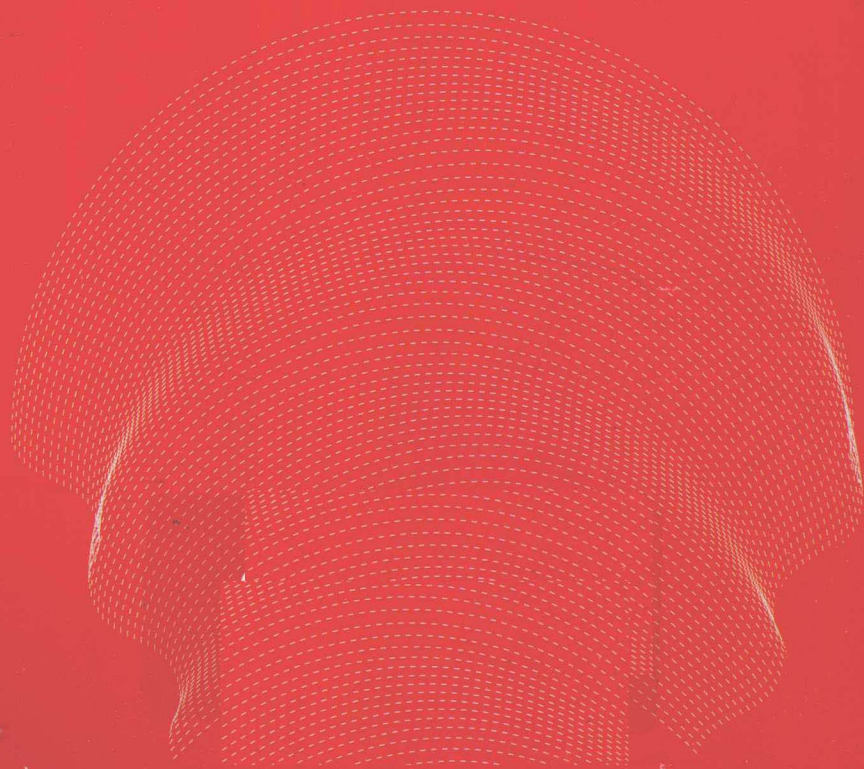


中国高等院校计算机基础教育课程体系规划教材

丛书主编 谭浩强

数据结构上机实验指导C++语言描述

任 燕 段隆振 编著



清华大学出版社



中国高等院校计算机基础教育课程体系规划教材

丛书主编 谭浩强

数据结构上机实验指导C++语言描述

任 燕 段隆振 编著

清华大学出版社
北 京

内 容 简 介

本书是《数据结构C++语言描述》的配套用书,目的是通过上机实验,帮助读者领会每种数据结构实现方式的异同,通过感性认识加深对基本概念的理解。全书共分12章,内容包括实验指导思想和要求、Microsoft Visual C++ 6.0介绍、线性表、栈、队列、顺序串、多维数组、广义表、树、图、排序和查找。对于每种数据结构,包括基本概念的介绍,框架操作、基本操作以及应用的演示实例。本书的附录给出了《数据结构C++语言描述》各章习题的参考答案。

本书可以作为高等院校计算机、信息工程、电子技术类及相关专业学生的实验教材,同时也可供从事计算机软件开发和应用的工程技术人员阅读、参考,是计算机科学与工程领域从业人员不可多得的一本参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构上机实验指导C++语言描述/任燕等编著. —北京:清华大学出版社,2011.1
(中国高等院校计算机基础教育课程体系规划教材)

ISBN 978-7-302-22656-7

I. ①数… II. ①任… III. ①数据结构—高等学校—教学参考资料②C语言—程序设计—高等学校—教学参考资料 IV. ①TP311.12②TP312

中国版本图书馆CIP数据核字(2010)第083701号

责任编辑:张民 顾冰

责任校对:梁毅

责任印制:何芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjjc@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260

印 张:14

字 数:338千字

版 次:2011年1月第1版

印 次:2011年1月第1次印刷

印 数:1~4000

定 价:22.00元

产品编号:034927-01

中国高等院校计算机基础教育课程体系规划教材

编审委员会

主 任：谭浩强

委 员：(按姓氏笔画排序)

王路江	冯博琴	曲建民	刘瑞挺	杨小平
吴文虎	吴功宜	李凤霞	张 森	汪 蕙
高 林	黄心渊	龚沛曾	焦 虹	詹国华

策划编辑：张 民

从20世纪70年代末、80年代初开始,我国的高等院校开始面向各个专业的全体大学生开展计算机教育。特别是面向非计算机专业学生的计算机基础教育,牵涉的专业面广、人数众多,影响深远。高校开展计算机基础教育的状况将直接影响我国各行各业、各个领域计算机应用的发展水平。这是一项意义重大而且大有可为的工作,应该引起各方面的充分重视。

20多年来,全国高等院校计算机基础教育研究会和全国高校从事计算机基础教育的老师始终不渝地在这片未被开垦的土地上辛勤工作,深入探索,努力开拓,积累了丰富的经验,初步形成了一套行之有效的课程体系和教学理念。20年来高等院校计算机基础教育的发展经历了3个阶段:20世纪80年代是初创阶段,带有扫盲的性质,多数学校只开设一门入门课程;20世纪90年代是规范阶段,在全国范围内形成了按3个层次进行教学的课程体系,教学的广度和深度都有所发展;进入21世纪,开始了深化提高的第3阶段,需要在原有基础上再上一个新台阶。

在计算机基础教育的新阶段,要充分认识到计算机基础教育面临的挑战:

(1) 在世界范围内信息技术以空前的速度迅猛发展,新的技术和新的方法层出不穷,要求高等院校计算机基础教育必须跟上信息技术发展的潮流,大力更新教学内容,用信息技术的新成就武装当今的大学生。

(2) 我国国民经济现在处于持续快速稳定发展阶段,需要大力发展信息产业,加快经济与社会信息化的进程,这就迫切需要大批既熟悉本领域业务,又能熟练使用计算机,并能将信息技术应用于本领域的新型专门人才。因此需要大力提高高校计算机基础教育的水平,培养出数以百万计的计算机应用人才。

(3) 从21世纪初开始,信息技术教育在我国中小学中全面开展,计算机教育的起点从大学下移到中小学。水涨船高,这样也为提高大学的计算机教育水平创造了十分有利的条件。

迎接21世纪的挑战,大力提高我国高等学校计算机基础教育的水平,培养出符合信息时代要求的人才,已成为广大计算机教育工作者的神圣使命和光荣职责。全国高等院校计算机基础教育研究会和清华大学出版社于2002年联合成立了“中国高等院校计算机基础教育改革课题研究组”,集中了一批长期在高校计算机基础教育领域从事教学和研究的专家、教授,经过深入调查研究,广泛征求意见,反复讨论修改,提出

了高校计算机基础教育改革思路和课程方案，并于2004年7月公布了《中国高等院校计算机基础教育课程体系2004》（简称CFC 2004）。CFC 2004公布后，在全国高校中引起强烈的反响，国内知名专家和从事计算机基础教育工作的广大教师一致认为CFC 2004提出了一个既体现先进性又切合实际的思路和解决方案，该研究成果具有开创性、针对性、前瞻性和可操作性，对发展我国高等院校的计算机基础教育具有重要的指导作用。根据近年来计算机基础教育的发展，课题调研组对CFC 2004进行了修订和补充，使之更加完善，于2006年和2008年公布了《中国高等院校计算机基础教育课程体系2006》（简称CFC 2006）和《中国高等院校计算机基础教育课程体系2008》（简称CFC 2008），由清华大学出版社出版。

为了实现课题调研组提出的要求，必须有一批与之配套的教材。教材是实现教育思想和教学要求的重要保证，是教学改革中的一项重要的基本建设。如果没有好的教材，提高教学质量只是一句空话。要写好一本教材是不容易的，不仅需要掌握有关的科学技术知识，而且要熟悉自己工作的对象、研究读者的认识规律、善于组织教材内容、具有较好的文字功底，还需要学习一点教育学和心理学的知识等。一本好的计算机基础教材应当具备以下5个要素：

(1) 定位准确。要十分明确本教材是为哪一部分读者写的，要有的放矢，不要不问对象，提笔就写。

(2) 内容先进。要能反映计算机科学技术的新成果、新趋势。

(3) 取舍合理。要做到“该有的有，不该有的没有”，不要包罗万象、贪多求全，不应把教材写成手册。

(4) 体系得当。要针对非计算机专业学生的特点，精心设计教材体系，不仅使教材体现科学性和先进性，还要注意循序渐进、降低台阶、分散难点，使学生易于理解。

(5) 风格鲜明。要用通俗易懂的方法和语言叙述复杂的概念。善于运用形象思维，深入浅出，引人入胜。

为了推动各高校的教学，我们愿意与全国各地区、各学校的专家和老师共同奋斗，编写和出版一批具有中国特色的、符合非计算机专业学生特点的、受广大读者欢迎的优秀教材。为此，我们成立了“中国高等院校计算机基础教育课程体系规划教材”编审委员会，全面指导本套教材的编写工作。

这套教材具有以下几个特点：

(1) 全面体现CFC 2004、CFC 2006和CFC 2008的思路和课程要求。本套教材的作者多数是课题调研组的成员或参加过课题研讨的专家，对计算机基础教育的方向和思路有深切的体会和清醒的认识。因而可以说，本套教材是CFC 2004、CFC 2006和CFC 2008的具体化。

(2) 教材内容体现了信息技术发展的趋势。由于信息技术发展迅速，教材需要不断更新内容，推陈出新。本套教材力求反映信息技术领域中新的发展、新的应用。

(3) 按照非计算机专业学生的特点构建课程内容和教材体系，强调面向应用，注重

培养应用能力,针对多数学生的认知规律,尽量采用通俗易懂的方法说明复杂的概念,使学生易于学习。

(4) 考虑到教学对象不同,本套教材包括了各方面所需要的教材(重点课程和一般课程;必修课和选修课;理论课和实践课),供不同学校、不同专业的学生选用。

(5) 本套教材的作者都具有较高的学术造诣,有丰富的计算机基础教育的经验,在教材中体现了研究会所倡导的思路和风格,因而符合教学实践,便于采用。

本套教材统一规划、分批组织、陆续出版。希望能得到各位专家、老师和读者的指正,我们将根据计算机技术的发展和广大师生的宝贵意见随时修订,使之不断完善。

全国高等院校计算机基础教育研究会荣誉会长
“中国高等院校计算机基础教育课程体系规划教材”编审委员会主任

谭浩强

数据结构是计算机科学的一门主干课程，同时也是一门实用性很强的课程，学好它不仅要在理论上打好基础，更要在实践上多下工夫。

本书是《数据结构C++语言描述》的配套用书，目的在于帮助读者通过实验环节，强化基本概念的理解，验证所学数据结构的特性，并通过具体的应用实例，引导读者学习抽象数据类型的选择和使用，为今后学习面向对象的编程做一些铺垫，从而培养良好的程序设计风格，提高读者的程序设计水平。

本书采用面向对象的设计思想，实验形式多样，具有针对性，符合当今计算机科学与程序设计的发展趋势。每个实验与《数据结构C++语言描述》教程中的章节彼此呼应，环环相扣，目的是希望读者加深对抽象数据结构的理解，能够根据实际问题的特点选择合适的数据结构，设计实用高效的算法，提高动手能力。

书中给出了基本数据结构的概念介绍，框架操作、基本操作的实现以及应用的演示实例。全书共分12章，各章内容安排如下：

第1章 实验指导思想和要求。强调实验的重要性，详细介绍实验步骤，并且给出实验报告的示例。

第2章 Microsoft Visual C++ 6.0 介绍。介绍VC++ IDE的编程环境，建立控制台项目的实验步骤，包括在项目中添加头文件（基类头文件、派生类头文件、测试头文件）和主程序文件的方法，编译、连接、运行和调试程序的过程，以及扩展数据结构操作的方法。

第3章 线性表。测试线性表不同存储结构的实现，包括顺序表、非循环单链表、循环单链表和循环双链表。

第4章 栈。测试顺序栈和链栈存储结构的实现，以及顺序栈在表达式和迷宫问题求解中的应用。

第5章 队列。测试循环顺序队列和非循环链队存储结构的实现，利用循环顺序队列解决运动会比赛项目的安排，以及综合利用栈和队列模拟停车场调度。

第6章 顺序串。测试顺序串存储结构的实现，以及利用顺序串实现简单的文本编辑器功能。

第7章 多维数组。测试稀疏矩阵采用三元组表顺序存储和十字链表存储结构的实现。

第8章 广义表。测试广义表头尾链表存储结构的实现。

第9章 树。测试二叉树、中序穿线二叉树的顺序和链式存储结构的实现，以及二叉树在哈夫曼编码中的应用。

第10章 图。测试图的邻接矩阵和邻接表存储结构的实现，利用拓扑排序合理地安排课表，利用最小生成树原理求解乡村公路的最低造价，利用最短路径算法完成校园路径导游。

第11章 排序。测试插入排序、交换排序、选择排序、归并排序以及分配排序，并把排序的思想应用到实际的学生信息管理中，如按学号排序，按姓名排序以及按某门课程成绩排序等。

第12章 查找。测试静态查找表上的顺序查找、折半查找，以及二叉排序树、平衡二叉排序树、B-树、哈希表上的动态查找；并把查找的思想应用于实际的通讯簿管理，实现按记录位置查找、按姓名查找以及按电话号码查找等。

本书全部实验都在 Visual C++ 6.0 环境下调试通过，并给出了运行结果的实例。读者可以真正体会到数据结构从抽象到具体以及最后回归于应用的完整过程。

本书内容丰富、注重理论联系实际，强调知识的应用，图文并茂，学生易于按图索骥完成实验。因为采用C++模板类实现抽象数据结构，所以可以根据不同的实验要求，进行扩展，具有很高的可重用性。

本书可以作为高等院校计算机、信息工程、电子技术及相关专业学生的实验教材，同时也可供从事计算机软件开发和应用的工程技术人员阅读和参考，是计算机科学与工程领域从业人员不可多得的一本参考书。

在本书的写作过程中，慈爱的母亲虽然读不懂书中的内容，但她能读懂我日夜伏案的艰辛，递给我的一杯杯热茶是母亲对我无尽的爱怜；在寂寞而漫长的写作过程中，是先生那无形的臂膀支撑着我，摒弃心的躁动，用寂静的心像春蚕吐丝一样编织出字字句句；更不能忘记的还有我的爱子，写书的日子母亲不再能尽心地去欣赏他的每一个进步和每一次淘气，……每当想起所有的这些，泪水都会润湿我的双眼，所有感谢的语言在他们的付出前都显得那么的苍白，我只能用此书镌刻下这段美好的回忆。

虽然经过严格的审核、精细的编辑以及上机调试，但由于作者水平有限，缺点和疏漏之处在所难免，恳请读者批评指正，欢迎广大读者和专家提出宝贵建议。

编者

2010年8月

第1章 实验指导思想和要求	1
1.1 实验的重要性	1
1.2 实验步骤	1
1.2.1 需求分析	1
1.2.2 概要设计	2
1.2.3 详细设计	2
1.2.4 编码实现和静态检查	2
1.2.5 上机调试	3
1.2.6 用户使用说明	3
1.2.7 测试结果	3
1.3 实验报告示例	3
第2章 Microsoft Visual C++ 6.0 介绍	10
2.1 Visual C++ 6.0 集成开发环境简介	10
2.2 上机调试步骤	12
2.2.1 建立控制台项目	12
2.2.2 添加头文件	13
2.2.3 添加主程序文件 main.cpp	22
2.2.4 编译、连接和运行程序	24
2.2.5 调试程序	26
2.3 扩展项目操作的步骤	27
2.3.1 扩展基类操作	27
2.3.2 扩展测试头文件	28
2.3.3 扩展主程序文件 main.cpp	29
2.3.4 重新编译、连接和运行程序	31
2.4 参考代码	32
第3章 线性表	36
3.1 顺序表	36

3.1.1	框架操作	37
3.1.2	基本操作	38
3.1.3	综合实验	41
3.2	非循环单链表	48
3.2.1	框架操作	48
3.2.2	基本操作	50
3.2.3	综合实验	54
3.3	循环单链表	58
3.3.1	框架操作	58
3.3.2	基本操作	59
3.3.3	综合实验	61
3.4	循环双链表	62
3.4.1	框架操作	62
3.4.2	基本操作	64
实验习题		67
第4章	栈	68
4.1	顺序栈	68
4.1.1	框架操作	69
4.1.2	基本操作	69
4.1.3	综合实验	72
4.2	链栈	76
4.2.1	框架操作	76
4.2.2	基本操作	77
实验习题		79
第5章	队列	81
5.1	循环顺序队列	81
5.1.1	框架操作	81
5.1.2	基本操作	82
5.1.3	综合实验	85
5.2	非循环链队	87
5.2.1	框架操作	87
5.2.2	基本操作	87
5.2.3	综合实验	91
实验习题		94
第6章	顺序串	95
6.1	框架操作	95

6.2 基本操作	97
6.3 综合实验	102
实验习题	106
第7章 多维数组	107
7.1 稀疏矩阵(采用三元组表顺序存储)	107
7.1.1 框架操作	108
7.1.2 基本操作	109
7.2 稀疏矩阵(采用十字链表存储)	112
7.2.1 框架操作	112
7.2.2 基本操作	114
实验习题	115
第8章 广义表	116
8.1 框架操作	116
8.2 基本操作	117
实验习题	120
第9章 树	121
9.1 二叉树	121
9.1.1 框架操作	122
9.1.2 基本操作	123
9.2 中序穿线二叉树	127
9.2.1 框架操作	128
9.2.2 基本操作	129
9.3 哈夫曼树	130
实验习题	134
第10章 图	135
10.1 图(采用邻接矩阵存储)	135
10.1.1 框架操作	135
10.1.2 基本操作	136
10.1.3 综合实验	138
10.2 图(采用邻接表存储)	141
10.2.1 框架操作	142
10.2.2 基本操作	143
10.2.3 综合实验	145
实验习题	147

第 11 章 排序	148
11.1 插入排序	149
11.2 交换排序	151
11.3 选择排序	152
11.4 归并排序	154
11.5 分配排序	155
11.6 排序的应用	156
11.6.1 框架操作	156
11.6.2 基本操作	158
实验习题	161
第 12 章 查找	162
12.1 顺序查找	162
12.2 折半查找	163
12.3 二叉排序树	164
12.4 平衡二叉排序树	167
12.5 B-树	169
12.6 哈希表	170
12.7 查找的综合应用	171
12.7.1 框架操作	172
12.7.2 基本操作	172
实验习题	175
附录 A 习题参考答案	176
参考文献	209

第1章

实验指导思想和要求

“数据结构”是计算机科学与技术专业、软件工程专业、网络工程专业的一门核心课程。本书通过实验系统地介绍了软件设计中常用的数据结构,相应的存储结构以及操作的实现。通过本书的学习,可以提高对抽象数据结构的感性认识,为后续课程的学习打下良好的基础。

1.1 实验的重要性

“数据结构”是一门实践性较强的课程,为了学好这门课程,必须在掌握理论知识的同时,与实际应用相结合,把课堂上学到的理论知识应用于实际问题的求解。在实际生活中,复杂问题的求解,可能同时涉及多种技术和方法,如选择合适的数据结构(线性表、二叉树、树和图等)、存储结构(顺序存储、链式存储等)、操作的算法实现和递归技术等。不少初学者在选择这些实践技术和方法时,觉得无从下手。其实,解决问题所需的各种技术大多可从教材中找到,只不过其呈现的形式多样化,因此需要仔细体会,反复实践,才能逐渐掌握。

通过对一些典型问题的分析和求解,可逐步提高学生独立思考、分析问题、解决问题和深入研究的能力,突出构造性思维训练,提高组织数据及编写大型程序的能力,为后续课程,特别是软件课程学习打下厚实的实践基础,所以数据结构的上机实践环节十分重要。

1.2 实验步骤

随着计算机性能的提高,问题的复杂度也日趋增加,因此软件开发需要系统的方法。常用的软件开发方法,是将软件开发过程划分为分析、设计、实现和维护四个阶段。虽然数据结构课程中的实验项目远不如实际问题中的复杂,但为了培养软件工作者所应具备的科学工作方法和作风,也应遵循以下8个步骤。

1.2.1 需求分析

实验项目的求解和实现,首先应该进行需求分析,即充分地分析和理解问题,划分出问

题的边界,指出具体的参量或前提条件,明确说明程序设计的任务。强调的是做什么,而不是怎么做。对问题的描述应避免算法,只要对所需完成的任务作出明确的回答。例如:

- ① 输入数据的类型、值的范围以及输入的形式;
- ② 输出数据的类型、值的范围以及输出的形式;
- ③ 会话式输入的结束标志;
- ④ 是否接收非法的输入,对非法输入的应答方式等;
- ⑤ 程序所能达到的功能;
- ⑥ 为调试程序准备好测试数据,包括合法的和非法的输入数据。

1.2.2 概要设计

概要设计(逻辑设计)阶段,对实验项目所涉及的对象定义相应的**抽象数据类型**即数据结构。并以数据结构为中心划分**模块**,即定义数据结构及这些结构上的操作。数据结构的存取通过这些操作加以实现,尽可能做到数据封装。在这个过程中,要综合考虑系统功能,使之结构简单、清晰、合理并且易于调试。

作为概要设计的结果,应写出每个抽象数据类型的定义包括数据结构的描述和每个基本操作的功能说明,主程序的流程以及各程序模块之间的层次调用关系。

1.2.3 详细设计

详细设计的目的是对概要设计进一步求精。用 if、while、赋值语句,以及自然语言写出数据存储结构的定义,为每个操作写出伪代码,对主程序和其他模块写出伪代码,画出函数的调用关系。在求精的过程中,应尽量避免语言细节,不必过早表述辅助数据结构和局部变量。

1.2.4 编码实现和静态检查

编码实现是把详细设计的结果进一步求精为高级语言程序。编写程序特别注意以下问题,可以有效地帮助后继工作的完成。

- 程序的每行不要超过 60 个字符,每个(过程)函数体一般不要超过 40 行,否则应该分割成较小的过程(函数)。
- 必须明确函数功能。对每一段程序的功能加以注释,这会对程序的调试提供很多方便。
- 程序的书写应按照一定的规范,分清每条语句的层次,对齐括号,以便发现语法错误。
- 重要变量应进行注释。
- 要控制 if 语句连续嵌套的深度,分支过多时应考虑使用 switch 语句。
- 控制 for 和 while 等语句连续嵌套的深度。
- 根据情况可以设立若干调试点,即输出若干信息,用于验证与当初的设想是否一致。
- 对于输入输出语句,必须对它们的作用加以说明并给出相应的提示信息;否则,在

调试或运行程序时,无法了解系统需要什么输入,而同时系统的输出又是什么。

在上机之前,应该用笔在纸上写出详细的程序编码,并对这些程序编码进行静态检查。静态检查主要有两种方法,一是用一组测试数据手工执行程序(通常应先分模块检查);二是深入全面地理解程序逻辑思想,在这个过程中再加入一些注释和断言。

1.2.5 上机调试

上机调试程序时,要带一本高级语言教材或手册。熟练运用高级语言的程序调试器,掌握调试工具,设计调试方案和测试数据,手工得出正确结果,以便核实程序是否正确。

调试最好分模块进行,自底向上,先调试底层模块,再调试上层模块,最后把整个程序进行联调。

在调试过程中可以不断借助调试器的各种功能,提高调试效率。调试中遇到的各种异常现象,可用排除法,缩小疑点范围,逐一清查。

调试正确后,认真整理源程序及其注释,形成格式和风格良好的源程序清单和结果。总结经验,包括调试过程中遇到的问题,采用的解决方案,分析算法的时间复杂度并且提出改进设想。

1.2.6 用户使用说明

说明程序的执行方法,详细列出操作步骤。

1.2.7 测试结果

完成实验后,应该设计完整和严格的测试方案,列出测试数据和运行结果。

严格按照实验步骤规范进行,可以避免不良的编程习惯,培养软件工作者不可缺少的科学工作方法和作风。

1.3 实验报告示例

下面通过一个简单的示例,给出以上实验步骤的具体实现。

【实验题目】

实现非循环单链表指定结点的删除和取某个结点的操作。

【实验步骤】

1. 需求分析

为实现非循环单链表上任意位置的删除和取某个结点的操作,我们进行如下需求分析:

① 输入的形式和输入值的范围:删除指定结点时,输入待删除结点的数据域;取非循环单链表的某个结点时,需输入待取结点的序号。

② 输出的形式:删除操作成功后显示非循环单链表的内容,取非循环单链表的某个结点操作后,应显示所取结点的数据域。若遇不成功的操作,应给出操作失败的信息。

③ 程序所能达到的功能：完成非循环单链表的创建(通过插入操作),实现指定结点的删除和取某个结点的操作。

④ 测试数据：依次执行以下步骤,测试程序的正确性。

A. 调用插入操作,创建一个包含 94,33,18,74,93,24,82,32,50 的非循环单链表。

B. 删除操作中依次输入 18,55,测试在以上非循环单链表中删除数据域为 18 和 55 的结点。

C. 取非循环单链表的某个结点操作中依次输入 5,9,测试取非循环单链表相应序号的结点。

2. 概要设计

为了实现上述功能,需要定义如下模块:

① 抽象非循环单链表模块如下:

抽象非循环单链表模块的定义只考虑此类数据结构共同有的特性和操作。

ADT LinkList

{

数据对象: $D = \{a_i \mid a_i \in \text{IntegerSet}, i = 1, 2, \dots, n, n > 0\}$

数据关系: $R = \{ \langle a_i, a_{i+1} \rangle \mid a_i, a_{i+1} \in D \}$

基本操作: Status deleteElem(ElemType e)

操作结果: 删除非循环单链表中数据域为 e 的第一个结点

基本操作: Status getElem(int i, ElemType&e)

操作结果: 取非循环单链表的第 i 个结点

⋮

}

bool deleteElem(ElemType e)

{ 如果非循环单链表中没有数据域为 e 的结点,则无法删除,报错;否则,删除满足条件的第一个结点 }

Status getElem(int i, ElemType& e)

{ 如果非循环单链表中有第 i 个结点,则通过 e 返回第 i 个结点的数据域;否则,报错 }

⋮

② 派生非循环单链表模块

非循环单链表可以存放一组整数、字符或学生的信息(即每个结点记录了一位同学的学号、姓名和籍贯等信息)。随着存放信息的不同,每个实际应用的非循环单链表,除延续抽象非循环单链表的特性和操作外,可能拓展新的特性和操作,所以在抽象非循环单链表模块的基础上,需定义如下派生非循环单链表模块。

ADT MgLinkList: 继承 LinkList

{

⋮

基本操作: void print()

操作结果: 打印非循环单链表

⋮

}