

Java Software Structures: Designing and Using Data Structures, Second Edition

Java软件结构: 设计和使用数据结构 (第2版)

John Lewis 著
Joseph Chase
任学群 译



清华大学出版社

国外经典教材·计算机科学与技术

Java 软件结构：设计和使用数据结构

（第 2 版）

John Lewis
Joseph Chase 著

任学群 译

清华大学出版社
北 京

Simplified Chinese edition copyright © 2005 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

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

Original English language title: Java Software Structures: Designing and Using Data Structures, 2nd by Lewis & Chase, Copyright © 2005

EISBN: 0-321-24584-9

All Rights Reserved.

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

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 Addison-Wesley 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: 01-2005-2876 号

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

Java 软件结构: 设计和使用数据结构 (第 2 版) / (美) 里维斯 (Lewis, J.), (美) 蔡斯 (Chase, J.) 著; 任学群译. —北京: 清华大学出版社, 2005.8

书名原文: Java Software Structures: Designing and Using Data Structures

ISBN 7-302-11263-0

I. J… II. ①里… ②蔡… ③任… III. ①JAVA 语言—程序设计 ②数据结构 IV. ①TP312 ②TP311.12

中国版本图书馆 CIP 数据核字 (2005) 第 068866 号

出版者: 清华大学出版社 地 址: 北京清华大学学研大厦
http://www.tup.com.cn 邮 编: 100084
社 总 机: 010-62770175 客户服务: 010-62776969

责任编辑: 常晓波

印刷者: 北京市世界知识印刷厂

装订者: 三河市李旗庄少明装订厂

发行者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 28.75 字数: 713 千字

版 次: 2005 年 8 月第 1 版 2005 年 8 月第 1 次印刷

书 号: ISBN 7-302-11263-0/TP · 7431

印 数: 1 ~ 3000

定 价: 56.00 元

John Lewis 著
Joseph Chase

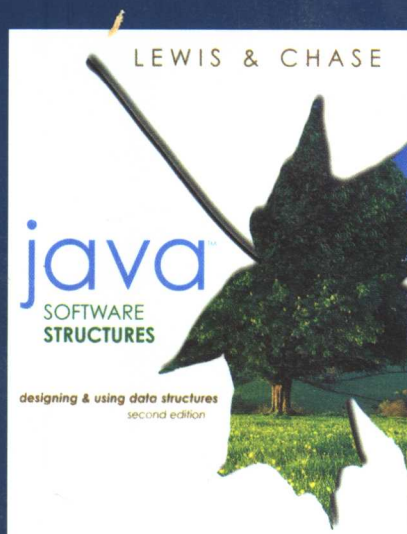
作者简介

“作者完美地涵盖了各种各样的经典数据结构。本书并不是简单地提及Java集合中包含的数据结构，而是引领学生深入各种结构的设计和创建。随着章节的引申，不断挑战学生，让他们独立完成不断递增的结构。”

——Peter J. DePasquale
新泽西大学

“本书是数据抽象、封装及责任分割等的绝佳范例——尽管这些概念在课堂上难以讲授。本书还自发地说明了这些概念如何应用于贯穿整个学期的各种数据结构。”

——Manuel A. Pérez-Quinones
弗吉尼亚科技大学



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學思想和教學方法，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科学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了审读，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不足，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

序 言

本书适合做数据结构及算法课程教科书之用。该课程常常被称为 CS2 课程，因为其通常是作为中级计算机专业课程而设置的。本书满足了计算机课程计划 2001（Computing Curricula 2001，CC2001）的原则。

从教学的角度看，本书仿效了 *Java Software Solutions: Foundations of Program Design*（作者 John Lewis 与 William Loftus）该书的一流 CS1 课本的教授风格和方法。本书拥有许多在该书中受到很高评价的特色，比如关键概念框和完整代码实例。作为计算机专业学生的两门或者三门入门系列课程，这两本书一起为此提供了一个坚实系统的方案。也就是说，本书不需要要求学生在以前的课程中已经使用过 *Java Software Solutions*。

可能在任一课程中出现的内容（比如递归和排序），本书也同样做了阐述。本书首先对面向对象的概念及其在 Java 中如何实现做了一个全面概述。该章可以做复习之用，也可以用来让不同背景的学生赶上最新的进度。

我们知道数据结构和算法内容在课程中的关键作用，同时我们认为本书较好地满足了该门课的要求。

本书第二版

在第二版中我们对本书做了许多改进和提高。最显著的是所有集合（collection）都是用泛型（generics，一种最近添加到 Java 5.0 中的高效工具）设计和实现的。在这门课中我们还强化了面向对象设计的重要性，据于此，我们增加了用来展现完整软件系统设计及实现的 4 个案例。

我们的方法

这种类型的书籍在整体方法方面差别相当大。我们的方法是建立在一些我们强烈遵循的重要原则之上的。首先，我们连贯一致地叙述我们在本书中将要考察的各种集合。第二，我们强调健全的软件设计技巧的重要性。第三，我们对本书结构加以组织以突出和强化本书的重要目标：数据结构和算法学习。我们将更深入地考察这些原则。

叙述连贯

在考察某一特定类型的集合时，将按照以下顺序仔细处理每一问题。

1. 概念：我们会从概念上讨论集合，并确定它所提供的服务（其接口）。
2. 用途：我们将考察有关集合的特定属性在解决问题时是如何派上用场的例子（不管集合是如何被实现的）。
3. 实现：我们将考察集合的各种实现选项。
4. 分析：我们对各种实现进行了比较和区分。

恰当的时候，我们将讨论 Java Collections API。如果有必要讨论 API 中某个特定集合类型，那么我们将讨论该集合及其实现。因此我们利用 API，但并不完全局限于它。并且我们会毫不犹豫地指出其缺陷。

我们将分析步骤置于了一个十分重要的地位。在第 1 章建立 Big-Oh 记法的概念并在全书中应用，但这更多地是直觉上而不是数学上的分析。

健全程序设计

全书始终将健全软件工程实践置于一个高度优先的地位。我们对集合实现以及对使用它们的程序的设计遵循一致和恰当的标准。

将集合的接口与其底层实现分隔开具有非常大的重要性。某一集合提供的服务通常在 Java 接口中有着正式定义。为了强化该集合为一个抽象的思想，只要合适，我们都将接口名称用做该集合的类型指定。

除了实践坚实的设计原则外，我们还在全书的论述中对其进行了强调。我们努力尝试既通过示例又通过不断强化的方式来进行教学。

组织结构清晰

我们对本书内容进行了细致地组织以将偏离主题带来的干扰降到最低，并对本书的总体目标予以了强化。本书的组织使得本书既可以充当数据结构和算法的教学用书，又可以作为一本颇有价值的参考用书。

本书划分为众多部分：第一部分包括前两章，它提供了有关面向对象设计和 Java 的背景知识。第二部分包括随后的 6 章，它覆盖了影响数据结构和算法的所有方面的入门和深层问题。第三部分涉及线性集合（栈、队列和列表）。第四部分涉及非线性集合（树、堆、散列和图）。除了树之外的每一个集合类型都自成一章。有关树的内容分布于一系列章节，用于考察其各个方面及各种目的。

分布于全书每一章节都是 4 个案例，每个案例都给出了一个完整的例子，包括从问题陈述到设计原理，以及完整的实现过程。

章节划分

第 1 章（软件开发）讨论了软件质量涉及的各个方面并对软件开发问题进行了一个全面概述。该章用于在着手数据结构和算法设计的细节之前建立一套正确的开发思想。该章同样介绍了支撑算法分析的基本概念。

第 2 章（面向对象设计）为任何需要回顾面向对象的基本概念及其在 Java 中如何实现等内容的读者提供参考。该章包含的概念包括抽象、类、封装、继承、多态性以及许多相关的 Java 语言结构，比如接口。

第 3 章（集合）建立了集合的概念，强调了将接口和实现区分开来的必要性。比如，它介绍了一个 set（集）集合并讨论了某一基于数组的实现。

第 4 章（链式结构）讨论了使用引用来创建链式数据结构。该章考察了有关链表管理的基本问题，然后使用底层链式数据结构（将在第 3 章介绍）对 set 集合的其他实现进行了定义。

第 5 章（Black Jack）给出了一个 Black Jack 游戏程序的完整案例，该程序应用了第 3 章和第 4 章中定义的 set 集合。

第 6 章（堆栈）开始了研究特定集合的一系列章节。我们从相当直观的集合——栈开始，从概念和实现两个角度考察。该章既考察了基于数组的实现也考察了链实现方式，然后对其进行了比较。

第 7 章（队列）考察了先进先出队列的概念和实现。我们通过一个有效使用队列的例

子来讨论基数排序。所涉及的实现选项包括底层链表以及固定和环形数组。

第 8 章（列表）论述了 3 种类型的列表：有序表、无序表及索引表，通过讨论这 3 种类型列表所共有的及各自独有的操作来对它们进行比较和区分。在各种类型的列表设计中我们将恰当地使用继承，且通过基于数组的表示和链式表示两种方式实现这些列表。

第 9 章（计算器）给出了一个计算器程序的完整案例。

第 10 章（递归）对递归概念和递归解决方案有何种优越性做了一个一般性介绍。该章探讨了递归的实现细节并讨论了递归算法分析的基本思想。

第 11 章（排序和查找）讨论了线形及二分查找算法，以及若干排序的算法（选择排序、插入排序、冒泡排序、快速排序，以及归并排序）。本章着重讨论与查找和排序相关的编程问题，比如将 Comparable 接口（Comparable interface）用做对象比较的基础。以特定数据结构为基础的查找和排序（如堆排序）将在本书后面的适当章节讨论。

第 12 章（树）对树进行了一个全面概述，并确立了那些关键的术语和概念。本章讨论了各种实现方式并通过一个二叉树来表示和评估某一算数表达式。

第 13 章（二叉查找树）利用第 12 章确立的基本概念定义了一个经典的二叉查找树。本章先考察了一个二叉查找树的链式实现，然后讨论了树结点的平衡如何对其性能起到关键作用。这就通向了 AVL 及二叉查找树的红/黑实现的学习。

第 14 章（祖先树）给出了一个管理某一祖先树的程序的完整案例，该祖先树用于映射某一特定个人的血统。

第 15 章（堆）探讨了堆的概念、使用及实现。我们通过某个堆排序来举例说明其用途。链式实现和基于数组的实现都将被讨论到。

第 16 章（多路查找树）是前面章节的自然延伸。本章讨论了 2-3 树、2-4 树，以及广义 B-树，还讨论了实现选项。

第 17 章（散列）包括散列及相关问题的概念，比如散列函数及冲突。本章讨论了散列的各种 Java Collections API 选项。

第 18 章（图）探讨了无向图和有向图的概念并确立了一些重要的术语。本章考察了若干普通图算法并讨论了实现选项，包括邻接矩阵。

第 19 章（Web 爬虫工具）给出了一个在 Web 中搜索特定文本的 Web 爬虫工具程序的完整案例。

补充资料

本书的所有读者都可以在 www.aw-bc.com/cssupport 上获得以下补充资料。

本书中出现的所有程序的源代码。

以下教师补充资料仅限于满足条件的教师。请你与当地 Addison-Wesley 销售代表联系，或者发送电子邮件至 aw.cse@aw.com，咨询有关资料获取的信息。

本书部分练习题及编程项目的答案。

- 试题库，含有测验用的问题。
- 用于本书内容讲解的 PowerPoint 幻灯片。

感谢

首先且最为重要的是我们要感谢我们的学生，本书是为他们编写的，如果没有他们也就永远不会有这本书。你们的反馈有助于我们成为更优秀的教育家和作家。请大家继续关注我们的工作。

感谢下面列出的所有审阅人员，感谢他们花时间分享了他们对本书及前一版内容和资料编排上的深刻见解。他们的贡献是无价的。

Mary P. Boelk, Marquette University
Robert Burton, Brigham Young University
Robert Cohen, University of Massachusetts-Boston
Jack Davis, Radford University
Bob Holloway, University of Wisconsin-Madison
Nisar Hundewale, Georgia State University
Chung Lee, California State Polytechnic University
Mark J. Llewellyn, University of Central Florida
Ronald Marsh, University of North Dakota
Eli C. Minkoff, Bates College; University of Maine-Augusta
Ned Okie, Radford University
Manuel A. Perez-Quinones, Virginia Tech
Salam Salloum, California State Polytechnic University-Pomona
Don Slater, Carnegie Mellon University
Ashish Soni, University of Southern California

Addison-Wesley 公司的朋友们也对本书的出版提供了全力支持并与我们一道促进了本书的面世。本书是一项真正的集体成果。高级组稿编辑 Michael Hirsch 和他的助理 Maria Campo 总是会随时随地地伸出援助之手。营销经理 Michelle Brown 和她的助理 Jake Zavracky，以及整个 Addison-Wesley 销售人员不知疲倦地加以工作以确保教师们能够掌握本书的目标和优点。尽管日程极其紧张，Patty Mahtani 还是完美无缺地使本书付梓面世，Joyce Wells 出色的封面和内部设计也备受赞誉。Argosy Publishing 公司的 Daniel Rausch 和 Edalin Michael 也向他们提供了卓越的技术支持。Caroline Fell 总是可以想到办法让本书按时印刷，所以我们这本书才能在你需要在课堂上使用时及时到达你的手上。非常感谢你们为本书所付出的全部辛勤工作和贡献。

不应该疏忽的是，我们还应该感谢 ACM 计算机科学教育特别兴趣组（ACM Special Interest Group on Computer Science Education）的卓越贡献，其出版物和会议对于每一个认真对待计算机教学的人来说都具有极其重要的作用。如果你还不是这个小组的成员，那你真是错过了一个很好的机会。Kentucky SIGCSE 讨论会（SIGCSE Symposium）上自发形成的、涉及内容非常广泛的小组讨论也给我们提供了特别的帮助。

最后，我们还要感谢我们的家人，他们对我们投入的任何项目都给以了支持和鼓励。总之，你就是我们所做一切的原因所在。

目 录

第 1 章 软件开发	1
1.1 软件工程	1
专业水准与道德规范	2
1.2 软件质量	3
正确性	3
可靠性	3
健壮性	4
可用性	4
可维护性	4
可重用性	5
可移植性	5
效率	6
质量问题	6
1.3 开发生命周期模型	6
瀑布模型 (Waterfall Model)	8
螺旋模型 (Spiral Model)	9
演化开发模型	10
1.4 统一建模语言 (Unified Modeling Language, UML)	10
UML 关系	11
1.5 错误处理	14
1.6 算法分析	15
增长函数 (Growth Function) 及 Big O() 记法	15
增长函数的比较	16
循环执行分析	17
嵌套循环	18
1.7 软件工程与数据结构	19
第 2 章 面向对象设计	23
2.1 面向对象内容综述	23
2.2 对象的使用	24
抽象	24
对象的创建	25
2.3 类的库与包	26
import (导入) 声明	27
2.4 状态与行为	28
2.5 类	29
实例数据	31
2.6 封装	31
可见性修饰符	32
局部数据	33
2.7 构造函数	34
2.8 方法重载	34
2.9 再论引用	35
空 (null) 引用	36
this 引用	36
别名	38
垃圾回收	39
将对象作为参数传递	40
2.10 static (静态) 修饰符	40
静态变量	40
静态方法	41
2.11 包装类	41
2.12 接口	42
Comparable 接口	44
Iterator 接口	44
2.13 继承	44
派生类	45
protected (保护) 修饰符	47
super (超级) 引用	47
方法的重写	48
2.14 类的层次结构	48
Object 类	49
抽象类	50
接口的层次结构	51

2.15 多态	52	find、contains 的分析	90
引用和类的层次结构	53	union 的分析	90
基于继承的多态	53	equals 的分析	91
基于接口的多态	55	第 4 章 链式结构	94
2.16 泛型	56	4.1 用做链接的引用	94
2.17 异常	57	4.2 管理链表	96
异常消息	58	插入节点	96
try 语句	58	删除节点	97
异常传播	59	哑节点	98
异常类的层次结构	60	4.3 没有链接的元素	98
第 3 章 集合	66	双向链表	99
3.1 集合介绍	66	4.4 实现 set: 利用链接	99
抽象数据类型	67	LinkedSet 类	100
Java 集合 API	69	add 运算	102
3.2 set 集合	69	removeRandom 运算	103
接口	70	remove 运算	104
迭代器	72	iterator 运算	105
异常	72	4.5 LinkedSet 的分析	107
3.3 使用 set: bingo	73	add 的分析	107
3.4 实现 set: 利用数组	76	remove 的分析	108
容量的管理	77	removeRandom 的分析	108
3.5 ArraySet 类	77	第 5 章 Black Jack	111
size 运算和 isEmpty 运算	79	5.1 Black Jack 游戏	111
add 运算	80	5.2 初始设计	111
addAll 运算	82	5.3 实现 Black Jack 游戏	115
removeRandom 运算	82	Card 类	115
remove 运算	83	Deck 类	117
union 运算	84	Hand 类	122
contains 运算	85	BlackJack 类	125
equals 运算	85	BlackJackGUI 类	128
iterator 运算	86	BlackJackDemo 类	135
toString 运算	88	第 6 章 堆栈	137
UML 描述	88	6.1 堆栈 ADT	137
3.6 ArraySet 的分析	88	6.2 堆栈的使用: 计算后缀表达式 的值	140
add 的分析	89	6.3 堆栈的使用: 穿越迷宫	146
remove 的分析	90	6.4 堆栈的实现: 链接实现	151
removeRandom 的分析	90	push 操作	151
addAll 的分析	90		

pop 操作	152	有序列表的添加 (add) 操作	206
其他操作	153	无序列表的特有操作	207
6.5 堆栈的实现: 数组实现	153	索引列表的特有操作	207
push 操作	154	8.5 实现列表: 利用链接	208
pop 操作	154	remove 操作	208
其他操作	155	双向链表	209
6.6 堆栈的实现: java.util.Stack 类	156	8.6 列表的实现分析	212
惟一操作	156	有序列表的实现分析	214
继承和实现	156	无序列表的实现分析	214
6.7 堆栈实现的分析	157	索引列表的实现分析	216
push 操作的分析	157		
pop 操作的分析	158	第 9 章 计算器	220
第 7 章 队列	162	9.1 计算器	220
7.1 队列 ADT	162	9.2 初始设计	220
7.2 使用队列: 代码密钥	164	9.3 实现计算器	224
7.3 使用队列: 售票口模拟	167	infixToPostfix 类	224
7.4 使用队列: 基数排序	171	listPostfixEvaluator 类	228
7.5 实现队列: 利用链接	175	CalculatorGUI 类	230
enqueue 运算	175	CalculatorDemo 类	235
dequeue 运算	176	第 10 章 递归	238
其他运算	178	10.1 递归地思考	238
7.6 实现队列: 利用数组	178	无穷递归	239
enqueue 运算	178	数学中的递归	240
dequeue 运算	179	10.2 递归地编程	240
其他运算	180	递归与迭代	242
7.7 实现队列: 利用环形数组	181	直接递归与间接递归	243
7.8 分析各种队列实现	184	10.3 使用递归	243
enqueue	184	遍历迷宫	243
递增 count	184	河内塔	247
dequeue	185	10.4 分析递归算法	251
第 8 章 列表	190	第 11 章 排序和查找	256
8.1 列表 ADT (Abstract Data Type, 抽象数据类型)	190	11.1 查找	256
8.2 运用有序列表: 锦标赛主办者	196	线性查找法	257
8.3 运用索引表: Josephus 问题	201	二分查找法	258
8.4 实现列表: 利用数组	203	查找算法的比较	260
remove 操作	204	11.2 排序	261
Contains 操作	206	选择排序法	263
		插入排序法	265
		冒泡排序法	266

快速排序法	268	AVL 树中的右左旋	318
归并排序法	270	AVL 树中的左右旋	318
第 12 章 树	275	13.6 实现二叉查找树: 红黑树	319
12.1 树	275	红黑树中的插入	320
树的分类	276	红黑树中的元素删除	322
12.2 树的实现策略	278	13.7 实现二叉查找树: Java 集合	
树的数组实现之计算策略	278	API	324
树的数组实现之模拟链接策略	279	第 14 章 祖先树	331
树的分析	279	14.1 祖先树	331
12.3 树的遍历	280	14.2 初始设计	331
前序遍历	280	14.3 祖先树的实现	335
中序遍历	281	Person 类	335
后序遍历	282	AncestorTreeNode 类	335
层序遍历	282	AncestorTree 类	336
12.4 二叉树的实现	283	AncestorGUI 类	345
removeLeftSubtree 方法	288	AncestorDemo 类	358
find 方法	289	第 15 章 堆	361
iteratorInOrder 方法	290	15.1 堆	361
12.5 二叉树的使用: 表达式树	291	addElement 操作	362
第 13 章 二叉查找树	302	removeMin 操作	363
13.1 二叉查找树	302	findMin 操作	364
13.2 实现二叉查找树: 利用链接	304	15.2 堆的使用: 堆排序	365
addElement 运算	304	15.3 堆的使用: 优先级队列	365
removeElement 运算	306	15.4 堆的实现: 利用链接	368
removeAllOccurrences 运算	309	addElement 操作	368
removeMin 运算	310	removeMin 操作	371
13.3 使用二叉查找树: 实现有序		findMin 操作	373
列表	310	15.5 堆的实现: 利用数组	373
BinarySearchTreeList 实现的		addElement 操作	373
分析	313	removeMin 操作	374
13.4 平衡二叉查找树	314	findMin 操作	376
右旋	314	15.6 堆实现的分析	376
左旋	315	addElement 操作	376
右左旋	315	removeMin 操作	376
左右旋	316	findMin 操作	377
13.5 实现二叉查找树: AVL 树	317	堆排序	377
AVL 树中的右旋	317	第 16 章 多路查找树	381
AVL 树中的左旋	317	16.1 整合树的概念	381

16.2	2-3 树	381	Hashtable 类	404
	向 2-3 树中插入元素	382	HashSet 类	405
	从 2-3 树中删除元素	384	HashMap 类	406
16.3	2-4 树	387	IdentityHashMap 类	406
16.4	B 树	388	WeakHashMap 类	407
16.5	创建和使用 B 树的动因	388	LinkedHashSet 与	
	B*树	389	LinkedHashMap	408
	B+树	389		
	B 树的分析	390		
16.6	B 树的实现策略	390		
第 17 章 散列 (Hashing)		394	第 18 章 图	413
17.1	散列	394	18.1 无向图	413
17.2	散列函数	396	18.2 有向图	415
	除法方法	396	18.3 网络	416
	折叠法	397	18.4 常见的图算法	417
	平方取中间值法	397	遍历	417
	基数转换法	397	测试连通性	421
	数字分析法	398	最小生成树	422
	长度依赖法	398	判定最短路径	424
	Java 语言中的散列函数	398	18.5 图的实现策略	425
17.3	解决冲突	398	邻接表	425
	链锁法	399	邻接矩阵	426
	开地址	400		
17.4	从散列表删除元素	402	第 19 章 Web 爬虫工具	430
	从链锁式实现中删除	402	19.1 Web 爬虫工具	430
	从开地址实现中删除	403	19.2 初始设计	430
17.5	Java 集合 API 中的散列表	403	19.3 实现 Web 爬虫工具	432
			WebCrawler 类	433
			WebcrawlerGUI 类	439
			WebCrawlerDemo 类	441

第 1 章 软件开发

本章学习目标

- 论述软件开发的目标
- 明确软件质量考核的各个方面
- 考察各种软件开发生命周期模型
- 探究统一建模语言（Unified Modeling Language, UML）的记法
- 考察有关错误处理的问题
- 介绍算法分析的概念

我们将对软件工程的底层原理进行一个全面概述，并以此开始我们对软件开发的探究。我们力求的是设计软件，而不仅仅是编写程序。我们希望开发出可以经受住用户以及时间考验的高质量软件系统。遵循软件工程的原则就会使我们达到这个目标。本章将讨论一系列与软件开发相关的问题，并创立一些在数据结构和软件开发研究中甚为重要的术语。

1.1 软件工程

设想这样一个场景：你正在走近大河上最近修建的一座桥。在靠近这座桥时，你看到一个标牌，它告诉你这座桥是由当地建筑工人设计并修建的，该工程没有工程师的参与。你会打算继续过桥吗？如果该标牌告诉你这座桥是由工程师设计然后由建筑工人建造的话，那么你会不会做出不同的选择？

本文中的“工程师”一词指的是受过工程学历史、理论、方法和实践教育的人。该定义包括诸如电气工程、机械工程，及化学工程等学科领域。软件工程是一门关于高质量软件开发技术和理论的学科。

当“软件工程”这一术语于 20 世纪 70 年代首次被提出的时候，它肩负的是一项目标——在该行业的领导者们意识到本行业创造的许多软件的质量都甚为低劣时，他们定下了这一目标。他们希望开发人员放弃编写程序这种过分简单的想法，转向软件工程这种更为训练有素的思想。为了开发软件我们首先必须明白该术语不仅仅是一个名称——事实上它代表的是一种完全不同的态度。

对于软件工程是否已经成为一门真正的工程科学存在许多争论。我们在软件工程课程中将不会涉及这些争论。作为软件开发人员，我们拥有共同的历史，我们受共同的理论约束，以及为了协同工作我们必须熟知当前的实践方法，明白这些就够了。

最后，我们希望让客户满意，这些客户包括为所开发软件付费的个人或组织，以及系统的最终用户（根据具体情况也可能包括我们的客户）。

软件工程的目标很大程度上和其他工程学科是一样的。

- 解决特定的问题
- 按时且在预算内给出解决方案
- 给出高质量的解决方案

听起来奇怪的是，我们还需担心所解决的问题会不会是个错误的问题，这一问题几乎会给每个项目带来麻烦。常常会有软件开发人员在交付产品后竟然发现该产品与客户的要求并不完全一致，这种例子数不胜数。因此，任何软件开发过程中的先前步骤之一就是确保完全了解我们所要解决的问题细节。因此，我们必须为解决问题所要满足的需求开发出一套准确的规范。

问题分析包括诸如访问客户、观察现存的处理过程，以及分析现存的解决方案。通过这些活动开发出的需求不仅包括建立该解决方案所需函数，比如允许某一用户使用某一用户名和口令登录某一系统，也包括建立这些函数约束条件以及开发这些约束条件，比如对什么字符可以用做合法口令的规范。通过理解掌握所需解决的问题，我们就能够开发出一项更加恰当地解决特定问题的解决方案。

关键概念

软件开发的第一步是分析问题并开发出一套全面准确的需求集。

专业水准与道德规范

作为专业人员，如果我们同意了某一客户对某一系统在某一特定日期以某一价格交付的需求，那么我们就有义务按时且在预算内交付产品。显然，如果某一企业不断地让其客户感到失望，那么这一企业的生存时间就不可能长久。但是这一问题已超出了现实利益的范畴。工程学科不可或缺的一部分就是要求我们能够做出精确的计划、进度表以及预算。不能按期在预算内交付产品不仅会给某一公司或某些相关公司带来严重损害，而且还会损害我们作为一个行业所共有的形象。

真正的软件工程师都遵循着这种包含能力概念在内的道德准则，如果我们认为项目无法按照客户的需求来完成，那么我们有责任在建立客户需求的时候就将这一情况告知客户。

关键概念

基于现实利益和道德规范两个原因，一名软件工程师必须力求按时和在预算内交付产品。

为了最大化软件质量，首先我们必须认识到质量对于不同客户有不同的含义。我们将在下一节讨论软件质量的问题。