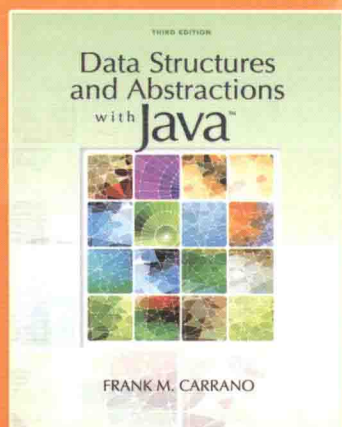


数据结构与抽象 (Java版) (第三版)

Data Structures and Abstractions with Java
Third Edition



[美] Frank M. Carrano 著

张引 等译



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

国外计算机科学教材系列

数据结构与抽象 (Java 版)

(第三版)

Data Structures and Abstractions with Java
Third Edition

[美] Frank M. Carrano 著
张 引 等译

电子工业出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书是为数据结构入门课程而编写的教材。作者在编写过程中自始至终考虑到了 Java 与对象,为教师和学生提供了一种精心设计并经过教学实践的方式,借助 Java 讲授 ADT 和对象。本书独特的设计将内容组织为相对较短的章节。这种方式使学习更容易,并留出了教学的机动性。本书教给学生如何使用线性表、词典、栈、队列等来组织数据。利用这些数据组织方式,学生们将学到算法设计的相关技术。

本书可作为高等院校计算机等相关专业数据结构课程的主教材,也适合广大计算机爱好者和软件开发人员学习与参考。

Authorized translation from the English language edition, entitled Data Structures and Abstractions with Java, Third Edition, 978-0-13-610091-1 by Frank M. Carrano, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright © 2012 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 PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright © 2016.

本书中文简体字版专有版权由 Pearson Education(培生教育出版集团)授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书贴有 Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2012-6202

图书在版编目(CIP)数据

数据结构与抽象:Java版:第三版/(美)卡拉诺(Carrano,F.M.)著;张引等译.

北京:电子工业出版社,2016.10

书名原文:Data Structures and Abstractions with Java, Third Edition

国外计算机科学教材系列

ISBN 978-7-121-27616-3

I. ①数… II. ①卡… ②张… III. ①数据结构-高等学校-教材 ②JAVA 语言-程序设计-高等学校-教材
IV. ①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字(2015)第 277728 号

策划编辑:冯小贝

责任编辑:李秦华

印 刷:三河市良远印务有限公司

装 订:三河市良远印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:39.5 字数:1122 千字

版 次:2016 年 10 月第 1 版(原著第 3 版)

印 次:2016 年 10 月第 1 版

定 价:89.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888, 88258888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:fengxiaobei@phei.com.cn。

前言

作者的话

欢迎大家阅读《数据结构与抽象 (Java 版) (第三版)》。这是一本针对数据结构入门课程 (通常被称为 CS-2) 而编写的书, 它可以作为我写的另外一本书 *Imagine! Java* 的续篇。

无论是一名教师还是一位学生, 我想告诉你, 这本书的撰写是基于我执教本科计算机科学课程 30 余年的经验。我希望这本书具有很好的读者友好性, 使学生可以毫不费力地学, 教师可以事半功倍地教。为了达到这个目标, 你会发现学习材料是以小块——我称之为“小节”——的形式组织的, 这样易于消化学习内容, 促进学习。本书用大量模拟现实世界问题的例子作为新素材, 帮助学生更好地学习和掌握抽象概念。并使用大量简单的图表阐明、厘清了复杂的思想。

我希望你喜欢读这本书。像很多已经读过此书的读者一样, 可以有效且不断地学习或者讲授数据结构。

致以亲切的问候

Frank M. Carrano

附言: 十分乐意与使用本书的教师和学生们保持联系。下面是我的几种联系方式:

我的 Facebook: www.facebook.com/makingitreal

我的 Twitter: twitter.com/Frank_M_Carrano

我的电子邮箱: carrano@acm.org

我的博客: frank-m-carrano.com/makingitreal

学生注意事项

本书所包含的主题涉及各种数据组织方法的处理, 可以使应用程序高效地访问和操作数据。这些主题是以以后学习计算机科学的基础, 因为它们提供了创建复杂和可靠的软件所需要的基础知识。无论是对设计电子游戏感兴趣还是喜欢开发机器人控制手术的软件, 数据结构的学习对成功而言都至关重要。即使现在不打算学习本书中的所有主题, 以后也有可能接触到它们。衷心希望大家喜欢阅读本书, 并希望它成为未来课程学习的有用参考工具。

在看完这个前言之后, 读者应该阅读一下引言。这样就可以快速地了解它的主要内容, 以及在开始学习前所需要了解的相关 Java 的知识。附录 A 到附录 G 回顾了 Java 的基础知识、类、继承、异常、文件和 javadoc 风格的注释。请注意, 在本书的最后还给出了 Java 的保留字、基本数据类型、运算符的优先级和 Unicode 字符的列表。

另外, 请一定要浏览此前言的剩余部分, 从而了解本书将对你的学习有帮助的特色之处。

组织结构

本书主题的组织、顺序的安排和内容展开的幅度都更易于学习和教学。它使读者一次专注于一个概念, 为读者提供主题学习顺序的灵活性, 并清晰地区分抽象数据类型 (ADT) 的说明和

具体实现。为了达到这些目标，本书将内容组织成 30 章，每章都由一些短小的编了号的小节组成，每个小节讲解一个概念。每一章或是讲述一个 ADT 的说明，或是讲述它的实现。读者可以选择在了解一个 ADT 的说明后紧接着学习它的实现，也可以选择考虑任何实现问题之前先探讨若干个 ADT 的说明和使用。本书的组织结构可以使读者方便地选择自己喜欢的主题顺序进行阅读。

内容一览表

下面的各章名称列表显示了本书内容的整体组成情况。下面将会更进一步地描述各个章节的主要内容。请注意，第 30 章(英文版本)和附录是在线提供的内容^①。

引言	第 20 章 词典的实现
第 1 章 袋子	第 21 章 散列概述
第 2 章 用数组实现袋子	第 22 章 用散列实现词典
第 3 章 用链表实现袋子	第 23 章 树
第 4 章 算法的效率	第 24 章 树的实现
第 5 章 栈	第 25 章 二叉查找树的实现
第 6 章 栈的实现	第 26 章 堆的实现
第 7 章 递归	第 27 章 平衡查找树
第 8 章 排序引论	第 28 章 图
第 9 章 快速排序方法	第 29 章 图的实现
第 10 章 队列、双端队列和优先队列	第 30 章 可变的、不可变的和可复制的对象
第 11 章 队列、双端队列和优先队列的实现	
第 12 章 线性表	附录
第 13 章 用数组实现线性表	A. Java 基础
第 14 章 用链表实现线性表	B. Java 类
第 15 章 迭代器	C. 由已有类创建新类
第 16 章 有序表	D. 类的设计
第 17 章 继承及线性表	E. 异常处理
第 18 章 查找	F. 文件输入与输出
第 19 章 词典	G. 文档与程序设计风格

第 3 版的新内容

根据读者和审稿人的意见，对部分素材进行了重新组织。很多学生对栈和队列已经相当熟悉了，所以对这些数据组织方式的内容出现在这一版较靠前的章节中。而且，重新组织以后，学生可以更好地理解链接数据的难点。在链表中添加或者删除头节点是最简单的操作。在介绍袋子的过程中，本书使用这些简单的链表操作来实现袋子。在袋子之后介绍的是栈，栈是一个更加有用的组织方式，它的一种实现也是基于相同的简单链表。队列的实现为讨论对链表尾节点进行添加和删除提供了机会。最后，本书对线性表的处理中关注了对链表中间节点进行添加和删除的操作。

^① 在线章节请登录华信教育资源网(www.hxedu.com.cn)下载。

读者将注意到相对于前一版本,算法的效率(包括提升动机)、递归和排序等内容也出现在此版本的靠前章节中。为了保持对数据结构的关注,将原来的前三章——Java 类、由已有类创建新类和类的设计移到了附录部分。现在引言之后直接就开始介绍第一个数据合集——袋子。不过,对于在开始学习本书主题之前需要学习 Java 类的读者,仍然可以在附录中找到原来这些章节的完整内容。

最后,这一版增加了一些新的特色。本书以“问题解决”的形式展示了大量的例子。在这种方式下,首先提出一个问题,然后讨论它的解决方案并进行实现。偶尔会出现的“设计决策”会探讨一个解决方案的多种设计选择。这两个新的特色帮助学生思考程序设计的重要方面,也让他们在特定的场景下思考概念。上一个版本中已有的注释、编程小贴士和自测问题(含答案)在本版中均进行了保留。此外,读者将发现 Deque 接口、ArrayDeque 类的介绍以及其他的项目实践题目。

下面是本书新内容的总结:

- 较早介绍抽象数据类型、可变大小数组和链接数据。
- 对链接数据的循序渐进的讲解。
- 较早介绍算法效率、栈、递归、排序和队列等内容。
- 有需要较好的算法效率的动力。
- 第 1 章至第 3 章——袋子、用数组实现袋子和用链表实现袋子——介绍和实现 ADT 袋子。
- 新元素包括问题解决和设计决策。
- 第二版中出现在开头章节的对 Java 类的回顾内容现在移到了附录中。
- 标准接口 Deque 和类 ArrayDeque 的涵盖。
- 附加的实践项目。
- 自测问题的答案出现在每章的最后而不是附录中。

促进学习的特色之处

首先,每章一开始都给出了内容列表,学生需要阅读的预备章节和附录列表,以及本章知识的学习目标。其次,贯穿本书还出现了如下所示的其他教学元素:



注 以楷体字印刷的段落对重要的思想进行描述或者总结,且需要和上下文一起阅读。



编程小贴士 在相关之处给出的改进或者有助于编程的建议。



示例 阐明新概念的大量例子。



解决问题 以“解决问题”的形式给出的大型例子,其中首先提出问题,然后对其解决方案进行讨论、设计和实现。



设计决策 在制定解决方案的时候,为读者可能做出的设计选择进行探讨。“设计决策”这一元素列出这样的设计选项,并给出为特定例子所做出的最终选择的依据。这些讨论经常出现在解决问题例子的上下文中。



自测问题答案 在每一章中都会提出一些自测问题,融合在章节的文字中,来巩固刚刚提出的概念。这些“自我检测”问题帮助读者理解这些知识,因为回答这些问题需要停顿和反思。这些问题的答案在每一章的最后给出。

练习题和实践项目 可以通过解决每一章最后的练习题和实践项目来进行更深入的实践。透

憾的是, 尽管不是课堂授课, 我们也不能给读者提供这些练习题和实践项目的答案。只有采用本书进行教学的教师才可从出版商那里获取选中的答案。如果需要获得这些练习题和实践项目的帮助, 读者需要联系你的指导老师。

教师和学生可使用的教辅资源

以下各项资源在出版商的网站 pearsonhighered.com/carrano 上都可以访问到:

- 本书中出现的 Java 代码
- 本书出版以来已发现的所有印刷错误的链接
- 其他的在线内容的链接, 下面将进行介绍

教师资源^①

以下访问受限的资源只提供给采用本书进行教学的教师。教师需要登录 Pearson 教师资源中心 (访问 pearsonhighered.com/carrano) 来获取这些资源:

- PowerPoint 授课幻灯片
- 教师解决方案手册
- 书中的图表

除此之外, 教师可以访问本书的同步网站来获取以下在线高级内容, 仍然可以通过 pearsonhighered.com/carrano 进行访问:

- 第 30 章
- 术语表
- 附录 B、附录 C 和附录 D 的练习题和实践项目

请联系你的 Pearson 销售代表来获取教师访问码。联系方式可以通过访问 pearsonhighered.com/replocator 进行查看。

学生资源^②

以下资源学生可以通过登录本书同步网站 (访问 pearsonhighered.com/carrano) 来获得:

- 第 30 章
- 术语表
- 附录 B、附录 C 和附录 D 的练习题和实践项目

学生必须使用书前面的访问卡来注册和进入同步网站。没有访问码的学生可以按照同步网站上列出的步骤来购买访问权限。

请注意 Java 类库可以通过访问 download.oracle.com/javase/7/docs/api/ 来获得。

章节概述

本书的读者应当已经修完一门编程课程, 最好是 Java 编程。附录部分覆盖了假定读者要了解

① 为获取本书的教辅资源, 可参阅本书目录后所附的“教辅申请表”——编者注。

② 相关文件也可登录华信教育资源网 (www.hxedu.com.cn) 下载。

的 Java 基础知识。大家可以将附录作为复习内容,或者作为从其他编程语言向 Java 过渡的基础。这本书本身就以引言开头,为我们将要学习的数据组织做好了铺垫。

- **第 1 章至第 3 章:** 介绍袋子这种抽象数据类型(ADT)。通过将这部分内容划分为几个章节,清晰明了地将袋子的说明、使用和实现区分开来。例如,第 1 章详细说明袋子并提供了若干使用的例子。第 2 章涵盖了基于数组和向量的具体实现,而第 3 章则介绍了链表并使用链表对袋子类进行定义。

同样,在整本书中,当讨论多种多样的其他类型的 ADT 的时候,也将它们的说明和实现进行了分离。读者可以选择先学习对 ADT 进行说明和使用的章节,以后再学习具体实现的章节;也可以选择按照书中的顺序,在学习说明和使用之后马上实现这些 ADT。各章预备知识的列表将在这个前言的后面部分给出,以帮助大家规划自己学习整本书的路线。

第 2 章不仅仅是简单地实现 ADT 袋子,也展示了如何通过开始只关注核心方法来实现一个类的方法。在实现类的时候,一种大有益处的常见做法是首先实现和测试核心方法,以后再定义其他的方法。

- **第 4 章:** 介绍了算法的复杂度。我们将这一主题融入到后面的章节中。
- **第 5 章和第 6 章:** 第 5 章讨论栈,给出了栈的使用实例。第 6 章通过使用数组、向量和链表来实现栈。
- **第 7 章至第 9 章:** 接下来,介绍递归这种问题解决工具以及它和栈的关系。递归和算法的效率一样,是一个在整本书中不断涉及的主题。例如,在第 8 章和第 9 章讨论各种各样的排序技术和它们的相对复杂度时,我们考虑这些算法的迭代版本,也考虑其递归版本。
- **第 10 章和第 11 章:** 第 10 章讨论队列、双端队列和优先队列,第 11 章考虑它们的实现。在第 11 章中,介绍了循环链表和双向链表。
- **第 12 章、第 13 章和第 14 章:** 接下来的这三章介绍 ADT 线性表。我们在抽象层次上讨论了这种合集,然后分别使用数组、向量和链表对它进行了实现。
- **第 15 章:** 然后,基于线性表的讲解讨论迭代器。这一章考虑和实现了 Java 的迭代器接口 `Iterator` 和 `ListIterator`。本章还介绍了 `Iterable` 接口。
- **第 16 章和第 17 章:** 第 16 章继续讨论线性表,介绍了有序表,关注有序表的两种可能的实现及其效率。第 17 章说明如何将线性表作为有序表的超类,并且讨论了超类的一般设计。
- **第 18 章:** 在线性表和有序表的背景下,考察查找数组和链表的一些策略。这些讨论是后续章节的坚实基础。
- **第 19 章至第 22 章:** 第 19 章涵盖了 ADT 词典的说明和使用。第 20 章介绍词典的基于链表的实现和基于数组的实现。第 21 章介绍散列,而第 22 章则利用散列实现了词典。
- **第 23 章至第 27 章:** 第 23 章讨论了树及其可能的用法。在给出树的一些例子的同时,本章介绍了二叉查找树和堆。第 24 章考虑二叉树和一般树的实现,而第 25 章则关注二叉查找树的实现。第 26 章说明如何使用数组来实现堆。第 27 章介绍平衡查找树。这一章包括 AVL 树、2-3 树、2-4 树、红黑树和 B 树。
- **第 28 章至第 29 章:** 接着讨论图,并且着眼于它的几种应用和两种具体实现。
- **第 30 章:** 最后一章详细阐述可变对象和不可变对象的概念,介绍了克隆。在数据是可变的情况下,如果客户程序维护一个 ADT 内部数据的引用,那么它就可以改变这个数据,而不用使用类的公有方法。我们考虑对客户程序如此操作进行预防的步骤。
- **附录 A 至附录 G:** 附录提供关于 Java 的补充知识。如前所述,附录 A 回顾了到类之前(不包括类)的基础知识。不过,该附录也涵盖了 `Scanner` 类、枚举类型、装箱和拆箱以及 `foreach` 循环。附录 B 讨论 Java 类,附录 C 进一步扩展这一主题,着眼于组合和继承,附录 D

关注类的设计。附录 E 涵盖异常处理, 附录 F 讨论文件。附录 G 考虑程序设计风格和注释, 介绍了 javadoc 注释风格并且定义了我们在本书中所使用的标签。

致谢

在此, 衷心地感谢下列审稿人员。正因为他们认真阅读了上一版本, 提出了中肯的意见和建议, 才使本书的质量和水平得到极大地提高。

Steven Andrianoff—*St. Bonaventure University*

Brent Baas—*LeTourneau University*

Timothy Henry—*University of Rhode Island*

Ken Martin—*University of North Florida*

Bill Siever—*Northwest Missouri State University*

Lydia Sinapova—*Simpson College*

Lubomir Stanchev—*Indiana University*

Judy Walters—*North Central College*

Xiaohui Yuan—*University of North Texas*

特别感谢在这本书漫长的修订过程中支持我的搭档们。我的编辑 Tracy Dunkelberger 给予我持续不断的热情、鼓励、智慧和指导。Melinda Haggerty 和 Allison Michael 使审阅工作井井有条。Stephanie Sellenge 见证了本书及其增补内容的发展。与我长期合作的文字编辑 Rebecca Pepper, 确保我的表述清晰无误, 语法正确。Jeff Holcomb、Bob Engelhardt 和 Louise Capulli 指导了这本书的出版。每当我需要他们的时候, 这支队伍一直与我携手同行。非常感谢他们!

对前面诸位感谢并不能削减我对许多其他帮助者的感激之情。再次感谢本书前两个版本的审稿人:

第二版的审稿人:

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*

第一版的审稿人:

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*

还要感谢很多其他为旧版本提供帮助的人。他们是 Alan Apt, James Blanding, Lianne Dunn, Mike Jacobbe, Toni Holm, Charles Hoot, Brian Jepson, Rose Kernan, ChristiannaLee, Patrick Lindner, John Lovell, Vince O'Brien, Patty Roy, Walt Savitch, Ben Schomp, HeatherScott, Carole Snyder, Chirag Thakkar, Camille Trentacoste, Nate Walker, and Xiaohong Zhu。

最后,我想感谢我的家人和朋友——Doug, Ted, Vandee, Nancy, Sue, Tom, Maybeth, Marge 和 Lorraine——给了我一个远离电脑的生活。

感谢大家,感谢你们的专业支持和鼓励。

Frank M. Carrano

各章的预备知识

本书的每一章和每一个附录都假定读者已经在前期学习了一些知识。下面的表格列出了这些预备知识,其中的数字代表了章节的编号,字母指明了参考的附录。读者可以根据这些信息来规划自己的学习路径。

	预备知识
第1章 袋子	C, D
第2章 用数组实现袋子	1, D
第3章 用链表实现袋子	1, 2
第4章 算法的效率	2, 3, B
第5章 栈	1, D
第6章 栈的实现	2, 3, 4, 5
第7章 递归	2, 3, 4, 5, B
第8章 排序引论	3, 4, 7
第9章 快速排序方法	4, 7, 8
第10章 队列、双端队列和优先队列	5, 8, D
第11章 队列、双端队列和优先队列的实现	2, 3, 6, 10
第12章 线性表	6, B, D, E
第13章 用数组实现线性表	2, 4, 12, D
第14章 用链表实现线性表	3, 11, 12, 13
第15章 迭代器	13, 14, E
第16章 有序表	4, 7, 12, 14
第17章 继承及线性表	12, 13, 14, 16, C
第18章 查找	4, 7, 12, 13, 14, 16
第19章 词典	12, 15, 18
第20章 词典的实现	3, 4, 12, 13, 14, 15, 18, 19
第21章 散列概述	19, 20
第22章 用散列实现词典	4, 13, 14, 15, 19, 20, 21

	预备知识
第 23 章 树	5, 7, 14, 15, 18
第 24 章 树的实现	5, 10, 14, 23, C, E
第 25 章 二叉查找树的实现	7, 19, 23, 24, C
第 26 章 堆的实现	2, 13, 23
第 27 章 平衡查找树	23, 24, 25
第 28 章 图	5, 10, 23
第 29 章 图的实现	5, 10, 12, 15, 19, 23, 28
第 30 章 可变的、不可变的和可复制的对象	13, 14, 15, 16, 17, C, E
附录 A Java 基础	程序设计语言知识
附录 B Java 类	A
附录 C 由已有类创建新类	B
附录 D 类的设计	A, B, C, G
附录 E 异常处理	A, B, C
附录 F 文件输入与输出	A, D, E
附录 G 文档与程序设计风格	A

参与本书翻译工作的还有张弦、张文轩、敦婧瑜、张锐、王颖、钦南、姚亮、张扬扬、戴莉莉、任晓琳、黎磊、姜利成、王一兵、张月娇、李哲蓉、胡直峰、钱宏泽等。

引言

如果稍稍留意周围的人或事，就会意识到人们可能采取不同的组织方式去处理日常的事务。例如，今天早晨去商店买东西，在收银结算时将到付费队伍的最后面去排队。大家自觉地按照时间先后的顺序排队，排在最前面的人肯定第一个享受结算服务且最早离开队伍。最终，将轮到你排到队伍的最前面，付费后拿着装有所购物物品的袋子心满意足地离开。这时，购物袋里的物品是没有什么顺序之分的，有些还可能是相同的商品。

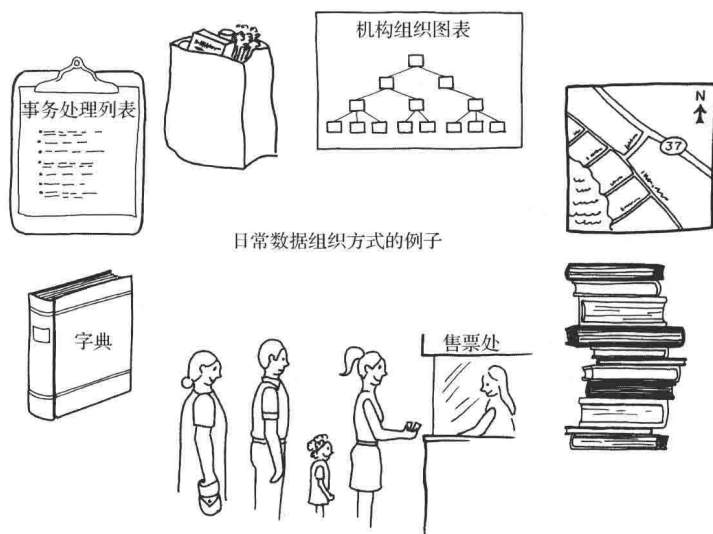
看到放在书桌上的一摞书或者一叠纸了吗？对这样的一摞东西（即栈），我们很容易查看或者取走放在最上面的物品，也很容易继续在最上面添加新的物品。一摞东西中的物品也是按照时间先后的顺序组织的，即最后添加的物品放在最上面，而最先添加的物品放在最下面。

再看看放在桌上的那张事务处理列表吧。对你而言，这张表中每一项书写的位置或许重要或许不重要。但是，还是有可能在写的时候考虑过是按照事务的重要程度排列，还是按照它们的字母顺序来排列。其实，书写的顺序是你自己确定的，事务处理列表只不过列出了所有待处理的事务条目而已。

字典是单词和它们的定义根据字母顺序组织的列表。我们往往需要查找某个单词，查看它的定义。如果字典是纸质印刷的，这种按照字母顺序的组织方式将有助于快速地找到单词。如果字典是计算机化的，虽然表面上看不到它内部按字母顺序的组织方式，但是该方式仍然提高了检索单词的速度。

既然提到了计算机，那么就让我们看一下文件的组织方式——它们被组织到文件夹或目录中，即每个文件夹包含若干其他的文件夹或文件。这种类型的组织方式是层次型的。如果对此组织方式画一张图的话，可以看到它与家族树或公司的内部部门图有相似之处。确实，这些数据的组织方式是相似的，称为树。

最后，注意一下为周末出游所设计的路线图。这张含有道路和城镇信息的图指示了如何从一个地方到达另一个地方，往往其中有多条可选的线路。例如，有一条路线可能是距离较短的，而另一条可能是更快到达的。这种路线图具有的数据组织方式称为图。



除了前面这些日常例子之外，计算机程序也需要组织其数据。也就是说，程序可以使用列表、栈、字典等数据组织方式。这些数据组织方式被表示为多个抽象数据类型。一个抽象数据类型

(Abstract Data Type, ADT)是一份规格说明书,描述了一个数据集及其数据上的操作,即每个 ADT 详细说明该存储什么样的数据,以及对这些数据要进行哪些操作。不过,ADT 并没有明确说明如何存储数据或者如何实现操作,因此对这些 ADT 的讨论可以不依赖于任何程序设计语言。相对 ADT 而言,一个数据结构是某个 ADT 在一种程序设计语言中的实现。

一个合集(collection)是更广的术语,是指一个包含一组对象的 ADT。一些合集允许重复的项,而另一些不允许。一些合集按照某种顺序组织它们的内容,而另一些则无序地组织。一个容器(container)是实现合集的一个类。大家以后可能看到,有些人对术语“容器”和“合集”会不加区别地进行使用。

我们可以创建一个 ADT 袋子(bag),它由一个允许重复元素的无序合集构成,很像是一个食品杂货袋,一个午餐袋或者是一个装满薯片的袋子。假设从薯片袋子中取出一片薯片,这时你不会知道这片薯片是何时放进袋子中的,也不可能知道袋子中是否还有一片与刚取出的薯片形状完全相同的薯片。但是,实际上并不需要关心这些。如果确实在意,就不会将薯片存放在一个袋子中。

袋子不会将其内容排序,但是有时候需要对物品进行排序的。各种各样的 ADT 使用不同的方式对它们所包含的元素进行排序。例如,ADT 线性表(list)将它所含的项简单地进行编号。因此,一个线性表有第一项、第二项,以此类推。尽管可以向线性表的末尾添加项,实际上也可以在线性表开头或者线性表的某两项之间插入项。这么做会导致新项之后的项的重新编号。此外,可以在线性表中删除某特定位置的项。因此,线性表中项的位置并不一定表明它被添加的时间。请注意,线性表并不决定项的存放位置,而是由编程人员决定。

相比之下,ADT 栈(stack)和队列(queue)则以时间顺序来组织它们的项。当从栈中删除一项时候,所删除的是最近被添加的那一项。当删除队列中的一项的时候,所删除的是最早被添加进队列中的那一项。因此,栈就像一摞书。可以取走最上面的书或者将另一本书添加到这摞书的最上面。而队列就像是人排队的队伍,人们从队首离开,从队尾加入。

如果项是可比的,一些 ADT 中的条目始终保持有序的状态。例如,字符串可以按照字母顺序进行组织。当向 ADT 有序表(sorted list)中添加一项的时候,ADT 决定要把该项放入哪个位置。不能像对待 ADT 线性表那样为待添加项指定一个位置。

ADT 字典(dictionary)包含成对的项,这一点非常像一本包含单词及其定义的语言类字典。在这个例子中,单词充当的是关键字的角色,它可以定位词条的位置。一些字典会将词条排序,另一些则不会。

ADT 树(tree)根据某种层次结构来组织它的条目。例如,在家族树中,人们与他们的孩子和父母相联系。ADT 二叉查找树(binary search tree)使用层次化的有序结构,这使得定位其中的条目更加简单。

ADT 图(graph)是 ADT 树的一般化。ADT 图关注的是条目之间的关系,而非层次化的组织。例如,一张路线图是一个显示了城镇间道路和距离的图。

本书向大家介绍如何使用和实现这些数据组织方式。在开始学习之前,需要了解 Java。附录 A 回顾了 Java 中的基本语句。附录 B 讨论类和方法的基本构造。可以选择简单浏览这些材料,或是认真阅读,也可以在需要的时候进行查看。附录 C 和附录 D 介绍 Java 的内容,有可能某些部分或者全部的内容对你而言是全新的。附录 C 涵盖了技术性内容,包括组合和继承,其目的是为了从已有的类构建新的类。附录 D 讨论如何设计类,定义方法和编写 Java 接口。使用接口和书写说明方法的注释对描述 ADT 至关重要。附录 E 介绍如何处理异常,附录 F 展示了对外部文件的读写,附录 G 则概述了如何书写适合 javadoc 的注释^①。

^① 本书共有 7 个附录(附录 A 至附录 G)。为减少本书的印制成本,已将其附录的电子版上载至华信教育资源网(<http://www.hxedu.com.cn>),有兴趣的读者可免费下载——编者注。



Pearson

尊敬的老师:

您好!

为了确保您及时有效地申请培生整体教学资源,请您务必完整填写如下表格,加盖学院的公章后传真给我们,我们将会 在 2-3 个工作日内为您处理。

请填写所需教辅的开课信息:

采用教材			☐ 中文版 ☐ 英文版 ☐ 双语版
作 者		出版社	
版 次		ISBN	
课程时间	始于 年 月 日	学生人数	
	止于 年 月 日	学生年级	☐ 专 科 ☐ 本科 1/2 年级 ☐ 研究生 ☐ 本科 3/4 年级

请填写您的个人信息:

学 校			
院系/专业			
姓 名		职 称	☐ 助教 ☐ 讲师 ☐ 副教授 ☐ 教授
通信地址/邮编			
手 机		电 话	
传 真			
official email(必填) (eg:XXX@ruc.edu.cn)		email (eg:XXX@163.com)	
是否愿意接受我们定期的新书讯息通知: ☐ 是 ☐ 否			

系 / 院主任: _____ (签字)

(系 / 院办公室章)

____年____月____日

资源介绍:

—教材、常规教辅(PPT、教师手册、题库等)资源: 请访问www.pearsonhighered.com/educator; (免费)

—MyLabs/Mastering 系列在线平台: 适合老师和学生共同使用; 访问需要 Access Code; (付费)

100013 北京市东城区北三环东路 36 号环球贸易中心 D 座 1208 室
电话: (8610) 57355003 传真: (8610) 58257961

Please send this form to:

目 录

第 1 章 袋子	1	第 4 章 算法的效率	69
袋子	1	动机	69
袋子的行为	2	算法效率的度量	70
袋子的规格说明	3	基本操作次数的统计	72
接口	7	最好、最坏和平均情况	74
ADT 袋子的使用	9	大 O 表示法	74
像使用自动售货机一样使用 ADT	12	程序结构的复杂度	77
Java 类库: 接口 Set	14	效率的图形化表示	78
第 2 章 使用数组实现袋子	19	ADT 袋子不同实现的效率	80
使用固定大小的数组实现 ADT 袋子	19	基于数组的实现	80
一个类比	19	基于链表的实现	82
一组核心方法	20	两种实现方法的比较	83
核心方法的实现	21	第 5 章 栈	89
核心方法的测试	27	ADT 栈的规格说明	89
更多方法的实现	30	利用栈处理代数表达式	92
删除物品的方法	31	应用问题: 中缀代数表达式中括号平衡的	
使用可变大小的数组实现 ADT 袋子	38	检查	93
调整数组的大小	38	应用问题: 中缀表达式向后缀表达式的	
袋子的一种新的实现	41	转换	97
使用数组实现 ADT 袋子的优缺点	43	应用问题: 后缀表达式的求值	101
第 3 章 使用链表实现袋子	48	应用问题: 中缀表达式的求值	102
链表	48	程序栈	104
通过添加节点到表头来创建链表	49	Java 类库: 类 Stack	105
ADT 袋子的链表实现	50	第 6 章 栈的实现	111
私有的类 Node	51	基于链表的实现	111
类 LinkedBag 的概要	52	基于数组的实现	114
一些核心方法的定义	53	基于向量的实现	117
核心方法的测试	56	Java 类库: Vector 类	117
方法 getFrequencyOf	57	使用向量实现 ADT 栈	118
方法 contains	58	第 7 章 递归	123
从链表中删除物品	58	什么是递归	123
方法 remove 和 clear	59	跟踪一个递归方法	127
具有方法 set 和 get 的类 Node	62	有返回值的递归方法	130
使用链表实现 ADT 袋子的优缺点	64	递归地处理一个数组	132

递归地处理一个链表	134	第 10 章 队列、双端队列和优先队列	195
递归方法的时间效率	135	ADT 队列	195
countDown 的时间效率	135	解决问题: 模拟排队	198
计算 x^n 的时间效率	137	解决问题: 计算出售股票时的资本	
一个复杂问题的简单解决方案	138	增益(一)	203
一个简单问题的拙劣解决方案	142	Java 类库: 接口 Queue	206
尾递归	144	ADT 双端队列	207
间接递归	146	解决问题: 计算出售股票时的资本	
使用栈代替递归	146	增益(二)	209
第 8 章 排序引论	156	Java 类库: 接口 Deque	209
组织 Java 对数组排序的方法	157	Java 类库: ArrayDeque 类	211
选择排序	158	ADT 优先队列	211
迭代选择排序	159	解决问题: 跟踪你的作业	212
递归选择排序	160	Java 类库: 类 PriorityQueue	214
选择排序的效率	161	第 11 章 队列、双端队列和优先队列的	
插入排序	161	实现	219
迭代插入排序	162	基于链表队列的实现	219
递归插入排序	163	基于数组队列的实现	222
插入排序的效率	165	循环数组	223
链表的插入排序	165	有一个未使用存储单元的循环数组	224
希尔排序	167	基于向量队列的实现	229
Java 代码	169	基于循环链表队列的实现	231
希尔排序的效率	170	由两部分构成的循环链表	231
算法比较	170	Java 类库: 类 AbstractQueue	236
第 9 章 快速排序方法	176	基于双向链表双端队列的实现	236
归并排序	176	实现优先队列的可用方法	240
数组的归并	176	第 12 章 线性表	245
递归的归并排序	177	ADT 线性表的规格说明	245
归并排序的效率	179	使用 ADT 线性表	250
迭代的归并排序	181	Java 类库: List 接口	253
Java 类库中的归并排序	181	Java 类库: ArrayList 类	253
快速排序	182	第 13 章 用数组实现线性表	257
快速排序的效率	182	用数组实现 ADT 线性表	257
创建划分	183	一个类比	257
快速排序的 Java 代码	185	Java 实现	259
Java 类库中的快速排序	187	用数组实现 ADT 线性表的效率	264
基数排序	188	用 Vector 实现 ADT 线性表	266
基数排序的伪代码	189	第 14 章 用链表实现线性表	272
基数排序的效率	189	操作链表节点	272
算法比较	190	在多种位置加入节点	272

在多种位置删除节点	276	设计一个基类	350
私有方法 getNodeAt	276	创建一个抽象基类	354
开始实现	277	有序表的一种高效实现	355
数据域和构造函数	278	方法 add	356
在列表结尾加入	279	第 18 章 查找	359
在列表给定位置加入	280	问题引入	359
方法 isEmpty 和 toArray	281	查找无序数组	360
测试核心方法	283	无序数组的迭代式顺序查找	360
继续实现	284	无序数组的递归式顺序查找	361
一个更好的实现	286	顺序查找数组的效率	362
尾引用	286	查找有序数组	362
用链表实现 ADT 列表的效率	289	有序数组的顺序查找	363
Java 类库: 类 LinkedList	291	有序数组的二分查找	363
第 15 章 迭代器	296	Java 类库: 方法 binarySearch	367
迭代器是什么	296	二分查找数组的效率	367
Iterator 接口	297	查找无序链表	368
使用 Iterator 接口	298	无序链表的迭代式顺序查找	369
独立类迭代器	302	无序链表的递归式顺序查找	369
内部类迭代器	305	顺序查找链表的效率	370
基于链表实现	305	查找有序链表	370
基于数组实现	309	有序链表的顺序查找	370
为什么迭代器方法在它们自己的		二分查找有序链表	371
类中	311	查找方法的选择	371
ListIterator 接口	313	第 19 章 词典	377
使用 ListIterator 接口	315	ADT 词典规格说明	377
ListIterator 接口基于数组的实现	318	Java 接口	380
内部类	319	迭代器	381
Java 类库: Iterable 接口	323	使用 ADT 词典	382
Iterable 和 for-each 循环	324	问题解决: 电话号码本	382
修改版接口 List	324	问题解决: 词频	386
第 16 章 有序表	330	问题解决: 词的索引	389
ADT 有序表的规格说明	330	Java 类库: Map 接口	392
使用 ADT 有序表	332	第 20 章 词典的实现	396
链表实现	333	基于数组的实现	396
方法 add	334	一个无序数组词典	397
链表实现的效率	340	一个有序数组词典	401
使用 ADT 线性表的实现	340	基于向量的实例	404
效率问题	343	链式实例	408
第 17 章 继承及线性表	348	一个无序链式词典	409
使用继承实现有序表	348	一个有序链式词典	409