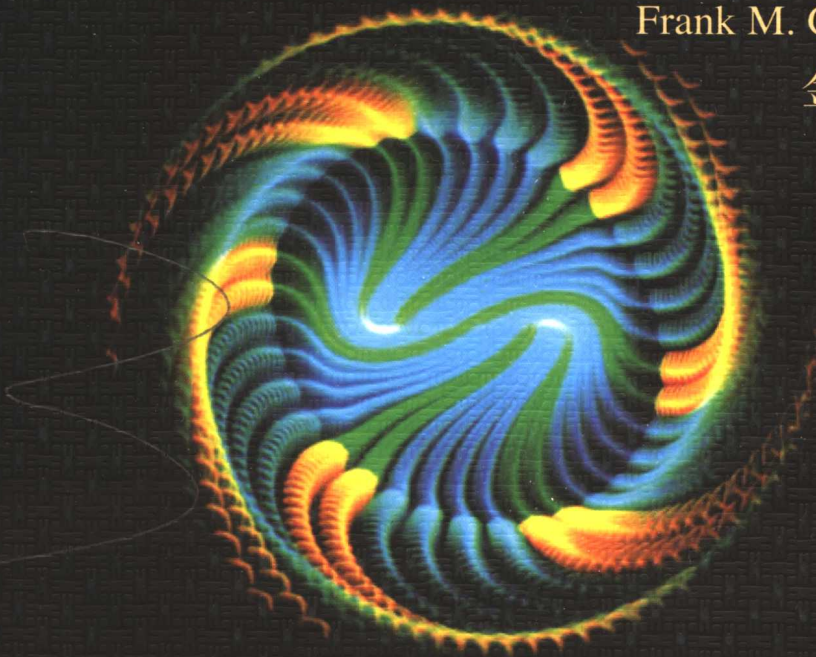
数据结构 与算法分析

Java语言描述 (第2版)

Frank M. Carrano

著
等译

金名



**DATA STRUCTURES
AND ABSTRACTIONS WITH JAVA**

Second Edition

PEARSON
Education

清华大学出版社

TP311.12/156

2007

世界著名计算机教材精选

数据结构与算法分析 (Java 语言描述)

(第2版)

Frank M. Carrano 著

金名等译

清华大学出版社

北京

Simplified Chinese edition copyright © 2007 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Data Structures and Abstractions with Java, 2E by Frank M. Carrano, Copyright © 2007

EISBN: 0-13-237045-X

All Rights Reserved.

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

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Prentice-Hall, Inc. 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字:01-2007-2037 号

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

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。

图书在版编目(CIP)数据

数据结构与算法分析(Java语言描述)(第2版)/(美)卡拉罗(Carrano, F. M.)著;金名等译. —北京:清华大学出版社,2007.12

(世界著名计算机教材精选)

书名原文: Data Structures and Abstractions with Java, 2E

ISBN 978-7-302-16269-8

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

中国版本图书馆CIP数据核字(2007)第155865号

责任编辑:龙啟铭

责任校对:张 剑

责任印制:孟凡玉

出版发行:清华大学出版社

<http://www.tup.com.cn>

c-service@tup.tsinghua.edu.cn

社总机:010-62770175

投稿咨询:010-62772015

地 址:北京清华大学学研大厦A座

邮 编:100084

邮购热线:010-62786544

客户服务:010-62776969

印 刷 者:北京密云胶印厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:56 字 数:1357千字

版 次:2007年12月第1版 印 次:2007年12月第1次印刷

印 数:1~3000

定 价:98.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:021888-01

“数据结构”是计算机专业的基础与核心课程之一,也是从事软件开发必不可少的入门和常用知识。程序编写得好不好,很大程度上取决于编程者对数据结构是否熟练地掌握和恰当地运用。由于它不仅重要,而且易学难精,“数据结构”一直被列入相关专业的研究生入学考试和相关行业的公司招聘考试的重点考查范围。

由于“数据结构”这门课程本身的特点,它必须依托于一种程序设计语言才能讲授,否则就成了空中楼阁、纸上谈兵。因此,尽管从抽象和逻辑的角度看来都大同小异,按照所依托的程序设计语言可以把“数据结构”的教材分为不同的版本——诸如 Pascal 版、C 版、C++ 版以及 Java 版。除了由于程序设计语言的不同特性而导致的程序实现上的差异,不同版本的“数据结构”教材所讲述的主要内容并无本质区别。因此,初学者可以根据自己已经掌握的或者将作为主要使用的程序设计语言选择相应版本的“数据结构”教材来学习。将来如果换用另一种程序设计语言,也不需要重新学习另一个版本的“数据结构”教材,只需将其作为参考,查阅同样的数据结构是如何用另一种语言实现的即可。这也是为什么不同版本的“数据结构”教材都有其存在的意义。

近年来,Java 越来越成为一种“热门”的语言。它不但是目前业界进行软件开发的主要语言之一,而且被国外不少大学选用为向计算机专业的学生传授的第一种程序设计语言。这固然除了 Java 语言本身的优点之外,也有商业利益和教育观念上的考虑,在此不作赘述。但本书并不是国内第一本基于 Java 的“数据结构”教材。之所以选择本书翻译并介绍给国内读者,主要是因为本书有以下几个独特的优点:

- 总的来说,数组和链表是数据结构的两种基本的实现方式。基于这两种方式,可以实现线性表、栈、队列、二叉树、树、图等从简单到复杂的数据结构及其各种变体。本书的内容涵盖了以上各个方面,以及对算法和算法复杂性的一些初步的介绍,从而构成了一本完整的数据结构教材。但本书不限于此,还在前 3 章和附录 A~D 中全面地介绍了 Java 语言,在某些方面其详尽程度甚至超过专门讲述 Java 语言的书籍。因此,即使未学过 Java 语言的读者,也可以将本书作为起点,在学习 Java 语言的同时学习数据结构。
- 本书在编写过程中特别考虑到了面向对象程序设计(OOP)的思想与 Java 语言的特性。它不是从基于另一种程序设计语言的数据结构教材简单地“改编”而来的,因此在数据结构的实现上更加“地道”地运用了 Java 语言,并且自始至终强调以面向对象的方式来思考、分析和解决问题。
- 本书在内容编排上,将每一种数据结构的说明与实现分开来,全书被划分为 31 个相对较短的章,从而为读者提供了阅读顺序的机动性。读者可以在阅读了一个数据结构的说明之后紧接着就学习它是如何用 Java 语言实现的,也可以首先通读各种数据结构的说明,然后才

考虑如何实现它们。

- 本书论述详细,深入浅出,含有大量的图示、应用示例和源程序,每章还配有自测题以及答案(在附录 E 中),不但可作为大专院校相关专业本科和研究生的教材,也非常适宜自学,以及作为开发软件时案头常备的参考书。

由于一些术语在国内存在不止一种译法,本书在翻译时只能尽量选用沿袭较广的译法,如果读者已经熟悉的是另一种译法,阅读时可能会感到不便,在此表示抱歉。书中的术语在第一次出现时均列出了相应的英文,供读者对照辨识。

本书的翻译出版是一个专家团队共同努力的结果。本书由金名主译,参加本书翻译工作的还有张长富、李伟、郭涛、高磊、王振营、冯哲、韩毅、马以辉、李腾、邓卫、邓凡平、周云、董武、郑晓蕊、陈占军、倪泳智、黄虹、吕巧珍、裘蕾、金颖、陈河南、王嘉佳、吴建伟、宋雁、何晓刚、段涛、马丽娟、郭翔、朱晓林、陈磊、李建锋、刘延军、刘子瑛、徐英武、魏宇、赵远锋、樊旭平、戴军、梁德成、梁彩隆、倪泳智、赵成璧、韦笑、李志云、余春、陈强、董岚枫、郭涛、周京平、王炯、程烨尔、徐冬、唐玮、贺军等。真诚地欢迎读者对书中可能有的错误和不妥之处批评指正,读者的意见是我们改进的动力和依据。

欢迎阅读《数据结构与算法分析(Java 语言描述)》的第 2 版。本书是作为数据结构入门课程的教科书而编写的。

本书的编写特别考虑了对象与 Java。为了使学习更容易,我们采用的方式是:将读者的注意力一次集中在一个问题上,为读者提供阅读顺序的灵活性,以及明确区分抽象数据类型(ADT)的说明与实现。为了达到这些目的,我们将内容划分成 31 个相对较短的章,在每一章中或是论述 ADT 的说明及使用,或是论述 ADT 的实现,读者可以选择在阅读了一个 ADT 的说明之后紧跟着学习它的各种实现方法,也可以选择考虑任何实现问题之前先探讨各种 ADT 的说明及使用。从本书的编写结构上讲,读者可以方便地选择自己偏爱的阅读顺序来学习。

第 2 版的新内容

- 涵盖和使用了 Java 5 的新特性,包括一般数据类型,如 Scanner 类、枚举、for-each 循环、Iterable 接口以及装箱与拆箱等。
- 用 assert 语句执行代码中的断言。
- 用两章而非一章来介绍 ADT 线性表的链表实现。可作为从关注核心方法开始逐步实现一个类的示例。
- 用一章而非两章来介绍迭代器。内容涵盖了新接口 Iterable,第 1 版中的非标准迭代器则不再介绍。
- 用两章而非一章来介绍散列。
- 改进了各种 ADT 的实现,尤其是线性表和二叉查找树的链表实现。
- 重要的 Java 代码被标识在一份列表中。
- 扩充和改进了练习、项目与自测题。
- 附录 B“异常处理”中增加了 finally 子句。
- 附录 C“文件输入与输出”做了修改以反映 Java 5 的特性。

特点

- 31 个相对短的章可以按各种顺序阅读。
- 单独但相关的章将 ADT 的说明与实现分开。
- 用很多例子说明新的概念。
- 突出的“注”强调了关键的内容并提供补充注释。
- “编程提示”给出附加的编程建议。
- 大量的插图使讲解更形象,更易于理解。
- 贯穿全书的自测题及其答案均是根据本书内容精心制作的。
- 前几章的内容覆盖了 Java 类、继承、多态性及类的设计。
- Java 代码包含 javadoc 注释。
- 附录复习了 Java 基础、异常、文件及文档。
- 网站 prenhall.com/carrano 提供本书中出现的 Java 代码以及任何更新与修正。
- 为教师提供的补充材料包括 PowerPoint 幻灯片、实验素材以及经过挑选的部分练习与项目设计的答案。

学生注意事项

在看完这个前言之后,读者应该先读引言。在那里你将迅速了解到本书要讲述的内容,以及在开始学习前需要对 Java 了解多少。从附录 A~D 中复习了 Java 的基础知识,并可作为关于异常、文件、javadoc 注释以及 Java 5 新特性的初级读物。注意,本书还给出了 Java 的保留字、基本数据类型、运算符优先次序以及 Unicode 字符列表。

本书涵盖的主题是读者今后学习计算机科学的基础。即使现在没有学完所有主题,以后也很可能会遇到它们。我们希望读者现在爱读本书,并希望它成为你学习将来课程用得着的参考书。

在每一章中,读者都能找到有助于理解内容的自测题,这些问题的答案给出在本书后的附录 E 中。然而,我们不能给出每一章末的练习与项目设计的答案,采用本书的教师能够从出版社收到精选的答案。未选这门课的读者就不能得到该答案了。关于这些练习与项目设计的帮助,读者必须与自己的教师联系。

本书中的所有 Java 代码均可从网站 prenhall.com/carrano 下载。

本书概述

本书的读者应该已经学完一门程序设计的课程,最好是关于 Java 程序设计的。附录 A 涵盖了假设读者已学习过的 Java 基础,读者可以用这个附录作为复习,或者作为从另一种程序设计语言过渡到 Java 的基础。读者还可以从附录中学到本书中用到的 Java 5 的一些新特性。

本书自身是从引言开始的,引言为以后的学习做准备,第 1 章全面地复习 Java 中的类与方法,紧接着在第 2 章中讨论合成、继承及多态性,这一章还介绍了 Java 的一种重要的新特性——一般数据类型。第 3 章引入面向对象设计,介绍 Java 接口,并初步介绍如 CRC 卡片与统一建模语言(Unified Modeling Language)这样的设计工具。

第 4 章~第 7 章引入一种抽象数据类型的线性表。我们把内容分成几章来讲,就能清楚地将线性表的说明、使用及实现分开。例如,第 4 章说明线性表,并列举几个使用示例,第 5 章介绍使用数组和向量的实现,第 6 章与第 7 章则讨论使用链表的实现。

在讨论其他各种 ADT 时,本书自始至终以类似的方式将说明与实现分开。读者可以先阅读所有说明与使用 ADT 的章,之后再阅读实现它们的各章,或者按各章出现的顺序阅读,每个 ADT 的实现都紧跟在学习它的说明与使用之后。

第 6 章和第 7 章不仅实现了线性表,还展示了如何逐步实现一个类。第 6 章从关注核心方法开始 ADT 线性表的链表实现。在定义一个类时,通常有效的做法是首先实现并测试核心方法,而把其余方法的定义留到以后。接下来的第 7 章则完成线性表的链表实现,并且探索这种方式的一些变体。

第 8 章以线性表为背景讨论迭代器,考察并实现 Java 的迭代器接口 `Iterator` 与 `ListIterator`。这一章还将介绍 `Iterable` 接口。

第 9 章与第 10 章介绍算法的复杂性与递归,这是两个在后续各章中要结合讨论的主题。第 11 章与第 12 章讨论各种排序方法与它们的相对复杂度,并同时考察这些算法的迭代版本与递归版本。

接下来的第 13 章与第 14 章,又回到线性表的概念。第 13 章讨论有序表,考察两种可能的实现及其效率。第 14 章介绍如何使用线性表作为有序表的基类,并讨论基类的一般设

计要求。

第 15 章介绍可变对象、不可变对象及克隆。如果客户可以维护 ADT 中数据的引用,如果数据是可变的,则它不使用 ADT 类的公有方法就能够修改此数据。我们将考察可用来防止客户这样做的步骤。

第 16 章以线性表与有序表为背景,考察在数组与链表中查找的策略。这些讨论是第 17 章的良好基础,后者涵盖 ADT 词典的说明与使用。第 18 章将介绍链表或数组的词典实现。第 19 章将介绍散列,第 20 章则介绍用散列实现词典。

第 21 章讨论栈,列举使用栈的例子,并考察栈与递归间的关系。第 22 章分别使用数组、向量及链表实现栈。

第 23 章介绍队列、双端队列及优先队列,第 24 章则考察它们的实现。在这一章中还讨论循环链表与双向链表。

第 25 章讨论树及其应用场合。在树的几个例子中,包括对二叉查找树与堆的初步介绍。第 26 章考察二叉树与通用树的实现,第 27 章则着重讨论二叉查找树的实现。第 28 章介绍如何用数组实现堆。第 29 章介绍平衡查找树,在这一章中包括 AVL 树、2-3 树、2-4 树及红黑树,还有 B 树。

第 30 章与第 31 章讨论图,考察图的几种应用及两种实现。

附录 A~附录 D 将提供有关 Java 的补充材料。如前所述,附录 A 复习 Java 除类之外的基本内容,还介绍 Java 5 的一些新特性,包括 Scanner 类,枚举,打包与拆包,以及 for-each 循环。附录 B 涵盖异常处理,附录 C 则讨论文件,附录 D 考察编程风格与注释,它介绍 javadoc 注释,并定义本书中使用的标记。

附录 E 包含全书各章的自测题答案。

教师使用的资源

我们为教师准备了若干补充材料,其中包括 PowerPoint 幻灯片、实验素材、精选的练习与项目设计的答案以及书中的 Java 代码的完整版本。在本书中经常把实现的一些方面留给读者完成,但提供给教师的代码是完整的。注意,书中出现的代码在网站 prenhall.com/carrano 上也可获得。该网站还包含了附加的资源以及本书的所有更新与修正。

要获得上述教辅资源,应该与 Prentice Hall 的销售代表联系,关于销售代表的姓名与 E-mail 地址,请访问网站 prenhall.com/relocator 或者致电 Prentice Hall 高校服务 1-800-526-0485。有关本书及其他 Prentice Hall 产品的附加信息,也可在 Prentice Hall 的主页 prenhall.com 上找到。

联系方法

读者的评论、建议及更正永远受到欢迎。请发 E-mail 到:
carrano@acm.org

致谢

感谢下列审稿人认真地阅读我们的手稿,并且做出中肯的评论与建议,这极大地改善了我们的著作:

Harold Anderson—Marist College

Razvan Andonie—Central Washington University

Tom Blough—Rensselaer Polytechnic Institute
 Chris Brooks—University of San Francisco
 Adrienne Decker—University at Buffalo, SUNY
 Henry Etlinger—Rochester Institute of Technology
 Derek Harter—Texas A&M University
 Timothy Henry—University of Rhode Island
 Robert Holloway—University of Wisconsin, Madison
 Charles Hoot—Oklahoma City University
 Teresa Leyk—Texas A&M University
 Robert McGlinn—Southern Illinois University, Carbondale
 Edward Medvid—Marymount University
 Charles Metzler—City College of San Francisco
 Daniel Zeng—University of Arizona
 再次感谢本书第 1 版的审稿人：
 David Boyd—Valdosta State University
 Dennis Brylow—Purdue University
 Michael Croswell—Industry trainer/consultant
 Matthew Dickerson—Middlebury College
 Robert Holloway—University of Wisconsin, Madison
 John Motil—California State University, Northridge
 Bina Ramamurthy—University at Buffalo, SUNY
 David Surma—Valparaiso University

特别感谢 Rebecca Pepper, 我们的文稿编辑, 她不但纠正了我们的语法错误, 还保证了我们的讲解清楚无误。感谢 Charles Hoot, 他提供了附加的练习与项目设计并准备了一些给教师的补充材料。感谢 Lianne Dunn, 她设计了封面图案, 并且绘制了本书第一版中所有的插图。感谢 Steve Armstrong, 他准备了 PowerPoint 幻灯片。

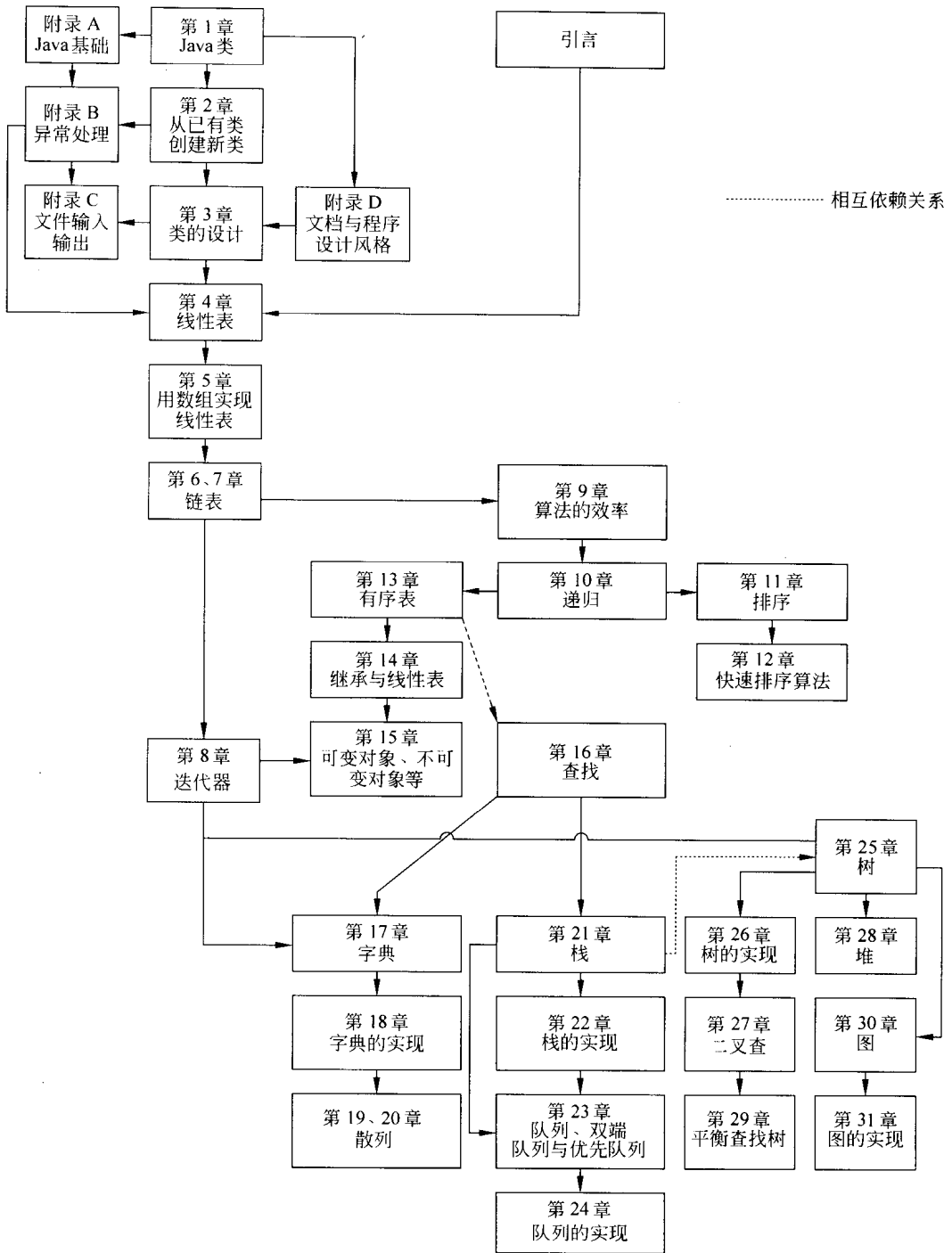
感谢 Prentice Hall 出版社为本书以及补充材料的制作与出版做出贡献的所有人。特别要感谢我们的编辑 Tracy Dunkelberger, 还有 Carole Snyder, Christianna Lee, Camille Trentacoste, Vince O'Brien, Xiaohong Zhu, Mike Giacobbe, John Lovell。对他们所付出的辛劳深表感谢。还要特别感谢我们的出版编辑 Rose Kernan 使我们的成果得以付梓。

尤其要感谢我的合作者 Walt Savitch 和我的前编辑 Alan Apt 与 Toni Holm 对本书的两个版本的贡献。还有许多人也为本书第一版提供了帮助, 他们是 Steve Armstrong, James Blanding, Charles Hoot, Brian Jepson, Patrick Lindner, Patty Roy, Ben Schomp, Heather Scott, Chirag Thakkar, Nate Walker, Xiaohong Zhu。

其他热心的人士以各种方式做出了贡献。他们是: Doug McCreddie, Ted Emmott, Lorraine Berube, Marge White, Tom Manning, Jeff Barbeau, Joan Peckham, James Kowalski, Ed Lamagna, Victor Fay-Wolfe, Gérard Baudet, Tim Henry, Lisa DiPippo, Jean-Yves Hervé, James Heltshe, Bala Ravikumar。

感谢各位的贡献以及鼓励。

Frank M. Carrano



目 录

第0章 引言	1	第3章 类的设计	70
第1章 Java类	2	3.1 封装	70
1.1 对象与类	2	3.2 方法的说明	72
1.2 在Java类中使用方法	5	3.3 接口	76
1.3 定义Java类	7	3.3.1 编写接口	76
1.3.1 方法定义	8	3.3.2 实现接口	78
1.3.2 实参与形参	10	3.3.3 作为数据类型的 接口	79
1.3.3 传递实参	11	3.3.4 接口的通用类型	80
1.3.4 Name类的定义	14	3.3.5 Comparable接口	80
1.3.5 构造函数	16	3.3.6 扩展接口	82
1.3.6 toString方法	18	3.3.7 接口与抽象类	83
1.3.7 调用其他方法的方法	18	3.3.8 符号常量	85
1.3.8 返回所属类实例 的方法	20	3.4 类的选择	86
1.3.9 静态域与静态方法	20	3.4.1 类的确定	87
1.3.10 方法的重载	22	3.4.2 CRC卡片	88
1.4 枚举类	23	3.5 类的复用	91
1.5 包	26	本章小结	91
本章小结	27	练习	92
练习	28	项目设计	93
项目设计	31	第4章 线性表	96
第2章 从已有类创建新类	35	4.1 ADT线性表说明	96
2.1 合成	35	4.2 使用ADT线性表	103
2.1.1 通用类型	38	4.3 像使用自动售货机一样使用 线性表	107
2.1.2 适配器	41	4.4 Java类库:List接口	109
2.2 继承	42	本章小结	109
2.2.1 从构造函数中调用构造 函数	45	练习	110
2.2.2 基类的私有域与私有 方法	46	项目设计	112
2.2.3 受保护的访问	47	第5章 用数组实现线性表	114
2.2.4 方法的覆盖与重载	47	5.1 使用定长数组实现ADT 线性表	114
2.2.5 多重继承	52	5.1.1 类比	114
2.3 类型兼容性与基类	53	5.1.2 Java实现	116
2.3.1 Object类	54	5.2 使用动态扩展数组实现 ADT线性表	124
2.3.2 抽象类与抽象方法	56	5.2.1 扩展数组	125
2.4 多态性	58	5.2.2 线性表的新实现	127
本章小结	63	5.3 Java类库:ArrayList与 Vector类	128
练习	64		
项目设计	68		

5.4 用数组实现 ADT 线性表的优缺点	132	8.2 Iterator 接口	180
本章小结	133	8.3 独立类迭代器	186
练习	134	8.4 内部类迭代器	189
项目设计	135	8.4.1 基于链表实现	190
		8.4.2 基于数组实现	194
第 6 章 用链表实现线性表	136	8.5 迭代器方法为何在自己的类中	197
6.1 链表	136	8.6 ListIterator 接口	198
6.1.1 在表头添加来创建链表	137	8.7 基于数组实现 ListIterator 接口	204
6.1.2 在表末添加来创建链表	138	8.8 Java 类库:Iterable 接口	211
6.1.3 在不同位置添加来创建链表	140	8.8.1 Iterable 与 for-each 循环	212
6.2 使用链表实现 ADT 线性表	142	8.8.2 重温 List 接口	212
6.2.1 私有类 Node	142	本章小结	213
6.2.2 数据域与构造函数	144	练习	213
6.2.3 选择要实现的核心方法组	145	项目设计	216
6.2.4 在线性表的末端插入元素	146		
6.2.5 在线性表的指定位置插入元素	148	第 9 章 算法的效率	218
6.2.6 私有方法 getNodeAt	152	9.1 动机	218
6.2.7 断言与 isEmpty 方法	153	9.2 度量算法的效率	220
6.2.8 display 方法	154	9.3 形式化	226
6.3 测试不完整的实现	155	9.4 效率的图形表示	229
本章小结	157	9.5 ADT 线性表不同实现的效率	232
练习	158	9.5.1 基于数组实现	232
项目设计	159	9.5.2 基于链表实现	234
		9.5.3 比较上述实现	237
		本章小结	238
第 7 章 完成线性表的链表实现	160	练习	238
7.1 从链表中删除一个元素	160	项目设计	240
7.2 完成 ADT 线性表的链表实现	162		
7.2.1 方法 remove	162	第 10 章 递归	243
7.2.2 方法 replace	165	10.1 何谓递归	243
7.2.3 方法 getEntry	165	10.2 跟踪递归方法	248
7.2.4 方法 contains	166	10.3 有返回值的递归方法	250
7.2.5 其他方法	166	10.4 递归处理数组	253
7.3 使用具有设置与获取方法的 Node 类	167	10.5 递归处理链表	255
7.4 表尾引用	170	10.6 递归方法的时间效率	257
7.5 用链表实现 ADT 线性表的优缺点	175	10.6.1 countDown 的时间效率	257
7.6 Java 类库:LinkedList 类	175	10.6.2 计算 x^n 的时间效率	258
本章小结	176	10.7 困难问题的简单解法	259
练习	176	10.8 简单问题的拙劣解法	264
项目设计	177	10.9 尾递归	266
		10.10 协同递归	268
		本章小结	268
第 8 章 迭代器	179	练习	270
8.1 什么是迭代器	179	项目设计	271

第 11 章 排序入门	275	13.2.2 链表实现的效率	331
11.1 组织用于数组排序的 Java 方法	276	13.3 使用 ADT 线性表的实现	331
11.2 选择排序	277	本章小结	336
11.2.1 迭代选择排序	278	练习	336
11.2.2 递归选择排序	280	项目设计	337
11.2.3 选择排序的效率	281	第 14 章 继承与线性表	339
11.3 插入排序	282	14.1 使用继承实现有序表	339
11.3.1 迭代插入排序	283	14.2 设计一个基类	342
11.3.2 递归插入排序	284	14.3 有序表的一种高效实现	348
11.3.3 插入排序的效率	286	本章小结	349
11.3.4 链表的插入排序	286	练习	350
11.4 希尔排序	289	项目设计	350
11.4.1 Java 代码	291	第 15 章 可变对象、不可变对象与可克隆	
11.4.2 希尔排序的效率	292	对象	352
11.5 算法比较	293	15.1 可变对象与不可变对象	352
本章小结	293	15.1.1 创建只读类	355
练习	293	15.1.2 同伴类	356
项目设计	296	15.2 可克隆对象	358
第 12 章 快速排序算法	298	15.2.1 克隆数组	364
12.1 归并排序	298	15.2.2 克隆链表	366
12.1.1 数组的归并	298	15.2.3 克隆体的有序表	370
12.1.2 递归归并排序	299	本章小结	373
12.1.3 归并排序的效率	302	练习	373
12.1.4 迭代归并排序	303	项目设计	376
12.1.5 Java 类库中的归并排序	304	第 16 章 查找	377
12.2 快速排序	304	16.1 问题描述	377
12.2.1 快速排序的效率	305	16.2 查找无序数组	378
12.2.2 创建划分	305	16.2.1 迭代顺序查找无序数组	378
12.2.3 快速排序的 Java 代码	308	16.2.2 递归顺序查找无序数组	379
12.2.4 Java 类库中的快速排序	311	16.2.3 顺序查找数组的效率	381
12.3 基数排序	311	16.3 查找有序数组	381
12.3.1 基数排序的伪代码	313	16.3.1 顺序查找有序数组	381
12.3.2 基数排序的效率	313	16.3.2 折半查找有序数组	382
12.4 算法比较	314	16.3.3 Java 类库: binarySearch	
本章小结	314	方法	386
练习	315	16.3.4 折半查找数组的效率	387
项目设计	316	16.4 查找无序链表	388
第 13 章 有序表	319	16.4.1 迭代顺序查找无序链表	388
13.1 ADT 有序表的说明	319	16.4.2 递归顺序查找无序链表	389
13.2 链表实现	323	16.4.3 顺序查找链表的效率	390
13.2.1 add 方法	324		

16.5 查找有序链表	390	19.3.4 开放定址的潜在问题	453
16.5.1 顺序查找有序链表	390	19.3.5 链地址	453
16.5.2 折半查找有序链表	391	本章小结	456
16.6 查找方法的选择	391	练习	457
本章小结	392	项目设计	458
练习	392		
项目设计	394		
第 17 章 词典	396	第 20 章 用散列实现词典	459
17.1 ADT 词典的说明	396	20.1 效率	459
17.1.1 Java 接口	399	20.1.1 装填因子	460
17.1.2 迭代器	400	20.1.2 开放定址的开销	460
17.2 使用 ADT 词典	402	20.1.3 链地址的开销	462
17.2.1 电话号码簿	402	20.2 再散列	463
17.2.2 词频	407	20.3 处理冲突的各方案比较	464
17.2.3 词的索引	411	20.4 使用散列的词典实现	465
17.3 Java 类库:Map 接口	414	20.4.1 散列表中的元素	465
本章小结	415	20.4.2 数据域与构造函数	466
练习	415	20.4.3 方法 getValue、remove 和 add	467
项目设计	416	20.4.4 迭代器	473
		20.5 Java 类库:类 HashMap	475
		本章小结	475
		练习	476
		项目设计	476
第 18 章 词典的实现	418	第 21 章 栈	478
18.1 基于数组的实现	418	21.1 ADT 栈的说明	478
18.1.1 基于数组的无序词典	419	21.2 利用栈处理代数表达式	481
18.1.2 基于数组的有序词典	424	21.2.1 检查中缀代数表达式的括号 是否平衡	482
18.2 基于向量的实现	428	21.2.2 将中缀表达式转化为后缀 表达式	487
18.3 基于链表的实现	433	21.2.3 后缀表达式求值	493
18.3.1 基于链表的无序词典	434	21.2.4 中缀表达式求值	495
18.3.2 基于链表的有序词典	434	21.3 程序栈	497
本章小结	438	21.4 使用栈代替递归	498
练习	438	21.5 Java 类库:类 Stack	501
项目设计	439	本章小结	502
		练习	502
		项目设计	504
第 19 章 散列概述	440	第 22 章 栈的实现	506
19.1 什么是散列	440	22.1 基于链表的实现	506
19.2 散列函数	442	22.2 基于数组的实现	509
19.2.1 计算散列码	443		
19.2.2 将散列码压缩为散列表的 索引	445		
19.3 处理冲突	446		
19.3.1 线性探测开放定址	446		
19.3.2 二次探测开放定址	451		
19.3.3 双散列开放定址	451		

22.3 基于向量的实现	513	25.2.2 树的遍历	576
本章小结	515	25.3 树的Java接口	577
练习	515	25.3.1 所有树的接口	577
项目设计	516	25.3.2 二叉树的接口	578
第23章 队列、双端队列与优先队列	517	25.4 二叉树举例	579
23.1 ADT队列的描述	517	25.4.1 表达式树	580
23.2 使用队列模拟排队	521	25.4.2 决策树	581
23.3 使用队列计算股份销售的资本 收益	527	25.4.3 二叉查找树	585
23.4 Java类库:Queue接口	530	25.4.4 堆	587
23.5 ADT双端队列的描述	530	25.5 树举例	589
23.6 使用双端队列计算股份销售的资本 收益	532	25.5.1 语法分析树	589
23.7 ADT优先队列的描述	534	25.5.2 博弈树	590
23.8 使用优先队列跟踪委派任务	535	本章小结	590
本章小结	537	练习	591
练习	537	项目设计	593
项目设计	539	第26章 树的实现	595
第24章 队列、双端队列与优先队列的 实现	541	26.1 二叉树的结点	595
24.1 基于链表的队列实现	541	26.1.1 结点的接口	596
24.2 基于数组的队列实现	545	26.1.2 BinaryNode的实现	597
24.2.1 循环数组	546	26.2 ADT二叉树的实现	599
24.2.2 含有一个未用位置的循环 数组	547	26.2.1 创建基本二叉树	599
24.3 基于向量的队列实现	552	26.2.2 方法privateSetTree	600
24.4 基于循环链表的队列实现	554	26.2.3 访问者与修改者方法	603
24.5 Java类库:AbstractQueue类	560	26.2.4 计算高度与统计结点	604
24.6 基于双向链表的双端队列实现	560	26.2.5 遍历	605
24.7 实现优先队列的可用方法	564	26.3 表达式二叉树的实现	610
24.8 Java类库:PriorityQueue类	564	26.4 树	612
本章小结	565	26.4.1 树的结点	612
练习	566	26.4.2 用二叉树表示树	613
项目设计	567	本章小结	614
第25章 树	569	练习	614
25.1 树的概念	569	项目设计	616
25.1.1 层次化的组织	569	第27章 二叉查找树的实现	617
25.1.2 树的术语	571	27.1 预备知识	617
25.2 树的遍历	574	27.1.1 二叉查找树接口	618
25.2.1 二叉树的遍历	575	27.1.2 重复元素	620
		27.1.3 开始类定义	621
		27.2 查找与检索	622
		27.3 遍历	624
		27.4 插入元素	624
		27.4.1 递归实现	625

27.4.2 迭代实现	628	29.4.1 红黑树的特性	688
27.5 删除元素	631	29.4.2 往红黑树插入元素	689
27.5.1 删除叶子结点中的元素	631	29.4.3 Java 类库:类 TreeMap	693
27.5.2 删除有一个孩子的结点中的 元素	631	29.5 B 树	693
27.5.3 删除有两个孩子的结点中的 元素	631	本章小结	694
27.5.4 删除根结点中的元素	635	练习	695
27.5.5 递归实现	635	项目设计	696
27.5.6 迭代实现	639	第 30 章 图	697
27.6 操作的效率	643	30.1 一些例子与术语	697
27.6.1 平衡的重要性	644	30.1.1 公路地图	697
27.6.2 插入结点的顺序	644	30.1.2 航线	700
27.7 ADT 词典的实现	645	30.1.3 迷宫	700
本章小结	648	30.1.4 先修课程	701
练习	649	30.1.5 树	701
项目设计	651	30.2 遍历	701
第 28 章 堆的实现	653	30.2.1 广度优先遍历	702
28.1 再论 ADT 堆	653	30.2.2 深度优先遍历	704
28.2 用数组表示堆	654	30.3 拓扑顺序	705
28.3 插入元素	656	30.4 路径	707
28.4 删除根	659	30.4.1 寻找路径	708
28.5 创建堆	662	30.4.2 无权图的最短路径	708
28.6 堆排序	664	30.4.3 加权图的最短路径	710
本章小结	667	30.5 ADT 图的 Java 接口	714
练习	668	本章小结	717
项目设计	669	练习	718
第 29 章 平衡查找树	670	项目设计	720
29.1 AVL 树	670	第 31 章 图的实现	722
29.1.1 单旋转	671	31.1 两种实现的概述	722
29.1.2 双旋转	673	31.1.1 邻接矩阵	722
29.1.3 实现细节	676	31.1.2 邻接表	723
29.2 2-3 树	681	31.2 顶点与边	724
29.2.1 2-3 树的查找	681	31.2.1 说明类 Vertex	724
29.2.2 往 2-3 树插入元素	682	31.2.2 内部类 Edge	726
29.2.3 插入期间分裂结点	683	31.2.3 实现 Vertex 类	727
29.3 2-4 树	685	31.3 ADT 图的实现	731
29.3.1 往 2-4 树插入元素	685	31.3.1 基本操作	731
29.3.2 比较 AVL 树、2-3 树 和 2-4 树	686	31.3.2 图的算法	735
29.4 红黑树	687	本章小结	737

附录 A Java 基础	741	A.11 使用 Scanner 抽取字符串的 一部分	780
A.1 引言	741	A.12 数组	783
A.1.1 应用程序和小程序	741	A.12.1 数组形参和返回值	785
A.1.2 对象和类	741	A.12.2 初始化数组	786
A.1.3 第一个 Java 应用程序	741	A.12.3 数组索引出界	786
A.2 Java 基础	744	A.12.4 对数组使用=与==	786
A.2.1 标识符	744	A.12.5 数组与 for-each 循环	788
A.2.2 保留字	744	A.12.6 多维数组	788
A.2.3 变量	745	A.12 封装类	790
A.2.4 基本类型	745	附录 B 异常处理	796
A.2.5 常量	746	B.1 基本的异常处理	796
A.2.6 赋值语句	746	B.2 Java 类库的异常类	799
A.2.7 赋值兼容性	747	B.3 定义自己的异常类	800
A.2.8 类型转换	748	B.4 复合 catch 代码块	802
A.2.9 算术运算符和表达式	749	B.5 finally 代码块	803
A.2.10 括号和优先规则	750	B.6 抛出但不捕获异常的方法	804
A.2.11 自增和自减运算符	751	B.7 不需要捕获的异常	805
A.2.12 特殊赋值运算符	752	附录 C 文件输入与输出	807
A.2.13 符号常量	752	C.1 概述	807
A.2.14 Math 类	753	C.1.1 数据流	807
A.3 用键盘和屏幕进行简单的输入和 输出	754	C.1.2 文件的优点	807
A.3.1 屏幕输出	754	C.1.3 文件的类型	808
A.3.2 用 Scanner 类进行键盘 输入	755	C.1.4 文件名	809
A.4 if-else 语句	757	C.1.5 包 java.io	809
A.4.1 布尔表达式	758	C.2 使用 PrintWriter 写文本文件	809
A.4.2 嵌套语句	761	C.3 读取文本文件	812
A.4.3 多重 if-else 语句	763	C.3.1 使用 Scanner 读取文本文件	812
A.4.4 条件运算符(可选)	764	C.3.2 使用 BufferedReader 读取 文本文件	814
A.5 switch 语句	764	C.3.3 定义打开数据流的方法	816
A.6 枚举	766	C.4 二进制文件的 I/O	817
A.7 作用域	768	C.4.1 使用 DataOutputStream 写 二进制文件	817
A.8 循环	769	C.4.2 使用 DataInputStream 读取 二进制文件	821
A.8.1 while 语句	769	C.5 File 类	823
A.8.2 for 语句	770	C.6 对象串行化	824
A.8.3 do-while 语句	772	附录 D 文档与程序设计风格	828
A.8.4 关于循环的其他信息	773	D.1 命名变量与类	828
A.9 String 类	774		
A.9.1 字符串中的字符	775		
A.9.2 字符串的联接	776		
A.9.3 String 方法	777		
A.10 StringBuilder 类	779		