

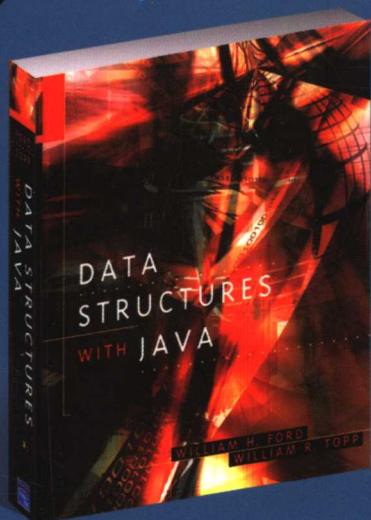


国外经典教材·计算机科学与技术

PEARSON
Prentice
Hall

数据结构 (Java版)

Data Structures with Java



(美) William H. Ford 著
William R. Topp 译
梁志敏

示例简练贴切

图表丰富直观

内容深入广泛



清华大学出版社

国外经典教材·计算机科学与技术

数据结构(Java 版)

(美) William H. Ford 著
William R. Topp 译
梁志敏



清华大学出版社

北 京

Simplified Chinese edition copyright © 2006 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 WITH JAVA™**, by William H. Ford, William R. Topp, Copyright © 2005

EISBN: 0-13-047724-9

All Rights Reserved.

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

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培生教育出版集团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: 01-2005-0937

版权所有, 翻印必究。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(Java 版)/(美)福特(Ford, W. H.), (美)托普(Topp, W. R.) 著; 梁志敏 译. —北京: 清华大学出版社, 2006. 11

书名原文: **Data Structures with Java**

(国外经典教材·计算机科学与技术)

ISBN 7-302-13544-4

I. 数… II. ①福…②托…③梁… III. ①数据结构 ②Java 语言—程序设计 IV. ①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字(2006)第 086801 号

出版者: 清华大学出版社

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

社总机: 010-62770175

组稿编辑: 曹 康

封面设计: 康 博

印刷者: 清华大学印刷厂

装订者: 三河市新茂装订有限公司

发行者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 62.25 字数: 1594 千字

版 次: 2006 年 11 月第 1 版 2006 年 11 月第 1 次印刷

书 号: ISBN 7-302-13544-4/TP · 8493

印 数: 1 ~ 3000

定 价: 118.00 元

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

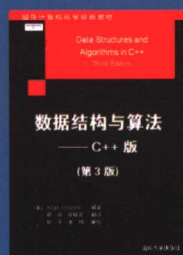
文稿编辑: 徐燕萍

版式设计: 康 博

梁志敏 译

译者简介

梁志敏，清华大学电子与通信工程专业硕士，现为某国防科研单位工程师，对电子工程及信息处理技术有着深入的研究。曾在《电视技术》和《电信技术》等多家媒体上发表多篇文章，翻译过多本计算机图书，并获得军队科技进步二、三等奖。



数据结构与算法——C++版 (第3版)

(美) Adam Drozdek 编著
郑岩 战晓苏 翻译
郭平 宋梅 审校

ISBN: 7-302-11998-8

定价: 69.00元

本书全面系统地介绍了计算机科学教育中的一个重要组成部分——数据结构,并以C++语言实现相关的算法。书中主要强调了数据结构和算法之间的联系,使用面向对象的方法介绍数据结构。本书包含大量的示例分析和图形,便于读者进一步理解和巩固所学的知识。



Java程序设计与数据结构 (第2版)

(美) Kenneth A. Lambert
Martin Osborne 等著

佟占杰 等译
郭平 张敏 审校

ISBN: 7-302-11006-9

定价: 59.80元

本书在介绍如何开发面向对象程序的同时,还着重讲解了标准数据结构。作者首先介绍了Java中需要特别掌握的部分,然后讨论了程序设计中的类、继承、多态性、递归和复杂度分析等概念。最后一章介绍了线程和同步技术,为学生转向计算机科学的高级课程作了铺垫。另外,作者还采用大量的案例分析贯穿全书始终,突出了软件的生命周期。

出版说明

近年来,我国的高等教育特别是计算机学科教育,进行了一系列大的调整 and 改革,急需一批门类齐全、具有国际先进水平的计算机经典教材,以适应当前我国计算机科学的教學需要。通过使用国外先进的经典教材,可以了解并吸收国际先进的教学思想和教学方法,使我国的计算机科学教育能够跟上国际计算机教育发展的步伐,从而培育出更多具有国际水准的计算机专业人才,增强我国计算机产业的核心竞争力。为此,我们从国外知名的出版集团 Pearson 引进这套“国外经典教材·计算机科学与技术”教材。

作为全球最大的图书出版机构,Pearson 在高等教育领域有着不凡的表现,其下属的 Prentice Hall 和 Addison Wesley 出版社是全球计算机高等教育的龙头出版机构。清华大学出版社与 Pearson 出版集团长期保持着紧密友好的合作关系,这次引进的“国外经典教材·计算机科学与技术”教材大部分出自 Prentice Hall 和 Addison Wesley 两家出版社。为了组织该套教材的出版,我们在国内聘请了一批知名的专家和教授,成立了一个专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动,各位委员根据国内外高等院校计算机及相关专业的现有课程体系,并结合各个专业的培养方向,从 Pearson 出版的计算机系列教材中精心挑选针对性强的题材,以保证该套教材的优秀性和领先性,避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量,我们为这套教材配备了一批经验丰富的编辑、排版、校对人员,制定了更加严格的出版流程。本套教材的译者,全部来自于对应专业的高校教师或拥有相关经验的 IT 专家。每本教材的责编在翻译伊始,就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华,在经过翻译、排版和传统的三审三校之后,我们还请编审委员或相关的专家教授对文稿进行审读,以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限,该套教材在出版过程中很可能还存在一些遗憾,欢迎广大师生来电来信批评指正。同时,也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材,共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

国外经典教材·计算机科学与技术

编审委员会

主任委员:

孙家广

清华大学教授

副主任委员:

周立柱

清华大学教授

委员(按姓氏笔画排序):

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
杨宗源	华东师范大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟祥旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

前言

本书从面向对象的角度讲述了数据结构的基本理论。总的来说，数据结构就是处理批量数据的存储处理机。数据结构又称为集合(collection)，它可进行添加和删除数据项的操作，并且能够提供对集合中元素的访问。面向对象的编程方法将数据结构视为可对数据进行特定操作的对象。类声明定义了数据底层的存储结构和能高效执行操作的实现方法。

数据结构在计算机科学的各个领域中都扮演着非常重要的角色。在几乎所有重要的计算机应用程序的设计和实现中，它都是一个关键的要素。所以大多数学生在回顾数据结构的课程时，都认为这是他们将计算机科学作为一种学科来认识和了解的第一步。在数据结构的学习中会介绍大量的重要概念。

0.1 本书的编排

现代的观点将数据结构分为面向对象的编程原理和算法分析的概念。我们的学习主要是围绕定义集合类型的类别的接口来进行。这种接口把数据结构视为一种抽象数据类型(ADT)，它描述了集合如何存储元素并定义了关键的数据处理操作。集合类是一种聚合数据结构，它用特定的方法来实现存储和访问元素的接口。

本书对数据结构的学习按简单的接口和类的层次结构来进行。位于最高层的接口描述了对集合中对象进行访问和更新操作有关的抽象结构。最底层含有的集合类使用不同的底层存储结构来保存元素和实现接口。接口和集合类的层次结构为数据结构创建了一个总的构造，我们将其称为集合架构(collections framework)。在本书中，我们模仿 Java Collections Framework 创建了一个简单的集合架构，该架构定义了使用有限的一组方法的接口，这些方法描述了集合类的基本功能。

Java Collections Framework 是一个商业级软件系统，它提供了多种方法来帮助应用程序的编程人员。Java Framework 中的集合类代码很大程度上依赖于抽象类和相当复杂的继承层次结构。书中的集合使用用于接口的相同名称和方法签名来模仿 Java Collections Framework，此外还创建了具有相同名称和底层存储结构的集合类。不过，我们使用直观的实现来设计集合类。通过这种方式，学生能够阅读书中的代码并理解各种方法及其算法。学习本书所有内容后，读者会发现在职业生涯中很容易掌握更完整的 Java Collection Framework。

图 0-1 示例了本书中集合架构的接口和集合类。类用阴影表示。List 接口是按位置存储元素的数据结构模型，而 Set 接口是按值存储元素的数据结构模型，它不允许集合中存在重复值。Map 接口是按键/值对存储元素的关联集合模型。映射中存储的键是不可重复的，并且提供了某种机制将键和值关联起来。在映射集合中，程序员通过键而不是索引来访问相应的值。Stack、Queue、PQueue 是用于适配器集合的接口，上述接口将另外一个集合用作底层存储结构，不过只为这些结构提供受限的操作。图是由顶点和连接边组成的数据结构。网络就是图的一种模型。本书会讲述图及其经典的搜索和最优化算法。Graph 接口中列出了这些操

作以及作为具体数据结构的有向图。

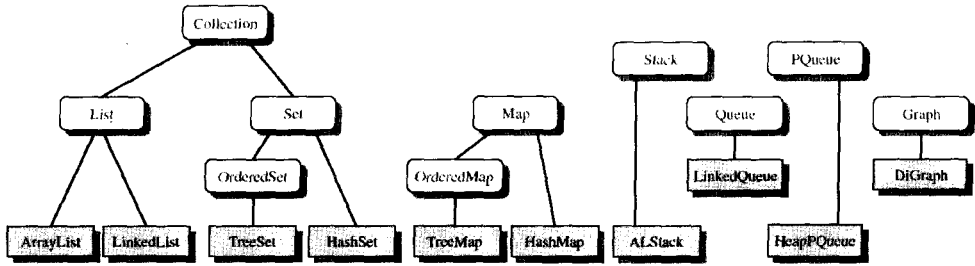


图 0-1 本书中介绍的接口与类

0.2 数据结构与算法

数据结构除了存储数据外，还具有处理数据的操作。这些操作的设计和实现涉及由完成任务所需的有限指令序列所组成的算法。学习数据结构必须对算法也有所了解，以便确保正确地处理数据。我们还必须考虑到大数据集带来的问题。数据结构的选择对访问和更新操作的效率影响非常大。假设要维护 100 000 张信用卡账户，类似数组的结构平均需要 50 000 次迭代才能定位某个元素，而用映射完成同样的任务只需要大约 16 次迭代。这两种结构都具有查找某个数据值的算法，但是两者的性能结果却大不相同。在讨论算法时，我们会使用简单、直观但却非常有效的方法来评估算法的执行时的时间性能，这种评估方式与所用的机器或编程语言无关。

数据结构需要优良的算法，反之亦然。在后面章节介绍图、数据压缩、密码术以及高级递归算法的时候，读者会看到这方面的示例。我们将重点讨论密切依赖于针对问题采用特定方式构造数据的算法和算法设计技术。通过这种方法，我们就能够使用在本书中讲述的数据结构特性。著名的计算机科学家 Niklaus Wirth 在其经典论文 *Algorithms + Data Structures = Programs* 一书中表明：对于计算机编程来说，算法和数据结构具有同样的重要性。在现代的面向对象编程中，数据结构对于算法设计和分析来说仍然至关重要。

0.3 Java 与数据结构

本书中讨论的大多数素材都能够应用于任何编程语言。我们重点强调了与数据结构的设计和使用相关的概念。不过，读者还必须理解如何实现数据结构和编写应用程序。这要求读者采用特定的编程语言来编写代码，在本书中我们选择了 Java 语言。Java 语言具有优良的、支持数据结构现代观点的面向对象设计构造。我们使用这种编程语言以一个层次架构来描绘数据结构。Java 接口定义了数据结构(集合)的类别，而 Java 类则定义了单独的集合类型。

Java 是可移植的，它允许程序不需重新编译就能够在任何平台上运行。这种特性具有许多实用的优点。学生可以先在 Windows、Macintosh 或 UNIX 系统中开发程序，然后再将代码移植到另一个系统中。大学里应该可以为实验室提供中央计算设备，这样学生就可以在自己的主系统上开始或完成所从事的项目。

Java 是创建交互式图形用户界面(Graphical User Interface, GUI)程序的理想语言。大多数现代应用程序的形式是 GUI 程序。第 3 章使用 Java Swing 组件独立地介绍了 GUI 编程。本书的目标是促进读者对数据结构的理解,并且绝大部分示例、程序和练习都基于控制台,也就是使用键盘输入和屏幕输出。不过,GUI 应用程序零散地遍布于全书以及练习当中。

Java 语言具有许多有助于数据结构使用和实现的编程构造。这些构造包括:

- 自动装箱和自动拆箱,自动装箱提供从某个原始类型到其相关包装类型的自动转换,自动拆箱则提供从某个包装类型到其相关原始类型的自动转换。这允许我们将原始数字和字符数据作为对象存储在集合中。
- 集合使用迭代器扫描数据。Java 引入了使编译器能够维护迭代器的“增强的 for”语句。这种语句使用变化形式的 loop 循环简化了初始化迭代器的需求以及使用迭代器方法访问集合元素。
- Java 提供了一个枚举工具,这个枚举工具允许程序定义其成员为命名常量的特殊类。编程人员能够使用 enum 类来增强程序的可读性。我们将在选定的应用程序和实现中示例说明这种特性。
- Scanner 对象为来自文件或键盘的文本数据输入提供一个简单的文本扫描器。这个扫描器将输入流中的文本分割为若干标记,并且使用多种方法将标记转换为一个串或一个原始类型值。

对于数据结构来说,Java 语言中最重要的特性是泛型。泛型扩展了将类型参数与集合类实例、接口或方法相关联的语言。Java 的编译器能够检查代码是否与指定类型一致;如果一致,那么代码是类型安全的。执行类型安全的代码不会导致某些运行时错误,这些运行时错误包括由于不正确的类型检查和类型处理所带来的代码质量和可靠性问题。

第 5 章详细阐述了 Java 泛型。我们首先阐明在 Java 语言中包含泛型的动机,随后讲述创建泛型类和方法时必须使用的机制。这些讨论的重点是读者必须了解的基础。本书的后面部分还将为更复杂的泛型开发相应的语法,并且始终重点讲述读者“需要知道”的基础知识。Java 泛型的其他一些特征则超出了本书的讨论范围。

0.4 本书的功能部件

本书包含能够提高学生知识水平的功能部件以及促进深入理解数据结构的介绍主题。

- UML: 对象技术的发展促进了用于程序设计的各种表示法的出现。其中一个最重要的表示法是通用建模语言(Unified Modeling Language, UML)。UML 使用图形规范来帮助完成对象和程序设计。通过使用 UML 图作为列出接口或集合类中方法的一种有效途径,使读者对 UML 有一定的了解。
- 练习集: 每一章都含有书面练习和编程练习。设计这些练习的目的是巩固对基本知识的理解,使学生能够设计与实现方法以及完成程序,促进对各种素材的创造性认识。每一章还含有编程项目,这些编程项目要求更高级的设计和编程以及开发更多算法。设计模式: 问题的面向对象解决方案经常使用软件重用的公共设计模式进行构造。我们选择介绍了一些设计模式,这些设计模式增强了读者对数据结构及其应用的理解,并且能够帮助构造类、方法以及对象的实现。

- 注释：在研究各种主题的过程中，我们在文中插入了一些主要概念的注释。这些注释突出了重要的知识点，并且为学生提供了一种复习素材的简单有效方法。

0.5 适用范围

本书假定学生至少已学过一门编程课程。针对没有 Java 编程经验的学生，我们在附录 A 中提供了“Java 入门”指南。“Java 入门”采用文本样式编写，并且含有示例和完整的程序。这个指南为读者阅读和理解本书提供了必要的背景知识。学生应当掌握中学的大学预备课程中的数学知识。递归的示例以及算法运行时效率的讨论使用了高等代数课程中所讲述的数学知识。

0.6 教师需要注意的问题

本书可以用作两学期计算机科学授课内容的一部分，也可以作为一门为学习数据结构专门设计的课程。某些学校(如笔者的学校)开设了一门课程来阐述数据结构和介绍数据结构在算法中的应用。我们已经发现：本书中知识点的深度与多样性所达到的教学效果非常不错，并且能够在今后的算法设计与分析课程中为学生提供某些编程工具。本书的设计方式允许教师能够非常灵活、自由地组织和讲解各种知识点。

我们广泛使用示例、图表以及注释来包涵所有知识点。本书的前 7 个章节介绍了学习数据结构时要用到的各种概念和工具。

- 第 1~3 章介绍了 Java 的对象技术特征，包括对象的使用、类的设计与实现、组合、继承、接口以及异常处理。这一部分通过许多示例和程序说明了使用 Time24 对象的数据结构。
- 第 4 章讨论了如何使用独立于任何机器或编程语言的算法运行时间性能来描述算法的特征。我们直观地提出了效率度量 Big-O，并且将其应用在全书范围内，以便有助于选择适用于特定程序应用的算法或数据结构。
- 第 5 章较详细地讲述了 Java 泛型类与方法。泛型的使用贯穿全书，并且这一章节的知识点为学生提供了至关重要的背景知识。
- 第 6 章和第 7 章介绍了递归及其在高效的归并排序与快速排序算法中的应用。对于学习二叉树来说，理解递归至关重要。
- 第 8~19 章覆盖了初次接触数据结构的读者应当了解的主题。我们使用一种体系方式逐渐展开每种数据结构。这种方式认为学生需要从下列 4 个方面来理解每种数据结构：存储数据的方式，指定具体集合类的接口，在应用程序中使用这个类的能力，以及对类实现的理解。
 - 动态数组、单链表和双链表以及二叉树是分别为 ArrayList、LinkedList、TreeSet 和 TreeMap 集合类提供底层存储结构的低级数据结构。我们首先介绍了这些数据结构，然后开发了定位、插入和删除元素的算法。
 - 集合类型(接口)指定了用于集合类的大多数设计特性。一个类可以包含利用底层存储结构特征的额外方法。例如，ArrayList 类定义了方法 ensureCapacity()，该方法更新底层动态数组的大小。

- 我们为每种数据结构都提供了若干应用程序。某些应用程序简单地示例说明了如何使用集合类方法；另外一些应用程序则利用了独特的特征并示例说明了算法与数据结构之间的交互。
- 在讲述数据结构的课程中，学生必须理解每种集合类型的抽象设计和实现设计。他们应该掌握需要使用哪种数据结构以及如何使用这种数据结构。我们在介绍每种集合类的同一章节中开发了各种集合类的实现。读者可以根据自己的需求随时研究这些实现。对集合类实现的讨论独立于学生需要了解的其他方面的内容。
- 第 20~29 章覆盖了数据结构高级课程与算法应用的主题。笔者建议：介绍集与映射的第 19 章既可以作为数据结构预备课程的内容，也可以作为数据结构高级课程的内容。在预备课程中，本章可以向学生介绍应用程序中所使用的重要集合类型以及作为键-值对的元素。在高级课程中，教师能够更详细地开发作为关联数组的映射并讨论区分元素键字段和值字段的迭代器访问方法。

数据结构预备课程介绍了简单的、具有节点和引用字段的二叉搜索树。数据结构高级课程将二叉搜索树的概念扩展到包含平衡树以及定义堆的基于数组的二叉树，此外，还将散列表开发为一种可供选择的搜索结构。第 21、22 和 27 章阐述了上述内容，并且使用它们创建用于有序与无序集、映射和优先队列的高效实现。

计算机科学专业的学生通常都需要参加算法设计与分析课程的学习。这门课程的重点是不同的算法设计策略，并且对算法效率进行更深入的数学分析。伪码被用于描述实现技术。针对实现算法的数据结构，通过示例说明这些数据结构的复杂与总是巧妙的用法，数据结构高级课程融会了算法设计与分析。因此，本书包含了数据与文件压缩(第 23 章)、图算法(第 25 章)、数论与密码术(第 28 章)以及杂类算法(第 29 章)等相关章节。这些章节都是独立的单元，教师可以选取某些主题进行教学。

教师指南

教师指南可供教师使用，它不仅提供了所有书面练习的答案以及大多数编程练习和编程项目的解决方案，而且提供了每章的测验模拟题。除此之外，我们还为练习中开发的程序和类给出了源文件。您可以通过 Prentice Hall 驻北京办事处的销售代理商购买教师指南。

0.7 补充内容

从如下所示的网址上可以得到本书的所有补充内容：

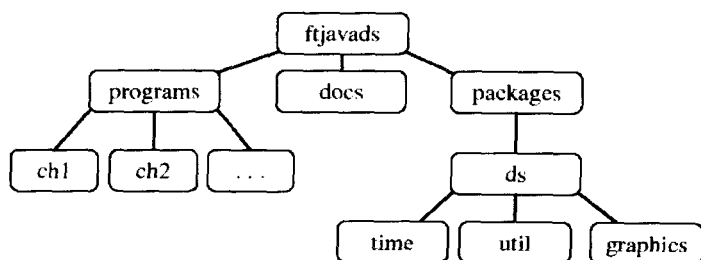
<http://www.fordtopp.com>

<http://www.tupwk.com.cn/downpage>

软件补充内容

软件补充内容位于 fjavads 目录中，这个目录由 programs、packages 和 docs 目录组成。

programs 目录包含用于本书每个章节的若干独立子目录, 这些子目录中存在用于本章中程序的源文件和数据文件。在某些情况下, 读者还会发现指定章节所引用的图形程序和其他程序清单。**packages** 目录包含全书范围内所使用的软件, 它的结构定义了 **ds** 子目录, 该子目录又分支为对应于包 **graphics**、**time** 和 **util** 的目录。**util** 软件包包含了本书所介绍的接口、集合类和静态方法库。



软件 API 文档

文件 **docs** 包含对 **packages** 目录中软件的完整 API 说明, 这为学生和教师提供了易于使用的、资料丰富的在线文档。**packages** 目录中的 Java 源文件清单包含文档注释。**javadoc** 工具使用这些注释来创建相应的 HTML 页面集。通过 Web 浏览器打开文件 **index.html**, 读者就可以使用 API 文档。

Web 补充内容

笔者尽可能使本书自成一體。不过在某些情况下, 我们引入了基本的功能与算法, 并且基于 Web 补充内容进行更详细的讨论。这些补充内容通过简单地扩展本书的版面来提供额外的素材。

现代应用程序通常使用依赖于 GUI 窗体的交互式 I/O, 而 GUI 窗体具有组件和事件处理程序。第 3 章介绍了 GUI 应用程序设计, 并且讨论了基本的组件集以及动作侦听器机制。

图形显示可以帮助读者理解各种算法, 如 Hanoi 塔、八皇后问题等。本书还使用图形显示来示例说明二叉树以及用于二叉搜索树、AVL 树、红-黑树和堆数组的插入与删除方法。软件补充内容包含基本的画图软件。Web 补充内容 “Graphics Package” 给出了对画图系统的详细讨论, 此外还提供了继承的一个优秀示例。

本书只简单描述了用于中缀表达式求解与红-黑树插入和删除方法的算法。对这些算法感兴趣的读者可以通过 Web 补充内容 “Infix Expression Evaluation” 和 “RBTree Implementation” 得到更详细的讨论以及 Java 实现的代码清单。

EZJava IDE

EZJava 是一个灵活的集成开发环境(Integrated Development Environment, IDE), 它允许在 Windows、Linux、Solaris 或 Macintosh OS X 系统下开发和运行 Java 程序。读者可以选择与当前系统匹配的版本。EZJava IDE 允许非常简单地编译和运行一个主类以及位于相同目录

中的任何支持类。对于更高级的软件开发来说，我们可以创建一个由相同软件包结构中多个源文件组成的项目。

EZJava 拥有十分复杂的编程环境才具有的特性。虽然 EZJava 可以用来开发大型的系统，不过它是为了在学术课程作业中方便 Java 的使用而专门设计的。我们可以创建若干单独的程序，并且可以在项目环境之外运行这些程序。应用程序使用一个用于多个运行的命令按钮在自己的运行时窗口中执行命令。EZJava 编译了 Synchronous Java，Synchronous Java 是一个具有并行编程工具的 Java 扩充，并行编程工具在操作系统课程中非常有用。

附录 E 讨论了如何获取和安装 EZJava 以及如何使用其主要功能创建和运行一个程序。这个软件所集成的帮助系统提供了更多的指南。

William Ford
William Topp

目 录

第 1 章 类与对象	1
1.1 本书内容	1
1.1.1 动态数组	1
1.1.2 链表	2
1.1.3 关联数组	3
1.1.4 图	4
1.1.5 适配器结构	4
1.1.6 数据结构与面向对象编程	5
1.1.7 数据结构与算法	5
1.2 面向对象编程	5
1.3 理解类的概念	7
1.4 Time24 类	8
1.4.1 设计 Time24 类	8
1.4.2 构造函数	9
1.4.3 toString 方法	10
1.4.4 Accessor 和 mutator 方法	11
1.4.5 静态方法	12
1.5 对象的声明和使用	12
1.5.1 对象方法的使用	13
1.5.2 引用与别名	14
1.6 一个使用 Time24 对象的应用程序	15
1.7 类的表示	16
1.8 类的实现	18
1.8.1 Time24 类声明	19
1.8.2 私有的工具方法	20
1.8.3 accessor 与 mutator 方法	21
1.8.4 构造函数	22
1.8.5 格式化的对象描述	23
1.8.6 增加时间	23
1.8.7 时间间隔	23
1.9 类的构造	24
1.10 String 类	26

1.10.1 串索引	27
1.10.2 串合并	27
1.10.3 串比较	28
1.11 枚举类	29
1.12 总结	32
书面练习	33
编程练习	34
编程项目	36
第 2 章 类之间的关系	39
2.1 wrapper 类	40
2.1.1 Integer 对象的比较	40
2.1.2 静态 wrapper 类成员	41
2.1.3 字符处理	42
2.2 自动装箱与自动拆箱	43
2.3 对象组合	44
2.3.1 TimeCard 类	45
2.3.2 TimeCard 类的实现	46
2.3.3 TimeCard 类的 UML 图	47
2.4 Java 中的继承	47
2.4.1 一个雇员层次结构	48
2.4.2 继承层次结构中成员的可见性	49
2.4.3 Employee 类的声明	50
2.4.4 SalaryEmployee 子类	51
2.4.5 继承层次结构中的关键字 super	52
2.4.6 HourlyEmployee 子类	53
2.5 多态性	55
2.6 抽象类	59
2.7 运行时错误的处理	60
2.7.1 throw 语句	60
2.7.2 处理异常的 try/catch 代码块	61
2.7.3 finally 子句	62

2.7.4	异常传播	63
2.7.5	Java 异常的层次结构	63
2.7.6	标准的异常	65
2.8	输入与输出	65
2.8.1	控制台 I/O	66
2.8.2	文件 I/O	67
2.8.3	使用 Reader 流的文本输入	67
2.8.4	输入串的分析	69
2.8.5	使用 Writer 流的文本输出	71
2.8.6	输出流的控制	72
2.9	Scanner 类	73
2.9.1	声明 Scanner 对象	73
2.9.2	输入流的读取	73
2.9.3	文件输入	75
2.9.4	Scanner 类 API	75
2.9.5	应用程序: Scanner 类的使用	76
2.10	总结	79
	书面练习	80
	编程练习	83
	编程项目	87
第 3 章	类的设计	89
3.1	Java 接口	89
3.2	作为模板的接口	91
3.2.1	使用接口类型	92
3.2.2	接口与继承	94
3.3	使用 javadoc 创建 API	96
3.4	设计模式	99
3.5	GUI 应用程序设计	100
3.5.1	图形组件	101
3.5.2	GUI 应用程序设计模式	102
3.5.3	事件侦听器与事件处理程序	105
3.5.4	骰子投掷动作事件	106
3.6	总结	107
	书面练习	108
	编程练习	108
	编程项目	112

第 4 章	算法介绍	115
4.1	选择排序	115
4.2	简单的搜索算法	119
4.2.1	顺序搜索	119
4.2.2	二叉搜索	121
4.3	算法的分析	126
4.3.1	系统/内存性能标准	126
4.3.2	算法性能: 运行时间分析	126
4.3.3	Big-O 符号	129
4.3.4	通用数量级	131
4.4	搜索算法的比较	133
4.5	总结	136
	书面练习	136
	编程练习	138
	编程项目	141
第 5 章	泛型类与方法	143
5.1	Object 超类	144
5.1.1	对象数组方法	145
5.1.2	通用的顺序搜索	146
5.2	Java 泛型介绍	147
5.2.1	基于 Object 的 Store 类	148
5.2.2	泛型集合	149
5.2.3	泛型 Store 类	150
5.3	泛型接口	151
5.4	泛型方法	155
5.4.1	基本的泛型方法	155
5.4.2	为泛型类型使用绑定	156
5.5	泛型与继承	157
5.5.1	被绑定的泛型类型	158
5.5.2	泛型与通配符	160
5.6	泛型搜索/排序算法	161
5.7	总结	164
	书面练习	165
	编程练习	166
	编程项目	168
第 6 章	递归	171
6.1	递归的概念	171
6.1.1	描述一个递归算法	173
6.1.2	递归方法的实现	173

6.1.3 递归的工作方式	175	8.2.6 Java Collections Framework	236
6.1.4 多种基数表示法	176	8.3 Bag 集合	237
6.2 递归的应用	179	8.3.1 创建和使用一个 Bag 集合	238
6.2.1 构建一个刻度尺	179	8.3.2 应用: Eratosthenes 筛选法	240
6.2.2 Hanoi 塔	181	8.4 Bag 类的实现	243
6.3 递归的评价	185	8.4.1 私有的 remove() 方法	244
6.3.1 Fibonacci 方法	186	8.4.2 插入和访问方法	245
6.3.2 使用递归的标准	189	8.4.3 集合的 toString() 方法	246
6.4 总结	189	8.5 总结	246
书面练习	190	书面练习	247
编程练习	192	编程练习	248
编程项目	194	编程项目	249
第 7 章 排序算法	195	第 9 章 基于数组的列表集合	251
7.1 插入排序	195	9.1 列表集合	251
7.2 分治排序算法	198	9.2 ArrayList 类	255
7.2.1 归并排序	198	9.2.1 调整 ArrayList 的大小	257
7.2.2 通用的排序方法	201	9.2.2 ArrayList API	259
7.2.3 msort() 方法	202	9.3 ArrayList 应用程序	259
7.2.4 归并排序的效率	205	9.3.1 ArrayList 的连接	259
7.3 快速排序	206	9.3.2 closest-pair 问题	262
7.3.1 使用基准值分割列表	206	9.4 实现 ArrayList 类	266
7.3.2 快速排序递归下降	209	9.4.1 ArrayList 类的设计	266
7.3.3 pivotIndex() 方法	211	9.4.2 准备更大的容量	267
7.3.4 quicksort() 方法	213	9.4.3 添加和删除元素	268
7.3.5 快速排序的运行时间	215	9.4.4 实现索引访问	272
7.3.6 排序算法的比较	216	9.5 Cloneable 对象	273
7.4 第 k 大元素的查找	219	9.5.1 克隆 Time24 对象	274
7.5 总结	221	9.5.2 克隆引用变量	275
书面练习	222	9.5.3 克隆 ArrayList 对象	277
编程练习	223	9.5.4 克隆数组	279
编程项目	225	9.6 ArrayList 集合的评价	279
第 8 章 集合类型	227	9.7 总结	280
8.1 集合介绍	227	书面练习	280
8.2 集合的概述	229	编程练习	282
8.2.1 List 集合	230	编程项目	285
8.2.2 Set 集合	232	第 10 章 链表	287
8.2.3 Map 集合	233	10.1 单链表	289
8.2.4 适配器集合	234	10.1.1 创建链表	289
8.2.5 Graph 集合	235	10.1.2 扫描链表	291