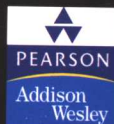




国外经典教材·计算机科学与技术



Data Abstraction and
Problem Solving With C++
Walls and Mirrors (Fourth Edition)

数据抽象和问题求解
——C++语言描述 (第4版)

(美) Frank M. Carrano 著
郭平 张敏 译
曹蓉蓉 宋红 审校



清华大学出版社

国外经典教材·计算机科学与技术

数据抽象和问题求解

——C++语言描述

(第4版)

(美) Frank M. Carrano 著
郭平 张敏 译
曹蓉蓉 宋红 审校

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: Data Abstraction and Problem Solving with C++: Walls and Mirrors,
fourth edition, 2005 by Frank M. Carrano, Copyright © 2005

EISBN: 0-321-24725-6

All Rights Reserved.

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

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: 01-2004-5629

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

数据抽象和问题求解——C++语言描述(第4版)/(美)卡雷诺(Carrano, F.M.)著; 郭平 张敏 译; 曹蓉蓉
宋红 审校. —北京: 清华大学出版社, 2005.11

书名原文: Data Abstraction and Problem Solving with C++: Walls and Mirrors, fourth edition
(国外经典教材·计算机科学与技术)

ISBN 7-302-11869-8

I. 数… II. ①卡… ②郭… ③张… ④曹… ⑤宋… III. C语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 109679 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

组稿编辑: 曹 康

文稿编辑: 徐燕萍

封面设计: 久久度文化

版式设计: 康 博

印刷者: 北京密云胶印厂

装订者: 北京国马印刷厂

发行者: 新华书店总店北京发行所

开 本: 185×260 印张: 45.25 字数: 1158 千字

版 次: 2005 年 11 月第 1 版 2005 年 11 月第 1 次印刷

书 号: ISBN 7-302-11869-8/TP·7714

印 数: 1~4000

定 价: 79.80 元

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和改革,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学需要。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

本书自第3版以来，我们积累了一些使用C++以面向对象的方式教授数据抽象的经验。第4版就反映了这些经验和C++的一些变化。

本书基于 Paul Helman 和 Robert Veroff 最初编著的 *Intermediate Problem Solving and Data Structures: Walls and Mirrors* (Benjamin/Cumming 公司, 1986 年), 继承了原著的组织方式和整体理念, 包括技术要点与正文内容、示例、图和练习题。Paul Helman 和 Robert Veroff 教授引入了两个内涵丰富的词: 墙和镜子, 方便了计算机科学的教學。

本书主要论述数据抽象和其他解决问题的工具, 是计算机科学的一门核心课程。考虑到计算机科学的动态性和多样性, 本书涵盖各种主题, 以求尽量适用于不同课程的教学要求。例如, 可将本书用作初级数据结构教材, 也可用作高级编程和问题求解教材。本书旨在使学生切实了解和掌握数据抽象、面向对象编程及其他主流的问题求解技术。

致读者

“墙”和“镜子”代表基本问题解决技术。“数据抽象”将模块实现细节与程序的其余部分隔离, 就像一堵将您和邻居隔开的墙。“递归”是重复技术, 通过解决同类型的更小问题来解决问题, 就像各映像逐渐变小的镜子。

本书在编写时充分考虑了学生的需求。作者以前也是大学生, 目前一直在从事教学工作, 很明白清晰表述的重要性。本书在风格上力求明晰精练, 通俗易懂。为了帮助学生学习本书, 并通过练习进行复习, 各章添加了小结、自我测试题及答案, 末尾有一个术语表。附录 A 提供 C++ 参考资料。在编程过程中, 附录和正文中还介绍了一些 C++ 辅导材料, 以供帮助。后面“教学特点”一节还列出了本书的主要特性。

本书对学生掌握的 C++ 知识做了一些基本的假定。学生可以参阅本书的附录 A 复习 C++ 语言。了解选择语句 if 和 switch; 迭代语句 for、while 和 do; 函数和参数传递; 数组、字符串、结构和文件。本书在第 1、3 和 8 章介绍了 C++ 类, 因为本书假定学生还没有学过 C++ 类, 也没有使用递归函数的经验, 所以在第 2 和 5 章讲述了递归函数。

本书的所有 C++ 源代码学生都可以使用。后面的辅助材料将说明如何得到这些文件。但要注意教师可能会给学生提供这些文件。

致教师

这一版使用 C++ 来强调数据抽象和数据结构这个重点, 仔细考虑了 C++ 的优缺点, 并采用相应的教学方法, 使低年级的学生能够理解本书的内容。

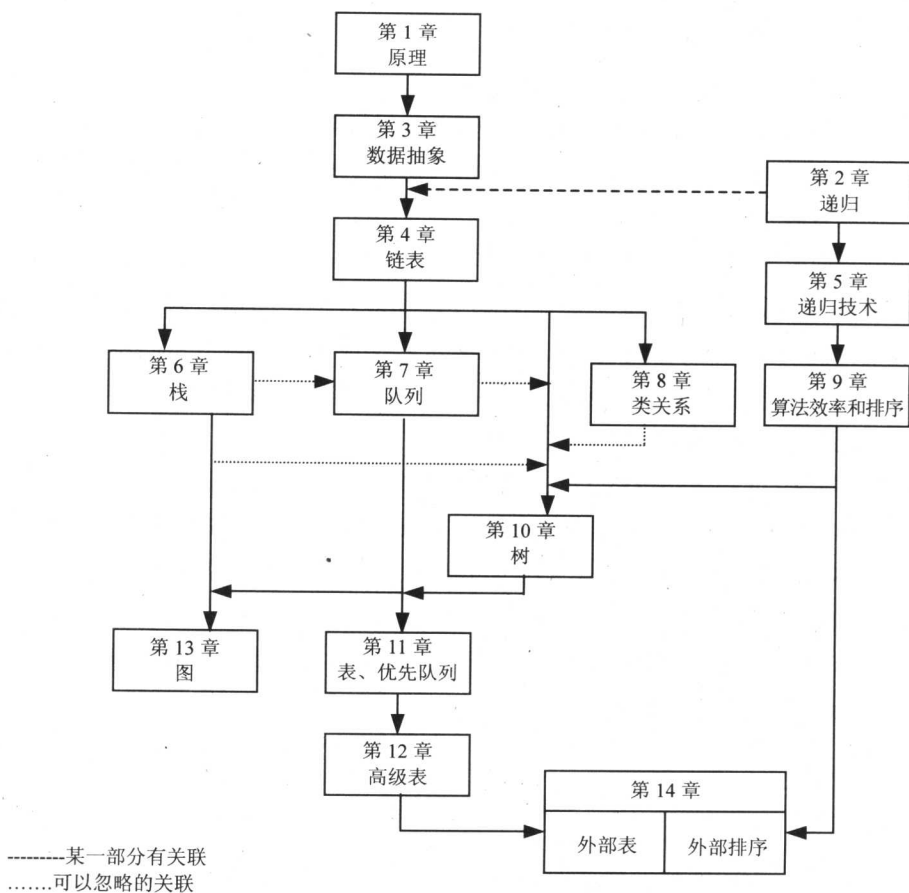
先决条件

我们假定读者有一定的 C++ 基础, 或者了解另一种语言, 并有教师帮助他们使用所提供的附录学习 C++。本书会详细介绍 C++ 类, 所以假定读者不了解这些知识。本书包括面向对象编程的基本概念、继承、虚函数和类模板, 以及 C++ 的所有内容。但本书只介绍与抽象数据类型 (ADT) 作为类来实现相关的主题, 重点仍是 ADT, 而不是 C++。我们会涉及基于对象的编程, 但假定将来的课程会详细介绍面向对象的设计和软件工程, 所以重点仍是数据抽象, 但把统一建模语言 (UML) 作为一个设计工具来介绍。

安排灵活

本书的扩展提供了本课程所需要的一些内容。学生可以根据自己的课程需要选择学习一些主题。下面列出关联图, 显示某一章的预备章节。

在第 1 部分, 可以根据学生的背景选择主题。这个部分的 3 章详细介绍了数据抽象和递归。这两个主题都非常重要, 先学习哪个主题有许多不同的选项。在本书中, 递归一章的前后都是讨论数据抽象的章节, 但学生可以重新安排学习顺序。



第 2 部分亦如此。例如，可将第 6 章(栈)排在第 8 章(高级 C++)的全部或部分内容之前或之后。在第 5 章后，可任意安排第 9 章(算法效率和排序)。可将树放在队列之前，将图放在表之前；在讲授表后，将按任意顺序安排散列、平衡二叉查找树或优先队列。可提前讲授第 14 章的外部方法，例如，可在第 9 章的归并排序后讲授。

数据抽象

在本书介绍的问题求解方法中，普遍使用了抽象数据类型(ADT)。一些例子说明如何将设计 ADT 作为解决方案总体设计的一部分。对于所有 ADT，首先用英语和伪码编写规范，然后将 ADT 用于简单应用程序，最后考虑实现。ADT 与数据结构的区别一直是中心议题。本书前面介绍了封装和 C++类，以演示 C++类如何对 ADT 的客户程序隐藏数据结构。诸如列表、栈、队列、树、表、堆和优先队列的抽象数据类型是讨论的重点。

问题求解

本书介绍计算机科学家的思考过程及所用技术，讲述如何整合问题求解和编程能力。学习计算机科学家如何开发、分析和实现解决方案与学习算法机制同等重要。

在示例问题的上下文中，包含开发方案的分析技巧。在设计问题的解决方案时，广泛采用抽象、算法与数据结构的逐步完善以及递归。

C++指针和链表处理过程在前面介绍，并用于建立数据结构。第 9 章简介算法的数量阶分析。先定性，后定量地比较基于数组和基于指针的数据结构。各种可能解决方案和实现的交替使用是问题求解的中心内容。

最后，在实现和验证解决方案时，编程风格、包含初始条件和结束条件的文档记录、调试工具和循环不变式是问题求解方法学的重要部分。

应用程序

在本书的重要主题中，列举了一些经典应用程序。例如，折半查找、快速排序和归并算法提供了递归的重要应用，并引入了数量阶分析。诸如平衡查找树、散列和文件索引的主题继续搜索的讨论。在介绍外部文件时，又讨论了查找和排序。

首先在递归中介绍了识别和计算代数表达式的算法，后来又作为栈的应用讨论了这些问题。其他应用程序，如八皇后问题作为回溯例子，事件模拟作为队列的应用，图查找和遍历作为栈和队列的其他重要应用。

新添和修订内容

本版沿承了第 3 版的基本方法和理念。在介绍数据抽象和编程时，既作为一般概念，又在 C++环境中讨论。为达到更新和明晰的目的，对原来的内容做了适当增删和修订。

本书几项重要修订如下。

- 所有的 C++ 程序都经过测试和修订，确保遵循语言的最新 ANSI 标准。
- 添加了使用标准模板库的例子。
- 每章添加了新的练习。
- 包含了新的编程问题。
- 改进了本书的整体设计，提高了内容的可读性。

本书概览

本书层次分明，组织精当，符合教材特点。教师可按具体专业要求做适当调整。

本书特色

本书的特色如下，以便读者学习和复习。

- 每章有“本章概述”。
- 每章有“小节”。
- 每章有“提示”，指明常见错误。
- 每章有“自我测试题”，书末附有“自我检测题答案”。
- 每章有“练习题”和“编程问题”。非常困难的标有星号。答案在《教师资源手册》中。
- 用英语和伪码编写所有主要 ADT 的规范。
- 所有主要 ADT 的 C++ 类定义。
- 实例演示类和 ADT 在问题求解过程的作用。
- 附录 A 简介 C++。
- 书末有一个“术语表”。

编排形式

本书分两部分。一般而言，第 1~11 章是一学期的课程。第 1~2 章可作为简介资料。可根据课程在全部课程中的安排来选用第 11~14 章。《教师资源手册》阐述本书在不同课程的使用。

第 1 部分：问题求解技术。 第 1 章简要介绍编程和软件工程的主要问题，并引入了统一建模语言(UML)。第 2 章分析递归，供学生巩固基础；递归思维能力是计算机科学家必须掌握的最有用技术之一，对理解问题本质极具价值。第 5 章深入分析递归。本书列举了大量递归实例，范围很广，从简单递归定义，到语言识别、查找和排序的递归算法。

第 3 章详细讨论数据抽象和抽象数据类型。在讨论了 ADT 列表的规范和使用后，接着讨论 C++ 类，并使用它们来实现 ADT。本章还简要介绍了继承、C++ 命名空间和异常。第 4 章在讨论 C++ 指针变量和链表时论述了另一个实现工具，还讲述了类模板、C++ 标准模板库(STL)、容器和迭代器。

可根据学生背景，选择并按适当顺序讲授这些主题。

第 2 部分：用 ADT 解决问题。 第 2 部分一直将数据抽象作为问题解决技术。首先指定诸如栈、队列、二叉树、二叉查找树、表、堆和优先队列的基本 ADT，然后将 ADT 实现为类。

在实例中使用 ADT，并比较各种实现。

第 8 章进一步开发继承、类模板和迭代器，扩展了 C++ 类的内容。接着介绍了虚函数和友元。第 9 章引入数量阶分析和大 O 表示法，规范化以前讨论的算法效率，分析了递归归并排序、快速排序等几种查找和排序算法的效率。

第 2 部分还包括几个高级主题，如平衡查找树(2-3 树、2-3-4 树、红-黑树和 AVL 树)和散列，并用它们实现表。分析这些实现，确定它们最适合支持的操作。

最后分析外部直接访问文件的数据存储。修改归并排序来排序数据，用外部散列和 B-树索引执行查找。这些查找算法是内部散列方案和 2-3 树的泛化。

补充资料

在 www.aw-bc.com.support 上可获得以下资料。

- 源代码。读者可使用本书所有 C++ 类、函数和程序。
- 勘误表。虽然精心编写，但缺点与错误在所难免。特设立一个勘误表，并根据需要更新。欢迎您提出宝贵意见。

另外，下面的辅助资料可供有资格的教师使用。请联系 Addison-Wesley 销售代表，或给 aw.cse@aw.com 发送电子邮件，了解如何访问他们：

- 教师资源手册
- PowerPoint 幻灯片
- 试题库

目 录

第 1 部分 问题解决技术	
第 1 章 编程原理与软件工程3	
1.1 问题求解与软件工程.....3	
1.1.1 问题求解的含义.....3	
1.1.2 软件的生命周期.....4	
1.1.3 优秀解决方案的含义.....10	
1.2 模块化设计.....11	
1.2.1 抽象与信息隐藏.....11	
1.2.2 面向对象的设计.....13	
1.2.3 自上而下的设计.....14	
1.2.4 一般设计原则.....15	
1.2.5 使用 UML 为面向对象的设计建模.....15	
1.2.6 面向对象方式的优点.....17	
1.3 关键编程问题.....18	
1.3.1 模块化.....18	
1.3.2 可修改.....19	
1.3.3 易用.....21	
1.3.4 防故障编程.....21	
1.3.5 风格.....25	
1.3.6 调试.....30	
1.4 小结.....31	
1.5 提示.....32	
1.6 自我测试题.....32	
1.7 练习题.....33	
1.8 编程问题.....35	
第 2 章 递归：镜子37	
2.1 递归解决方案.....37	
2.1.1 递归值函数：n 的阶乘.....39	
2.1.2 递归 void 函数：逆置字符串.....43	
2.2 计数.....52	
2.2.1 兔子繁殖(Fibonacci 序列).....52	
2.2.2 组织游行队伍.....53	
2.2.3 Spock 的困惑.....56	
2.3 数组查找.....57	
2.3.1 查找数组的最大项.....58	
2.3.2 折半查找.....59	
2.3.3 查找数组中的第 k 个最小项.....62	
2.4 组织数据.....64	
2.5 递归与效率.....69	
2.6 小结.....72	
2.7 提示.....72	
2.8 自我测试题.....73	
2.9 练习题.....73	
2.10 编程问题.....79	
第 3 章 数据抽象：墙80	
3.1 抽象数据类型.....80	
3.2 指定 ADT.....83	
3.2.1 ADT 列表.....84	
3.2.2 ADT 有序表.....88	
3.2.3 设计 ADT.....89	
3.2.4 公理.....92	
3.3 实现 ADT.....94	
3.3.1 C++类.....95	
3.3.2 C++命名空间.....102	
3.3.3 基于数组的 ADT 列表实现.....104	
3.3.4 C++异常.....109	
3.3.5 使用异常的 ADT 列表实现.....110	
3.4 小结.....112	
3.5 提示.....113	
3.6 自我测试题.....113	
3.7 练习题.....114	
3.8 编程问题.....116	

第4章 链表	117	5.3.1 factorial 递归算法的正确性	189
4.1 预备知识	117	5.3.2 Hanoi 塔的成本	190
4.1.1 指针	118	5.4 小结	191
4.1.2 数组的动态分配	123	5.5 提示	192
4.1.3 基于指针的链表	124	5.6 自我测试题	192
4.2 链表编程	125	5.7 练习题	192
4.2.1 显示链表的内容	126	5.8 编程问题	195
4.2.2 从链表中删除指定的节点	127		
4.2.3 在链表的指定位置插入节点	129		
4.2.4 ADT 列表的基于指针的实现	133		
4.2.5 比较基于数组的实现和基于 引用的实现	139		
4.2.6 使用文件存储和恢复链表	140		
4.2.7 将链表传给函数	143		
4.2.8 递归地处理链表	144		
4.2.9 把对象作为链表的数据	148		
4.3 链表的各种变化	148		
4.3.1 循环链表	148		
4.3.2 虚拟头节点	150		
4.3.3 双向链表	150		
4.4 清单应用程序	152		
4.5 C++标准模板库	156		
4.5.1 容器	157		
4.5.2 迭代器	157		
4.5.3 标准模板库类 list	158		
4.6 小结	162		
4.7 提示	164		
4.8 自我测试题	165		
4.9 练习题	167		
4.10 编程问题	169		
第5章 递归问题解决技术	172		
5.1 回溯	172		
5.1.1 八皇后问题	172		
5.1.2 使用 STL 类 vector 解决八皇后 问题	174		
5.2 定义语言	178		
5.2.1 语法知识基础	179		
5.2.2 两种简单语言	180		
5.2.3 代数表达式	182		
5.3 递归和数学归纳法的关系	189		
		第2部分 使用抽象数据类型 解决问题	
		第6章 栈	201
		6.1 抽象数据类型	201
		6.2 ADT 栈的简单应用	206
		6.2.1 检查括号匹配	206
		6.2.2 识别语言中的字符串	208
		6.3 ADT 栈的实现	209
		6.3.1 ADT 栈的基本数组的实现	209
		6.3.2 ADT 栈的基于指针的实现	213
		6.3.3 使用 ADT 列表的实现	217
		6.3.4 各种实现方式的比较	221
		6.3.5 标准模板库类 stack	221
		6.4 应用: 代数表达式	223
		6.4.1 计算后缀表达式	223
		6.4.2 中缀表达式与后缀表达式的 等价转换	224
		6.5 应用: 查找问题	227
		6.5.1 使用栈的非递归解决方案	228
		6.5.2 递归解决方案	234
		6.6 栈和递归的关系	236
		6.7 小结	238
		6.8 提示	238
		6.9 自我测试题	238
		6.10 练习题	239
		6.11 编程问题	242
		第7章 队列	247
		7.1 ADT 队列	247
		7.2 ADT 队列的简单应用	249
		7.2.1 读取字符串	249

11.2	ADT 优先队列: ADT 表的 变体	451	13.3	图的遍历	544
11.2.1	堆	454	13.3.1	深度优先查找	545
11.2.2	ADT 优先队列的堆实现	462	13.3.2	广度优先查找	546
11.2.3	堆排序	464	13.3.3	使用 STL 实现 BFS 类	548
11.3	STL 中的表和优先队列	466	13.4	图的应用	550
11.3.1	STL 关联容器	466	13.4.1	拓扑排序	550
11.3.2	STL 的 priority_queue 类和 堆算法	474	13.4.2	生成树	552
11.3	小结	477	13.4.3	最小生成树	555
11.4	提示	478	13.4.4	最短路径	556
11.5	自我测试题	478	13.4.5	回路	560
11.6	练习题	479	13.4.6	一些复杂问题	563
11.7	编程问题	481	13.5	小结	563
第 12 章	表的高级实现	483	13.6	提示	564
12.1	平衡查找树	483	13.7	自我测试题	564
12.1.1	2-3 树	484	13.8	练习题	565
12.1.2	2-3-4 树	499	13.9	编程问题	567
12.1.3	红-黑树	504	第 14 章	外部方法	569
12.1.4	AVL 树	506	14.1	了解外部存储	569
12.2	散列	510	14.2	排序外部文件的数据	571
12.2.1	散列函数	513	14.3	外部表	577
12.2.2	解决冲突	515	14.3.1	确定外部文件的索引	579
12.2.3	散列的效率	521	14.3.2	外部散列	582
12.2.4	如何确立散列函数	523	14.3.3	B-树	584
12.2.5	表遍历: 散列的低效操作	525	14.3.4	遍历	590
12.2.6	使用 STL 实现 HashMap 类	525	14.3.5	多索引	593
12.3	按多种形式组织数据	528	14.4	小结	594
12.4	小结	531	14.5	提示	594
12.5	提示	532	14.6	自我测试题	595
12.6	自我测试题	532	14.7	练习题	595
12.7	练习题	533	14.8	编程问题	597
12.8	编程问题	535	附录 A	C++基础	599
第 13 章	图	536	附录 B	ASCII 字符代码	653
13.1	术语	536	附录 C	C++头文件和标准函数	655
13.2	将图作为 ADT	538	附录 D	数学归纳法	659
13.2.1	实现图	539	附录 E	标准模板库	663
13.2.2	使用 STL 实现 Graph 类	541	术语表	673	
			自我测试题答案	690	

11.2	ADT 优先队列: ADT 表的 变体	451	13.3	图的遍历	544
11.2.1	堆	454	13.3.1	深度优先查找	545
11.2.2	ADT 优先队列的堆实现	462	13.3.2	广度优先查找	546
11.2.3	堆排序	464	13.3.3	使用 STL 实现 BFS 类	548
11.3	STL 中的表和优先队列	466	13.4	图的应用	550
11.3.1	STL 关联容器	466	13.4.1	拓扑排序	550
11.3.2	STL 的 priority_queue 类和 堆算法	474	13.4.2	生成树	552
11.3	小结	477	13.4.3	最小生成树	555
11.4	提示	478	13.4.4	最短路径	556
11.5	自我测试题	478	13.4.5	回路	560
11.6	练习题	479	13.4.6	一些复杂问题	563
11.7	编程问题	481	13.5	小结	563
第 12 章	表的高级实现	483	13.6	提示	564
12.1	平衡查找树	483	13.7	自我测试题	564
12.1.1	2-3 树	484	13.8	练习题	565
12.1.2	2-3-4 树	499	13.9	编程问题	567
12.1.3	红-黑树	504	第 14 章	外部方法	569
12.1.4	AVL 树	506	14.1	了解外部存储	569
12.2	散列	510	14.2	排序外部文件的数据	571
12.2.1	散列函数	513	14.3	外部表	577
12.2.2	解决冲突	515	14.3.1	确定外部文件的索引	579
12.2.3	散列的效率	521	14.3.2	外部散列	582
12.2.4	如何确立散列函数	523	14.3.3	B-树	584
12.2.5	表遍历: 散列的低效操作	525	14.3.4	遍历	590
12.2.6	使用 STL 实现 HashMap 类	525	14.3.5	多索引	593
12.3	按多种形式组织数据	528	14.4	小结	594
12.4	小结	531	14.5	提示	594
12.5	提示	532	14.6	自我测试题	595
12.6	自我测试题	532	14.7	练习题	595
12.7	练习题	533	14.8	编程问题	597
12.8	编程问题	535	附录 A	C++基础	599
第 13 章	图	536	附录 B	ASCII 字符代码	653
13.1	术语	536	附录 C	C++头文件和标准函数	655
13.2	将图作为 ADT	538	附录 D	数学归纳法	659
13.2.1	实现图	539	附录 E	标准模板库	663
13.2.2	使用 STL 实现 Graph 类	541	术语表		673
			自我测试题答案		690

第 1 部分 问题解决技术

第 1 部分共分 5 章，介绍问题解决技术的全部技能，是本书其余部分的基础。第 1 章首先描述什么是“好”的解决方案，以及如何获取“好”的方案。这些技术强调抽象、模块化和信息隐藏。第 1 部分的其余章节讨论解决方案设计的数据抽象、在实现方式中使用的 C++ 指针和类，以及作为问题解决策略的递归技术。

第 1 章 编程原理与软件工程

本章概述：

本章归纳一些基本编程原理，它们是处理大而复杂的程序的基础。这些原理包括了编程的基本原则，还说明了编写设计优秀、描述清晰的程序具有成本效益。本章还简要介绍了算法和数据抽象，讨论它们与作为本书主题的问题解决和编程技术的相关性。后续各章重点讨论组织和使用数据的技术，在学习这些新技术时，您将看到，所有的解决方案都遵循本章讨论的基本原则。

1.1 问题求解与软件工程

在上次编写程序时，您是从哪一步开始的？大部分编程新手也许仅花一点儿时间读一读问题规范，就开始编写代码。他们的目标很明确：使程序能够执行，并期望得到正确的结果。于是，运行程序、分析错误消息、插入分号、更改逻辑和删除分号等，反复处理，直到程序开始工作。大部分时间都耗费在检查语法错误和程序逻辑上。虽然这样做也能逐步提高编程技术，但能开发出大型软件吗？也许可以，但这是不规范的。

要知道，大型软件开发项目通常要求一个编程团队的参与，一个人的力量是无法完成的。团队工作要求进行总体规划、组织与交流。如果软件开发随意而为，将使团队成员无法正常工作，而且会增加成本。软件工程的出现解决了这些问题。软件工程是计算机科学的一个分支，提供了推动计算机程序开发的技术。

在计算机科学中，第 1 课一般重点讨论编程问题；但是，本书重点讨论外延更广的问题解决，因此，第 1 章将首先概述问题求解过程，以及处理问题的各种方式。

1.1.1 问题求解的含义

“问题求解”(problem solving)指描述问题，以及开发计算机程序来解决问题的整个过程。这个过程包含多个阶段，包括理解待解决的问题，从概念上设计解决方案，以及用计算机程序实现解决方案。

“解决方案”(solution)的准确含义是什么？解决方案通常由“算法”和“数据存储方式”两部分组成。“算法”指在有限时间内解决问题的方法的分步描述。算法经常在数据集合上执行操作。例如，某个算法可能必须向集合添加新数据，从集合中删除数据，或针对数据集合提出若干问题。

这可能会造成错觉：问题解决技术主要表现在算法上，而如何存储数据只起辅助作用。实则不然。您需要做的不仅是简单地存储数据。在构建解决方案时，必须组织数据集合，从而按算法要求的方式轻松地操作数据。实际上，本书将花较大篇幅讲述数据组织方法。

在为一个具体的问题设计解决方案时，可通过几种技术来简化设计任务。本章简要介绍这些技术，后续各章将进一步阐述。