国外经典教材·计算机科学与技术

PEARSON
Addison Wesley

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

PEARSON Education
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

EISBN: 0-321-24725-6

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，我们为该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

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 Stractures: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学。

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。

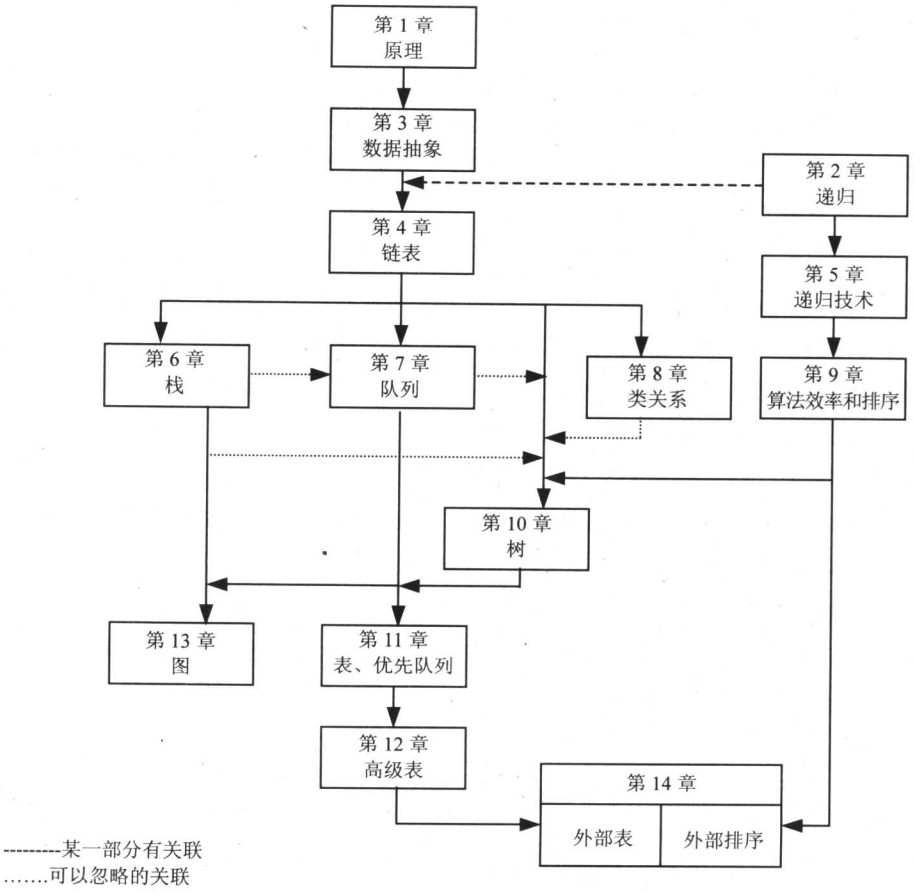
先决条件

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.comn 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 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。