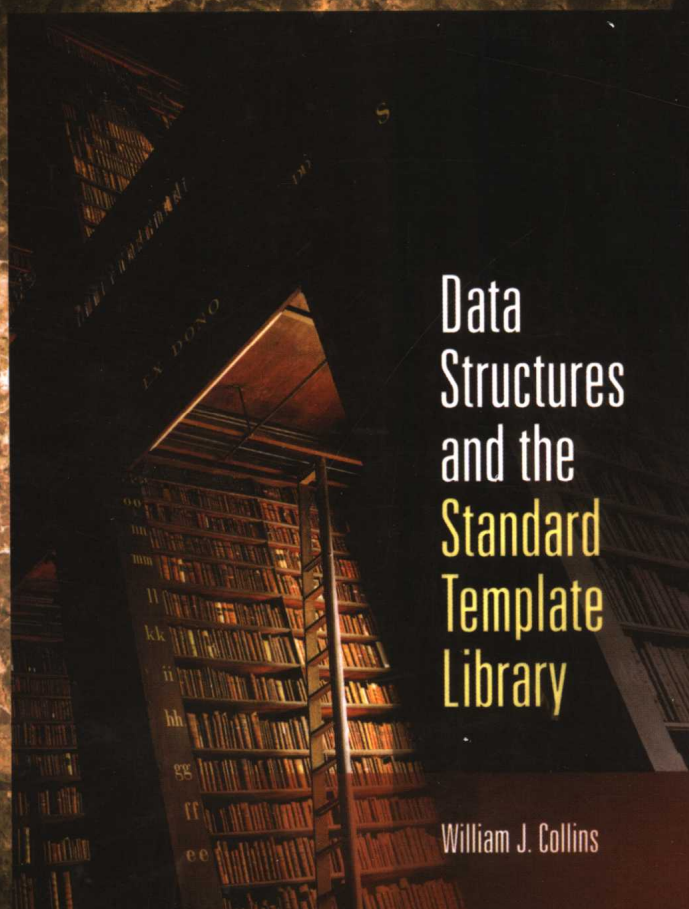


Mc  
Graw  
Hill Education

计 算 机 科 学 从 书

# 数据结构与STL

(美) William J. Collins 著 周翔 译



## Data Structures and the Standard Template Library



机械工业出版社  
China Machine Press

计

算

机

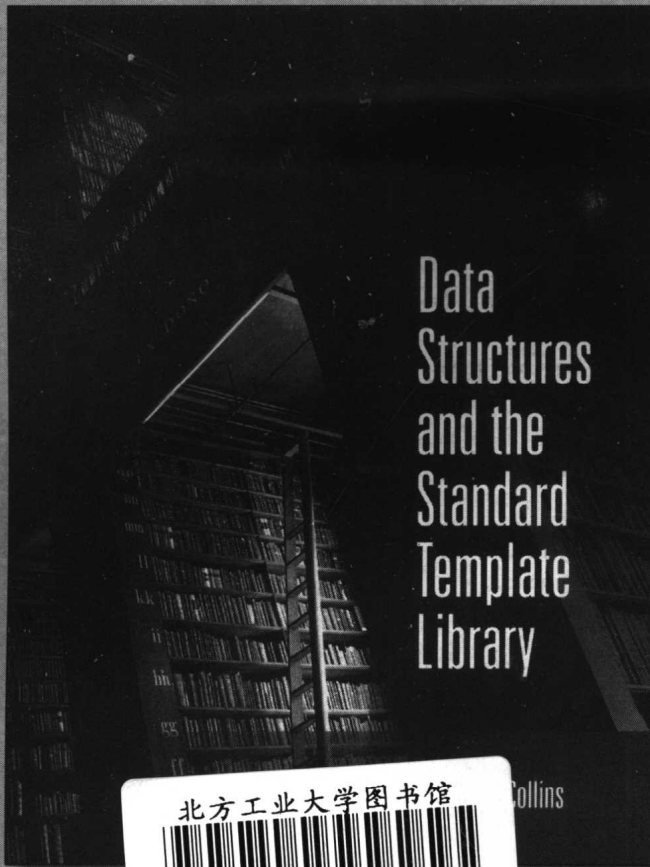
TP311.12  
60

丛

书

# 数据结构与STL

(美) William J. Collins 著 周翔 译



北方工业大学图书馆  
 00547000

## Data Structures and the Standard Template Library



机械工业出版社  
 China Machine Press

数据结构一直是计算机专业课程的核心内容，它是信息的组织方式。对于相同的算法，用不同的数据结构表示其中的抽象数据类型会造成不同的执行效率。

本书从面向对象程序设计的角度，具体使用C++语言，讲述了数据结构及其算法。通过对方法接口、示例和应用的学习，引导学生逐渐理解和掌握如何高效地使用数据结构。

本书与传统数据结构教材相比，除了保留系统、全面的风格之外，还具有重视与实际编程结合、侧重标准模板库的实现描述等特点；并配有丰富的习题及实验，是一本优秀的课堂和自学参考用书。

William J. Collins: Data Structures and the Standard Template Library (ISBN 0-07-236935-5)

Copyright © 2003 by the McGraw-Hill Companies, Inc.

Original English edition published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字翻译版由机械工业出版社和美国麦格劳-希尔教育(亚洲)出版公司合作出版。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。

本书封面贴有McGraw-Hill公司防伪标签，无标签者不得销售。

版权所有，侵权必究。

**本书版权登记号：图字：01-2003-1004**

### **图书在版编目 (CIP) 数据**

数据结构与STL / (美) 柯林斯 (Collins, W.J.) 著; 周翔译. -北京: 机械工业出版社, 2004. 4

(计算机科学丛书)

书名原文: Data Structures and the Standard Template Library

ISBN 7-111-13962-3

I. 数… II. ①柯… ②周… III. ①数据结构 ②C语言-程序设计 IV. ①TP311.12 ②TP312

中国版本图书馆CIP数据核字 (2004) 第014412号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 刘立卿

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2004年4月第1版第1次印刷

787mm × 1092mm 1/16 · 34.25 印张

印数: 0 001 - 4 000 册

定价: 49.00 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换  
本社购书热线: (010) 68326294

# 出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭开了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及收藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：[hzedu@hzbook.com](mailto:hzedu@hzbook.com)

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

# 专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周立柱	周克定	周傲英	孟小峰	岳丽华
范 明	郑国梁	施伯乐	钟玉琢	唐世渭
袁崇义	高传善	梅 宏	程 旭	程时端
谢希仁	裘宗燕	戴 葵		

## 秘 书 组

武卫东      温莉芳      刘 江      杨海玲

# 译者序

数据结构一直是计算机专业课程的核心内容。它是信息的组织方式。对于相同的算法，用不同的数据结构表示其中的抽象数据类型会造成不同的执行效率。因此，对数据结构的研究一直是计算机科技工作者努力的方向。本书从面向对象程序设计语言的角度讲述数据结构及其算法，使用了惠普公司提供的标准模板库（STL）作为基础，代码具有简洁性和高效率的特点，有很重要的实践意义。

在教材内容的安排上，本书仍按照各种不同的数据结构分类进行系统的介绍。在给出了面向对象程序设计及容器类的一般概念之后，通过堆栈、队列、树等常用数据结构剖析了方法接口及其简单的应用。随着C++的广泛应用，容器和模板也越来越受到编程者的重视。以往的教材虽然也详尽地介绍了各种数据结构，但总局限于理论上的探讨；与此不同的是，本书更强调标准模板库的使用，而不仅仅是一般的结构和算法。在介绍每种结构时，都尽可能地融入了它在STL中的表现形式和接口，这样就较好地解决了数据结构和实际编程应用脱节的问题。

本书与传统数据结构教材相比，除了保留系统、全面的风格之外，还增加了以下的新特点：

- 重视数据结构与实际编程的结合。书中的源代码都使用C++语言编写，并进行了验证，更方便有编程需要的读者理解和使用。
- 侧重STL。书中提供了大量的方法接口及实例分析。
- 使用了大量的文本和图表进行辅助说明。
- 每章都包含了章节目标及丰富的习题，对了解学习要求和进一步深入学习提供了很好的帮助。

对于学习和了解数据结构的本、专科学生及研究生而言，本书可作为他们的教材和教学参考书；而对于编程人员、技术服务人员以及程序使用者而言，本书也是一本很好的参考读物。

本书由周翔翻译，在翻译过程中得到了隋立恒、王勇等的帮助。由于本书的内容新，涉及面广，加之译者水平有限，书中难免会存在一些问题，恳请读者提出批评意见。

译者

2003年9月

# 前 言

本书讲述了数据结构及算法。实现语言选用了C++，这适用于已经学习过相关基础课程的学生。这些课程并不一定是面向对象的，但应当覆盖基本语句和数据类型，比如数组和文件处理的基础。

## 标准模板库

本书的显著特点是其依据为标准模板库（STL）——这个库由惠普公司提供。使用这种方法有几个优点。首先，这些代码是被详尽检测过的；其次，读者通过本书有机会学习到以前没有接触过的专业代码，它是相当高效率 and 简洁的；第三，这个库在今后的课程中也是非常有用的。

大多数情况下库不会描述数据结构的实现。这是有好处的，我们可以将注意力放在功能而不是实现细节上。关于这些类的定义，可以参阅惠普研究实验室的Stepanov等人的原始实现（参见Stepanov and Lee, 1994）。这个惠普实现是作者所知的全部实现的基础。

## 考虑过的其他实现

与标准模板库的惠普实现同样重要的是，本书不是只关注于数据结构和算法的基础课程。和惠普实现不同的方法也是值得考虑的。例如，`list`类的实现使用了有头节点的双向链表，它和单链表以及有头尾域的双向链表是不同的。另外本书还比较了不同实现的差异。当然，还有一些数据结构（像图）和算法（像回溯）是不在标准模板库里的。

本书也可以满足数据结构和算法课程的基本需要：让学生练习开发他们自己的数据结构。书中有许多编程项目，它们的数据结构要么是从头创建的，要么是从章中的范例扩展而来的。另外还有一些项目，是开发或扩展那些使用标准模板库的应用程序。

## 标准C++

所有的代码都是基于ANSI/ISO标准C++的，并且在Windows平台（C++ Builder和Visual C++）和Unix平台（G++）上测试过。标准模板库遵循的规范（不是指具体实现）是ANSI/ISO C++的一部分。

## 教学方法的特点

本书有几个特点，这些可以改善教学者的教学环境以及学生的学习环境。每章开头都给出了目标，末尾至少给出一个主要编程任务。每个数据结构都描述得很详细，每个方法都有一个前置条件和后置条件。另外，大部分方法都给出了调用示例以及调用结果。

对细节问题，特别是标准模板库的惠普实现，进行了认真的研究，并在29个实验中得到补充。参阅前言的“实验的组织”部分可以了解更多关于这些实验的信息。每章后都有多种



习题，教师可以使用这些习题的答案。

## 辅助材料

所有的辅助材料都放在网站上：[www.mhhe.com/collins](http://www.mhhe.com/collins)。

网站为学生提供了下列信息的链接：

- 实验的观察以及入门方法。
- 本文开发的所有项目的源代码。
- 项目的Applet，它有一个很强的可视化组件。

另外，教师可以从网站获取下列信息：

- 教师关于实验的选择。
- 每章的PowerPoint幻灯片（大约1500张幻灯片）。
- 每章习题的答案，PowerPoint里展示的习题以及实验的答案。

## 章节纲要

第1章介绍了作为后续章节基础的C++的特点。大部分材料表现的都是面向对象技术：类，继承，构造器，析构器以及运算符的重载。通过实验来回顾类、继承和运算符的重载。

第2章介绍了容器类以及和容器类存储相关的问题。顺序存储和链式存储都需要使用指针。为了说明链式存储，创建了一个单链表类。这个过分简单的Linked类为描述标准模板库的几个关键特色（如模板、迭代器和通用型算法）提供了背景。相关的实验是基于指针、迭代器、运算符重载和通用型算法的。

第3章是软件工程简介，概述了开发软件的四个步骤：分析，设计，实现和维护。使用统一建模语言（UML）作为设计工具来描述继承、复合和聚合。贯穿后续章节的大O表示法可用于脱离环境来评估方法的时间代价。本章还探讨了带驱动器的运行时评估和计时，而且每个主题都对应一个实验。

第4章讲述了递归，它将重点暂时从数据结构转移到算法上。介绍了回溯，把回溯作为解决问题的一般技术。并采用了相同的BackTrack类，用于搜索迷宫、在棋盘上放置八个皇后（使她们不能互相攻击）等应用，并阐述了一个马可以遍历棋盘上的每个空格，而且每个空格只经过一次。其他的递归调用，如汉诺塔游戏和生成置换，更进一步地突出了递归的优雅，尤其是将它和对应的迭代实现相比较时。在后面的章节里也会遇到递归，特别是快速排序法的实现和二叉树的定义中。此外，对每个专业程序员而言，递归是一个必不可缺的工具。

在第5章里，开始使用vector和deque类学习标准模板库。向量是一个灵巧的数组：自动调整大小，并配有方法处理任意位置的插入和删除操作。而且，向量是模板化的，因此将int类型元素插入int类型向量的方法和将string类型元素插入string类型向量的方法是完全相同的。设计过程从vector类中最常用方法的方法接口（前置条件，后置条件和方法头）开始，然后是惠普的大致实现和实验中的进一步的细节。vector类的应用、高精度的算法是公钥加密算法的基础。这个应用在实验和编程项目里进行了更深层次的研究。队列和向量很相似，至少从数据结构的角度来说是这样的。但是实现细节上仍有较大区别，这些细微差异将在实验里探讨。

第6章描述了list数据结构和类，它们的特征是，方法花费线性时间进行随机插入、删除和

检索。这个属性迫使它们使用**链表迭代器**：遍历list对象，并使用常数时间的方法在当前位置上插入、删除和检索对象。本章也介绍了双链式的环型实现，并在一个实验中介绍了其他细节。其应用是一个小的行编辑器，这很适合链表迭代器。在编程项目中对该应用做了进一步的扩展。还有关于迭代器种类，以及向量、双端队列和链表的运行时间比较的实验。

queue和stack类是第7章的主题。这两个类都是**容器配接器**：它们采用了其他一些类的方法接口。对queue和stack类而言，缺省的“其他”类都是deque类。因而得到的stack和queue类的方法定义都是单行的。队列的具体应用——计算洗车处的平均等待时间，属于计算机仿真的范畴。stack类有两个应用：递归的实现，以及中缀到后缀的转换。后一个应用在实验中进行了扩展，并构成了“求一个条件的值”编程项目的基础。

第8章的重点是一般的二叉树，特别是折半查找树。介绍了二叉树的基本特点；这些对理解AVL树、红黑树、堆、霍夫曼树和决策树是很重要的。折半查找树类是红黑树的惠普实现的单色版本。

第9章中考察了AVL树。作为实现重新平衡的机制，介绍了旋转。借助于斐波纳契树，确定了AVL树的高度总是和树中项的数量成对数关系。AVLTree类不是标准模板库的组成部分，但是包含了几个重要的特色，如函数对象；还对该难题提供了后续的实验研究。除了erase方法（编程项目9.1）之外，整个类都得到了实现。AVL树的应用是一个简单的拼写检查器。

红黑树在第10章中进行了研究。仔细研究了红黑树中的插入和删除算法，并且提供了相关的实验。红黑树不在标准模板库中，但是它们是标准模板库中四个关联容器类（set、map、multiset和multimap）大部分实现的基础。在一个集合里，每个项只由一个键组成，是不允许重复键的。多集合允许重复键。在一个映射里，每个项只由一个惟一的键部分和另一部分组成。多映射允许重复键。本章还介绍了一个应用（用来计算文件中每个单词出现的频率），以及有关四个关联容器类的实验。

第11章介绍了priority\_queue类，即另一个容器配接器。缺省使用的是vector类，但幕后还有一个堆，使得以常数平均时间进行插入，而且即使在最坏情况下，也能以对数时间删除最高优先级的元素。可以考虑基于list或基于set的实现。它的应用是在数据压缩领域，特别是霍夫曼编码：给定一个文本文件，生成一个最小的无前缀编码。编程项目的任务是将编码转换回原来的文本文件。实验将公平性融进了优先队列，这样即便是同为最高优先级的项，也总是先处理在优先队列中滞留时间最长的。

排序是第12章的主题。开发了基于比较的排序的最小上界的估算。研究了四种“快速”排序：树排序（多集合），堆排序（随机访问容器），归并排序（列表）和快速排序（随机访问容器）。本章的实验在随机产生的数值上比较了这些排序方法。编程项目是排序姓名和社会保障号码文件。

第13章先开始复习了顺序和折半查找，然后研究了散列。通常，在标准C++或标准模板库的惠普实现中都不支持散列类。本章开发了一个hash\_map类。这个类的方法接口和map类的是相同的，除了插入，删除或查找的平均时间花费是常数而不是对数时间！应用包含了字符表的创建和维护，以及对第9章中拼写检查器应用的修正。还比较了链式散列和开放地址散列；在编程项目中进一步探索了它们的不同。hash\_map类的速度是实验的主题。

第14章介绍了最常用的数据结构——图、树和网络。给出了基本算法的框架：广度优先迭代，深度优先迭代，连通性，寻找最小生成树以及查找两个顶点间的最短路径。开发的惟一

的类是使用邻接表实现的network类。其他类（如undirected\_graph和undirected\_network）可以直接定义为network类的子类。在实验中研究了货郎担问题，并且提出了一个编程项目，要求完成network类的邻接矩阵版本。本章还提出了另一个回溯应用，使用的仍然是第4章所介绍的BackTrack类。

每一章都对应一个相关的网页，其中包括了该章中开发的所有程序以及适于阐释概念的Applet。

## 附录

附录1包含了有助于学生理解书中数学概念的背景信息。累加符号和对数的初步性质是最基本的，而数学归纳原理使得我们能更深地分析二叉树和开放地址散列。

string类是附录2的主题。这个功能强大的类是标准模板库的一个重要组成部分，并使得学生可以绕开乏味的字符数组。

多态性是一个指针引用对象层次中不同对象的能力，在附录3中进行了说明。多态性是面向对象编程的一个基本特性，但被归入了附录，因为它并不是介绍数据结构和算法所必需的论题。

## 实验的组织

本书中共涉及到29个网络实验。学生和教师可以访问的URL是：

[www.mhhe.com/collins](http://www.mhhe.com/collins)

实验不只包含了基本素材，还提供了对文字材料的补充。例如，在研究了vector、deque和list类之后，用一个实验对这三个类进行了一些时间测试。

实验是独立的，因此教师可以很灵活地指定实验。可以将它们指定为：

- 1) 闭卷实验
- 2) 开卷实验
- 3) 不计分作业

除了能明显地提高学习积极性，这些实验还能鼓励学生运用科学的方法。学生们观察到一些现象，如标准模板库的list类的组织。然后阐明并提交一个关于该现象的假设——以及他们自己的代码。测试之后可能需要修正他们的假设，提交根据实验得到的最终结论。

Bill Collins

# 目 录

出版者的话	
专家指导委员会	
译者序	
前言	
第1章 C++中的类	1
1.1 类	1
1.1.1 方法接口	1
1.1.2 对象	2
1.1.3 数据抽象	4
1.1.4 构造器	6
1.1.5 一个Employee类	7
1.1.6 Employee类的定义	11
实验1: Company项目	13
1.1.7 继承	13
1.1.8 受保护的访问	14
1.1.9 HourlyEmployee类	15
实验2: 关于继承的更多的细节	18
1.1.10 运算符的重载	19
1.1.11 友元	20
实验3: 重载运算符“=”和运算符“>>”	21
1.1.12 信息隐藏	21
总结	22
习题	22
编程项目1.1: 一个Sequence类	25
第2章 容器类的存储结构	27
2.1 指针	27
2.1.1 堆和堆栈的对比	29
2.1.2 引用参数	29
2.1.3 指针字段	30
2.1.4 数组和指针	30
实验4: 指针变量赋值与动态变量赋值的对比	31
2.1.5 动态变量的存储空间释放	31
2.2 数组	32
2.3 容器类	33
2.3.1 容器类的存储结构	33
2.3.2 链式结构	34
2.3.3 迭代器	37
2.3.4 Iterator类的设计和实现	39
实验5: 定义其他的迭代器运算符	40
2.3.5 pop_front方法	40
2.3.6 析构器	41
实验6: 重载运算符operator =	42
2.3.7 通用型算法	42
实验7: 更多关于通用型算法的知识	46
2.3.8 数据结构和标准模板库	46
总结	46
习题	47
编程项目2.1: 扩展Linked类	47
第3章 软件工程简介	49
3.1 软件开发生命周期	49
3.2 问题分析	49
3.3 程序设计	51
3.3.1 方法接口和字段	51
3.3.2 依赖关系图	53
3.4 程序实现	54
3.4.1 方法验证	55
实验8: 驱动器	56
3.4.2 正确性实现的可行性	56
3.4.3 方法效率评估	56
3.4.4 大O表示法	57
3.4.5 快速获取大O估算	59
3.4.6 平衡折中	62
3.4.7 运行时间分析	63
3.4.8 随机性	65
实验9: 计时和随机性	66
3.4.9 类型转换	66
3.5 程序维护	67
总结	67
习题	67
编程项目3.1: Linked类的进一步扩充	69
第4章 递归	71

4.1 简介 .....	71	6.1.4 list类的字段和实现 .....	166
4.2 阶乘 .....	71	6.1.5 list节点的存储 .....	171
4.3 十进制到二进制的转换 .....	75	实验15: 更多list类的实现细节 .....	174
实验10: 斐波纳契数 .....	78	实验16: 计时顺序容器 .....	174
4.4 汉诺塔 .....	78	实验17: 迭代器, 第二部分 .....	174
4.5 回溯 .....	86	6.1.6 list类的其他实现 .....	175
4.6 折半查找 .....	97	6.2 链表应用: 一个行编辑器 .....	177
实验11: 迭代折半查找 .....	106	6.2.1 Editor类的设计 .....	180
4.7 生成置换 .....	106	6.2.2 Editor类的实现 .....	182
4.8 间接递归 .....	114	总结 .....	187
4.9 递归的代价 .....	115	习题 .....	187
总结 .....	116	编程项目6.1: 扩展Editor类 .....	189
习题 .....	116	编程项目6.2: list类的另一种设计和实现 .....	195
编程项目4.1: 汉诺塔的迭代版本 .....	121	第7章 队列和堆栈 .....	197
编程项目4.2: 八皇后问题 .....	122	7.1 队列 .....	197
编程项目4.3: 马的遍历问题 .....	123	7.1.1 queue类的方法接口 .....	197
第5章 向量和双端队列 .....	127	7.1.2 使用queue类 .....	200
5.1 标准模板库 .....	127	7.1.3 容器配接器 .....	201
5.2 向量 .....	128	7.1.4 一个接近的设计 .....	202
5.2.1 vector类的方法接口 .....	129	7.2 计算机仿真 .....	204
5.2.2 向量迭代器 .....	134	7.3 队列应用: 洗车仿真 .....	205
5.2.3 向量和其他容器的对比 .....	136	7.3.1 程序设计 .....	207
5.2.4 vector类可能的字段 .....	137	7.3.2 CarWash类的实现 .....	208
5.2.5 vector类的一个实现 .....	137	7.3.3 CarWash方法的分析 .....	212
实验12: vector类的更多的实现细节 .....	142	7.3.4 随机化到达时间 .....	212
5.3 向量的一个应用: 高精度算法 .....	142	实验18: 随机化到达时间 .....	214
5.3.1 very_long_int类的设计 .....	143	7.4 堆栈 .....	214
5.3.2 very_long_int类的一个实现 .....	144	7.4.1 Stack类的方法接口 .....	214
实验13: 扩展very_long_int类 .....	147	7.4.2 使用stack类 .....	215
5.4 双端队列 .....	147	7.4.3 stack类是一个容器配接器 .....	216
实验14: 惠普的deque类实现的更多细节 .....	154	7.5 堆栈应用1: 递归是如何实现的 .....	216
5.5 双端队列的一个应用: 非常长的整数 .....	154	7.6 堆栈应用2: 将中缀转换成后缀 .....	222
总结 .....	154	7.6.1 后缀表示法 .....	224
习题 .....	155	7.6.2 转换矩阵 .....	226
编程项目5.1: 扩展very_long_int类 .....	157	7.6.3 记号 .....	227
编程项目5.2: deque类的另一种实现 .....	157	实验19: 将中缀转化成后缀 .....	228
第6章 表 .....	159	7.6.4 前缀表示法 .....	228
6.1 表 .....	159	总结 .....	230
6.1.1 list类的方法接口 .....	160	习题 .....	231
6.1.2 迭代器接口 .....	163	编程项目7.1: 扩展洗车仿真 .....	232
6.1.3 链表方法和向量或双端队列 方法的差别 .....	165	编程项目7.2: 求一个条件的值 .....	233
		编程项目7.3: 一个迭代的迷宫搜索 .....	237

编程项目7.4: queue类的另一个设计 .....	237	四种情况 .....	331
第8章 二叉树和折半查找树 .....	239	10.2 标准模板库的关联容器 .....	331
8.1 定义和属性 .....	239	10.3 集合应用: 再次讨论拼写检查器 .....	334
8.1.1 二叉树定理 .....	245	10.3.1 multiset类 .....	335
8.1.2 外部路径长度 .....	247	实验24: 更多与set和multiset类相关的知识 .....	336
8.1.3 二叉树的遍历 .....	248	10.3.2 map类 .....	336
8.2 折半查找树 .....	253	10.3.3 multimap类 .....	339
8.2.1 BinSearchTree类 .....	254	实验25: 更多与map和multimap类相关的知识 .....	340
8.2.2 BinSearchTree类的Iterator类 .....	255	总结 .....	340
8.2.3 BinSearchTree类的字段和实现 .....	257	习题 .....	340
8.2.4 递归方法 .....	261	编程项目10.1: 一个简单的词典 .....	343
8.2.5 BinSearchTree迭代器 .....	269	编程项目10.2: 创建一个词汇索引 .....	343
实验20: BinSearchTree的平均高度 .....	270	第11章 优先队列和堆 .....	347
总结 .....	270	11.1 介绍 .....	347
习题 .....	271	11.1.1 priority_queue类 .....	348
编程项目8.1: BinSearchTree类的另一种实现 .....	274	11.1.2 priority_queue类的字段和实现 .....	350
第9章 AVL树 .....	277	11.1.3 堆 .....	351
9.1 平衡的折半查找树 .....	277	实验26: 优先队列中的公平性 .....	359
9.2 旋转 .....	277	11.1.4 priority_queue类的另一种设计及实现 .....	359
9.3 AVL树 .....	281	11.2 优先队列的应用: 霍夫曼编码 .....	360
9.3.1 AVL树的高度 .....	282	11.2.1 huffman类的设计 .....	364
9.3.2 函数对象 .....	284	11.2.2 huffman类的实现 .....	366
实验21: 更多的函数对象的细节 .....	286	总结 .....	371
9.3.3 AVLTree类 .....	286	习题 .....	372
9.3.4 fixAfterInsertion方法 .....	289	编程项目11.1: 解码一个消息 .....	374
9.3.5 insert方法的正确性 .....	297	第12章 排序 .....	377
9.4 AVL树的应用: 一个简单的拼写检查器 .....	299	12.1 介绍 .....	377
总结 .....	302	12.2 排序能有多快 .....	380
习题 .....	302	12.3 快速排序 .....	382
编程项目9.1: AVLTree类的erase方法 .....	305	12.3.1 树排序 .....	382
编程项目9.2: 改进的SpellChecker项目 .....	305	12.3.2 堆排序 .....	383
第10章 红黑树 .....	307	12.3.3 归并排序 .....	385
10.1 红黑树 .....	307	12.3.4 快速排序 .....	390
10.1.1 红黑树的高度 .....	310	12.3.5 分治法算法 .....	395
10.1.2 惠普的rb_tree类 .....	313	实验27: 排序算法的运行时间 .....	396
10.1.3 rb_tree类中的insert方法 .....	315	总结 .....	396
实验22: 使用全部三种情况的红黑树插入 .....	320	习题 .....	397
10.1.4 erase方法 .....	320	编程项目12.1: 排序一个文件 .....	402
实验23: erase的调用, 其中应用了全部			

第13章 查找和散列类 .....	405	14.3 树 .....	448
13.1 分析查找的框架 .....	405	14.4 网络 .....	450
13.2 查找方式复习 .....	405	14.5 图算法 .....	451
13.2.1 顺序查找 .....	405	14.5.1 迭代器 .....	451
13.2.2 折半查找 .....	406	14.5.2 连通性 .....	457
13.2.3 红黑树查找 .....	408	14.5.3 产生最小生成树 .....	458
13.3 hash_map类 .....	408	14.5.4 寻找网络中的最短路径 .....	462
13.3.1 hash_map类中的字段 .....	409	14.6 开发一个网络类 .....	465
13.3.2 散列 .....	409	14.7 network类 .....	465
13.3.3 链式 .....	412	14.7.1 network类中的字段 .....	467
13.3.4 iterator类的字段和实现 .....	419	14.7.2 network类的实现 .....	469
13.3.5 hash_map类的实现 .....	420	14.7.3 与边相关的方法的实现 .....	470
13.3.6 链式散列分析 .....	423	14.7.4 全局方法的实现 .....	472
13.3.7 value_type类 .....	425	14.7.5 get_minimum_spanning_tree方法 .....	475
13.3.8 应用 .....	425	14.7.6 get_shortest_path方法 .....	476
实验28: hash_map计时 .....	427	14.7.7 网络方法的时间花费估算 .....	478
13.4 hash_set类 .....	427	实验29: 货郎担问题 .....	479
13.5 开放地址散列 .....	427	14.7.8 network类的另一种设计和实现 .....	479
13.5.1 erase方法 .....	430	14.8 回溯通过网络 .....	481
13.5.2 主聚类 .....	434	总结 .....	483
13.5.3 双散列 .....	435	习题 .....	484
13.5.4 开放地址散列分析 .....	439	编程项目14.1: 完成邻接矩阵的实现 .....	486
总结 .....	441	编程项目14.2: 回溯通过一个网络 .....	486
习题 .....	442	附录1 数学背景 .....	489
编程项目13.1: 使用链式和双散列构造一个 符号表的运行时间比较 .....	444	附录2 string类 .....	501
第14章 图、树和网络 .....	445	附录3 多态性 .....	511
14.1 无向图 .....	445	参考文献 .....	515
14.2 有向图 .....	447	索引 .....	517

# 第1章 C++中的类

这是一本关于编程的书：重点是对数据结构及算法的理解和使用。C++标准模板库中聚集了大量的数据结构和算法。第2~12章将侧重于解释什么是库以及如何使用库来简化编程。要想使用这些信息，必须先熟悉本章提到的类的情况。有些类是读者已了解的；有些则可能是完全陌生的。所有这些，不论是对库自身还是对如何在编程中使用库而言，都是必须的。

## 目标

---

- 1) 理解类、对象和消息的基本原理。
- 2) 比较程序开发人员和用户对类的看法。
- 3) 能够灵活运用数据抽象的原理，开放—封闭原理，以及子类替换规则。

1<sup>⊖</sup>

## 1.1 类

类将变量和操作这些变量的函数连接了起来。

类是一个用户自定义类型，它由一些变量（称作字段<sup>⊖</sup>）和操作这些字段的函数（称作方法）组成。类将静态组件（字段）和动态组件（方法）封装在一个实体中。这个封装技术改善了程序的模块性：通过隔离类和程序的其他部分，可以更容易理解和修改程序。

假设现在想解决一些问题，需要使用日历上的数据。如果没有现成的可用类，那么就创建一个类Date。这个Date类将由一个或更多的字段组成，用来存放日期，还需要方法操作这些字段。开始时不必过多考虑如何选择表示日期的字段，也不要考虑操作这些字段的方法。因为我们的目的是使用Date，所以首先需要确定Date类的职责。也就是，这个类希望提供给用户什么？假设职责是：

- 1) 给定年、月、日，构造一个日期。
- 2) 从日期中读出年、月、日。
- 3) 判断给定的一个日期是否有效。
- 4) 返回给定日期之后的下一个日期。
- 5) 返回给定日期的前一个日期。
- 6) 返回给定日期是星期几（如星期二）。
- 7) 判断给定日期是否在某些日期之前。
- 8) 以年、月、日的形式输出一个日期，比如：2004, 5, 10。

### 1.1.1 方法接口

方法接口提供给用户有关方法的全部信息。

---

⊖ 此边栏的页码为该书原书页码，与书末索引中的页码相呼应。

⊖ 在其他参考书中，“字段”也可称为数据成员、成员变量、实例变量或属性；“方法”也称为成员函数、服务或操作。



类的职责都被精练在方法接口中：用户调用方法时需要的明确信息。每个方法接口应包含三个部分：一个前置条件，一个后置条件，一个以分号结束的方法头。**前置条件**是在方法运行前假设的程序状态，**后置条件**是在保证前置条件为真时方法运行后的程序状态。

例如，下面给出了isValid方法的方法接口：

```
//后置条件：如果这个Date是合法的就返回真： year应当是1800到2200之间的整数；
//          month必须是1到12之间的整数； day必须是1到给定年给定月的天数之
//          间的整数； 以上三条必须同时成立。否则，将返回假。
bool isValid();
```

这里没有给出前置条件，因为调用这个方法前程序状态无须特别限定。从技术性上而言，前置条件就是**真 (true)**。但是在书写时可以省略。每个方法都应完成一定的功能，因此总是需要明确地给出后置条件。为了阐明后置条件里的“这个Date”短语，需要先定义术语“对象”，它是面向对象编程技术的基础。

### 1.1.2 对象

给定一个类，**对象**（有时称作类的一个**实例**）是一个变量，它拥有这个类的字段并可以调用这个方法。例如，如果定义：

```
Date myDate;
```

那么myDate是一个Date类型的对象。如果在程序后面编写：

```
if (myDate.isValid( ))
    cout << "The date is valid.";
else
    cout << "The date is not valid.";
```

这表示对象myDate调用它的isValid方法。因此，myDate也被看作是一种**调用对象**——调用方法的对象。

在isValid方法的后置条件里，短语“这个Date”指的是调用对象。因此根据调用对象的当前值返回一个**布尔 (bool)** 值。例如，假若对象myDate的字段的价值分别是5, 17, 2003。那么调用

```
myDate.isValid()
```

将返回**真**。但是如果myDate的字段是4, 31, 2003，那么调用

```
myDate.isValid()
```

将返回**假 (false)**。

通常情况下，方法调用句的组成是一个对象、后跟一个点，然后是方法标识符，最后是圆括号括起的一串变元。在面向对象中，**消息**是对象对一个方法的调用。例如，myDate.next()返回调用对象日期的下一个日期。在这个消息里，对象myDate调用了Date类的next方法。术语“消息”是指程序的某一部分和另外部分的通信。例如，消息thisDate.next()也可能是从除了Date类之外的其他类的方法发送的。

精练Date方法的职责后可以得到下面的方法接口：

```
//后置条件：这个Date由month、day和year构成
Date (int month, int day, int year);
```