



高等学校计算机规划教材

数据结构课程设计 (C/C++描述)

■ 阮宏一 鲁 静 主编
■ 宋婉娟 张 琪 史毓达 白红武 副主编



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

TP311.12/140C

2011

高等学校计算机规划教材

数据结构课程设计 (C/C++描述)

阮宏一 鲁 静 主编

宋婉娟 张 琪 史毓达 白红武 副主编

北方工业大学图书馆



C00252378

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是《数据结构课程设计》的教材，在作者多年指导学生完成课程设计的经验基础上编写而成。本书算法全部使用 C/C++ 语言描述，所有应用程序源代码都在 Microsoft Visual C++ 6.0 下运行通过。

全书共分 10 章，第 1 章概述课程设计的目的和要求等；第 2 章至第 7 章对应数据结构课程的相关内容，分别讨论了相应数据结构的应用，如线性表、栈、队列、串、数组及广义表、树、图等；第 8 章和第 9 章分别讨论了动态存储管理、查找、排序及文件操作的应用；第 10 章对应用程序主界面的常用设计方法进行了介绍。其中，第 2 章至第 9 章是本书的重点。每章由本章知识要点、应用开发实例以及课程设计题选三部分组成。

本书应用程序源代码可以在华信教育资源网 (www.hxedu.com.cn) 免费注册下载。

本书适合作为计算机及相关专业《数据结构》课程设计教材，也可作为学生自学数据结构设计的辅助教材，也是软件开发者的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目(CIP)数据

数据结构课程设计：C/C++描述/阮宏一，鲁静主编. —北京：电子工业出版社，2011.1.

高等学校计算机规划教材

ISBN 978-7-121-12445-7

I. ①数… II. ①阮…②鲁… III. ①数据结构—课程设计—高等学校—教材②C 语言—程序设计—课程设计—高等学校—教材 IV. ①TP311.12-41②TP312-41

中国版本图书馆 CIP 数据核字(2010)第 234064 号

策划编辑：索蓉霞

责任编辑：索蓉霞

印 刷：北京市李史山胶印厂

装 订：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×1092 1/16 印张：14.75 字数：380 千字

印 次：2011 年 1 月第 1 次印刷

印 数：4000 册 定价：26.50 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

前言

数据结构是计算机程序设计的重要理论技术基础,《数据结构》课程对计算机学科具有承前启后的地位和作用。由于课堂教学和实践的深度和广度有限,要学生很好地理解和掌握数据结构的理论、相关算法及其应用往往比较困难。针对这种情况,我们编写了这本《数据结构课程设计(C/C++描述)》实践教材,其内容安排与《数据结构》课程主教材的相关章节知识点对应,符合数据结构课程教学大纲的要求。

《数据结构》作为一门专业基础课,其教学要求是:学会分析研究计算机所处理的数据结构的特性,以便为应用中涉及的数据选择适当的逻辑结构、存储结构及其相应的算法,并初步掌握算法的时间和空间分析技术。因此,在本书的编排及选题设计中,主要从上述几方面入手,精心挑选出十多个实际应用问题,并通过解决这些问题的应用程序的设计、开发与实现过程,带领读者一步一步地掌握数据结构课程的精华。要求重点掌握线性表、二叉树和树、图、数组等结构在实践中的应用以及相关算法的设计。其中,线性表是基础,也是数据结构课程的核心。树和图的应用是设计中的难点和重点。

本书的编写力求思路清晰、概念准确、算法精湛、格式规范、典型实用。考虑到本课程设计一般在大学二年级开设,所以本书主要采用C语言或引进部分C++的基本概念和语句来实现各种算法。本课程设计的实验环境选择在Microsoft Visual C++6.0之下,所有应用程序源代码都在Microsoft Visual C++6.0下运行通过。

全书共分10章。第1章概述课程设计的目的和要求等;第2章至第7章对应数据结构课程的相关内容,分别讨论了相应数据结构的应用,如线性表、栈、队列、串、数组及广义表、树、图等;第8章和第9章分别讨论了动态存储管理、查找、排序及文件操作的应用;第10章介绍应用程序主界面的常用设计方法。

第2章至第9章是本书的重点。其中,各章的内容由以下三部分组成:

第一部分是本章知识要点,是对本章主要知识点的提示,包括数据结构的定义、常用存储方法及经典算法等。

第二部分是应用设计实例,从设计要求、概要设计、模块设计、详细设计到测试分析、源程序清单等,按照应用设计开发的全过程进行编写,并对源程序进行了详细注释。目的是帮助读者在无形中学会模仿,起到潜移默化的效果。希望书中的应用实例能起到抛砖引玉的作用,以引来读者更多更优良的设计范例。

第三部分是课程设计题选,目的是希望读者在学习了相关章节的内容后能自己练习提高。选题有难有易。每个选题由问题描述、基本要求、测试数据、实现提示和选做内容五部分组成,希望能够帮助读者在实践过程中得到一些启发,各部分具体目标如下:

(1) 问题描述。旨在为读者建立问题提出的背景环境,指明问题“是什么”。

(2) 基本要求。对问题进一步求精,划出问题的边界,指出具体的参数或前提条件,并规定该题的最低限度和要求等。

(3) 测试数据。旨在为检查学生上机作业提供方便,在完成实习题时读者应自行设计完整和严格的测试方案,当数据输入量较大时,提倡以文件形式向程序提供输入数据。

(4) 实现提示。对实现中的难点及其算法思路等问题作出简要提示,个别问题给出了参考算法。

(5) 选做内容。向那些尚有余力的读者提出了更高的要求,同时也希望开拓读者的思路,在完成基本要求之后,尽可能寻求更优的算法,使得程序结构更合理,更容易修改和扩充。

本书由阮宏一、鲁静主编,并负责全书的总体策划与统稿、定稿工作,宋婉娟、张琪、史毓达、白红武任副主编。各章主要编写人员分工如下:第1章、第7章、第9章由阮宏一、史毓达、白红武编写,第2章、第3章、第10章由宋婉娟、张剑编写,第4章、第5章、第8章由鲁静、吴士泓编写,第6章由张琪编写。

本书适合作为计算机及相关专业《数据结构》课程设计教材,也可作为学生自学数据结构设计的辅助教材,也是软件开发者的参考书。

本书应用程序源代码可在华信教育资源网(www.hxedu.com.cn)免费注册下载。

由于时间仓促及作者水平有限,书中难免存在欠妥和疏漏之处,敬请广大读者批评指正。作者 E-mail: hyruan@hue.edu.cn。

编 者

目 录

第 1 章 课程设计概述	1
1.1 课程设计的地位与作用	1
1.2 课程设计的目标和要求	1
1.3 课程设计的步骤	2
1.4 实验报告规范	3
第 2 章 线性表及其应用	5
2.1 本章知识要点	5
2.2 “学生通讯录管理系统”的设计 与实现	6
2.2.1 设计要求	6
2.2.2 概要设计	6
2.2.3 模块设计	7
2.2.4 详细设计	8
2.2.5 测试分析	9
2.2.6 源程序清单	10
2.2.7 用户手册	16
2.3 “航空订票系统”的设计 与实现	17
2.3.1 设计要求	17
2.3.2 概要设计	17
2.3.3 模块设计	18
2.3.4 详细设计	19
2.3.5 测试分析	26
2.3.6 源程序清单	27
2.3.7 用户手册	33
2.4 课程设计题选	33
2.4.1 运动会分数统计系统	33
2.4.2 约瑟夫环问题	34
2.4.3 城市链表	34
2.4.4 线性表的逆置	35
2.4.5 长整数运算	35
第 3 章 栈、队列及其应用	36
3.1 本章知识要点	36
3.1.1 栈的存储结构	36

3.1.2 队列的存储结构	37
3.2 “表达式求值问题”的设计 与实现	38
3.2.1 设计要求	38
3.2.2 概要设计	38
3.2.3 模块设计	39
3.2.4 详细设计	40
3.2.5 测试分析	41
3.2.6 源程序清单	42
3.2.7 用户手册	43
3.3 “银行排队系统”的设计 与实现	44
3.3.1 设计要求	44
3.3.2 概要设计	44
3.3.3 模块设计	45
3.3.4 详细设计	46
3.3.5 测试分析	48
3.3.6 源程序清单	49
3.3.7 用户手册	57
3.4 课程设计题选	58
3.4.1 停车场管理系统	58
3.4.2 车厢调度问题	58
3.4.3 数制转换问题	59
3.4.4 回文判断	59
3.4.5 商品货架管理	59
3.4.6 括号匹配的检验	60
第 4 章 串结构及其应用	61
4.1 本章知识要点	61
4.1.1 串的定义及基本操作	61
4.1.2 串的存储结构	62
4.1.3 串的模式匹配算法	62
4.2 “串基本操作演示系统”的设计 与实现	64
4.2.1 设计要求	64

4.2.2	概要设计	65	5.3.4	测试分析	105
4.2.3	模块设计	65	5.3.5	源程序清单	105
4.2.4	详细设计	66	5.3.6	用户手册	111
4.2.5	测试分析	69	5.4	课程设计题选	111
4.2.6	源程序清单	70	5.4.1	模拟实现多维数组类型	111
4.2.7	用户手册	73	5.4.2	稀疏矩阵的转置	112
4.3	“文学研究助手系统”的设计 与实现	73	5.4.3	识别广义表的“头”或“尾”的 演示	112
4.3.1	设计要求	73	第6章	树结构及其应用	114
4.3.2	概要设计	74	6.1	本章知识要点	114
4.3.3	模块设计	74	6.1.1	树与森林	114
4.3.4	详细设计	75	6.1.2	二叉树	116
4.3.5	测试分析	77	6.1.3	哈夫曼树及其应用	119
4.3.6	源程序清单	78	6.2	“二叉树基本操作演示程序” 的设计与实现	120
4.3.7	用户手册	78	6.2.1	设计要求	120
4.3.8	“文学研究助手系统”实现方法 之二	79	6.2.2	概要设计	120
4.4	课程设计题选	82	6.2.3	模块设计	121
4.4.1	文本格式化问题	82	6.2.4	详细设计	122
4.4.2	简单行编辑程序	83	6.2.5	测试分析	124
第5章	多维数组、矩阵、广义表及其 应用	85	6.2.6	源程序清单	125
5.1	本章知识要点	85	6.2.7	用户手册	127
5.1.1	多维数组	85	6.3	“哈夫曼树”的建立及其应用	128
5.1.2	稀疏矩阵	86	6.3.1	设计要求	128
5.1.3	广义表	87	6.3.2	概要设计	128
5.2	“稀疏矩阵运算器”的设计 与实现	90	6.3.3	模块设计	128
5.2.1	设计要求	90	6.3.4	详细设计	129
5.2.2	概要设计	90	6.3.5	测试分析	131
5.2.3	模块设计	91	6.3.6	源程序清单	132
5.2.4	详细设计	91	6.3.7	用户手册	134
5.2.5	测试分析	94	6.4	课程设计题选	134
5.2.6	源程序清单	95	6.4.1	二叉排序树基本操作的实现	134
5.2.7	用户手册	99	6.4.2	重言式判别问题	134
5.3	“广义表基本操作演示系统” 的设计与实现	99	6.4.3	表达式类型的实现	135
5.3.1	设计要求	99	6.4.4	打印树形结构	136
5.3.2	模块设计	99	第7章	图结构及其应用	137
5.3.3	详细设计	100	7.1	本章知识要点	137
			7.1.1	图的存储结构	137
			7.1.2	图应用的相关算法	138

7.2	“校园导游系统”的设计 与实现	140	8.4	课程设计题选	193
7.2.1	设计要求	140	8.4.1	伙伴存储管理系统演示	193
7.2.2	概要设计	140	8.4.2	图书管理系统	194
7.2.3	模块设计	141	8.4.3	内部排序算法比较	195
7.2.4	详细设计	143	8.4.4	二叉排序树相关操作	195
7.2.5	测试分析	149	8.4.5	哈希表设计	196
7.2.6	源程序清单	151	8.4.6	统计成绩	196
7.2.7	用户手册	159	第9章	文件操作及其应用	197
7.3	课程设计题选	159	9.1	本章知识要点	197
7.3.1	图基本操作的实现	159	9.1.1	文件的基本概念	197
7.3.2	教学计划编排问题	159	9.1.2	C++ I/O 流的概念及流类库	198
7.3.3	最小生成树问题	160	9.1.3	键盘输入与屏幕输出	198
7.3.4	交通咨询系统设计	160	9.1.4	文件的输入/输出	200
第8章	动态存储管理、查找、内排序 及其应用	162	9.1.5	运算符重载	204
8.1	本章知识要点	162	9.2	二叉排序树与文件操作	206
8.1.1	动态存储管理	162	9.2.1	设计要求	206
8.1.2	查找	164	9.2.2	概要设计	206
8.1.3	排序	166	9.2.3	模块设计	208
8.2	“模拟动态存储管理演示系统”的 设计与实现	168	9.2.4	详细设计	209
8.2.1	设计要求	168	9.2.5	测试分析	211
8.2.2	概要设计	168	9.2.6	源程序清单	213
8.2.3	模块设计	169	9.2.7	用户手册	219
8.2.4	详细设计	170	9.3	课程设计题选	220
8.2.5	测试分析	173	9.3.1	外存文件的排序操作	220
8.2.6	源程序清单	174	9.3.2	索引文件的插入、删除和 查找	220
8.2.7	用户手册	182	第10章	应用程序主界面设计	222
8.3	“航班信息查询与检索系统”的 设计与实现	182	10.1	本章知识要点	222
8.3.1	设计要求	182	10.1.1	窗体型界面	222
8.3.2	概要设计	183	10.1.2	字符型界面	222
8.3.3	模块设计	183	10.2	字符型界面的设计与实现	222
8.3.4	详细设计	184	10.2.1	界面背景颜色设置	222
8.3.5	测试分析	188	10.2.2	界面背景大小设置	223
8.3.6	源程序清单	189	10.2.3	字符菜单设计实例一	223
8.3.7	用户手册	193	10.2.4	字符菜单设计实例二	226
			10.3	字符型界面设计总结	227
			参考文献		228

第1章 课程设计概述

根据《数据结构》课程本身的技术特性,设置《数据结构课程设计》实践环节十分重要。它比课堂教学实验复杂一些,涉及的深度与广度更扩展一些,而且更加实用。其主要目的是通过课程设计的综合训练,提高学生分析问题、解决问题及编写程序的能力。本章从课程设计的地位与作用、目标与要求;课程设计的步骤及实验报告规范等方面进行阐述。

1.1 课程设计的地位与作用

《数据结构》是计算机专业一门重要的专业基础课程,是计算机专业的一门核心课程。本课程较系统地介绍了软件设计中常用的数据结构以及相应的存储结构和实现算法,介绍了常用的多种查找和排序技术,并做了性能分析和比较,内容非常丰富。本课程的学习将为后续课程的学习以及软件设计水平的提高打下良好的基础。由于以下原因,使得掌握这门课程具有较大的难度:(1)内容丰富,学习量大,给学习带来困难;(2)贯穿全书的动态链表存储结构和递归技术是学习中的重点也是难点;(3)所涉及的技术多,而在此之前的各门课程中所介绍的专业性知识又不够,因而加大了学习难度;(4)隐含在各部分的技术和方法丰富,也是学习的重点和难点。

由于《数据结构》课程具有上述特点和难点,所以,后续课程《数据结构课程设计》的设置十分必要。为了帮助学生更好地巩固和掌握《数据结构》课程的精髓,理解和掌握算法设计所需的技术,为整个专业学习打好基础,要求学生能够运用所学知识,上机解决一些典型问题,通过分析、设计、编码、调试等各环节的训练,使学生深刻理解、牢固掌握所用到的一些技术。数据结构中稍微复杂一些的算法设计中可能同时要用到多种技术和方法,如算法设计的构思方法、动态链表、算法的编码、递归技术、与特定问题相关的技术等。这都需要在《数据结构课程设计》课程中,锻炼学生在掌握基本算法的基础上,进一步提高分析和解决实际问题的能力。

1.2 课程设计的目标和要求

上机实践是对学生的一种全面综合训练,是与课堂学习、自学和练习相辅相成的必不可少的一个教学环节。较大的课程设计题选比平时的习题复杂得多,也更接近实际。实践着眼于原理与应用的结合,使学生学会如何把书本上学到的知识用于解决实际问题,培养软件设计所需要的动手能力。实践还能使书上的知识变“活”,达到深化理解和灵活掌握教学内容的目的。平时的练习较偏重于如何编写功能单一的“小”算法,而课程设计题选是软件设计的综合训练,包括实际问题分析,总体结构设计,用户界面设计,程序设计基本技能和技巧等。此外,多人合作,以至一整套软件工作规范的训练和科学作风的培养也是开设课程设计的课程的目的。本课程设计的目标就是要达到理论与实际应用相结合,提高学生组织数据及编写大型程序的能力,并培养基本的、良好的程序设计技能以及合作能力。

设计中要求综合运用所学知识,上机解决一些与实际应用结合紧密的、规模较大的问题,通过分析、设计、编码、调试等环节的训练,使学生深刻理解、牢固掌握数据结构和算法设计技术,掌握分析、解决实际问题的能力。

通过课程设计,要求学生在数据结构的逻辑特性和物理表示、数据结构的选择和应用、算法的设计及其实现等方面,加深对《数据结构》课程基本内容的理解。同时,在程序设计方法以及上机操作等基本技能和科学作风方面受到比较系统和严格的训练。

1.3 课程设计的步骤

随着计算机性能的提高,它所面临的软件开发的复杂度也日趋增加,因此软件开发需要系统的方法。一种常用的软件开发方法,是将软件开发过程分为分析、设计、实现和维护 4 个阶段。虽然《数据结构课程设计》中的实践题选的复杂度远不如实际中真正的软件系统,但是如下所述完成实践的 5 个步骤,是一个软件工作者所应具备的科学工作的方法和作风。

1. 问题分析和任务定义

通常,课程设计题选的陈述比较简洁,或者说有模棱两可的含义。因此,在进行设计之前,首先应该充分地分析和理解问题,明确问题要求做什么,限制条件是什么。本步骤强调的是做什么,而不是怎么做。对问题的描述应避免算法和所涉及的数据类型,而是对所需完成的任务作出明确的回答。例如,输入数据的类型、值的范围以及输入的形式;输出数据的类型、值的范围及输出的形式;若是会话式的输入,则结束标志是什么,是否接受非法的输入,对非法输入的回答方式是什么等。这一步还应该为调试程序准备好测试数据,包括合法输入的数据和非法形式输入的数据。

2. 数据类型和系统设计

在这一步骤中,需分逻辑设计和详细设计两步实现。逻辑设计是指:对问题描述中涉及的操作对象,定义相应的数据类型;并按照以数据结构为中心的原则划分模块,定义主程序模块和各抽象数据类型。详细设计,则为定义相应的存储结构并写出各过程和函数的伪码算法。在这个过程中,要综合考虑系统功能,使系统结构清晰、合理、简单和易于调试,抽象数据类型的实现尽可能做到数据封装,基本操作的规格说明尽可能明确具体。作为逻辑设计的结果,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规格说明),各个主要模块的算法,并画出模块之间的调用关系图。详细设计的结果是对数据结构和基本操作的规格说明作出进一步的求精,写出数据存储结构的类型定义,按照算法书写规范用合适的语言写出过程或函数形式的算法框架。在求精的过程中,应尽量避免陷入语言细节,不必过早表述辅助数据结构和局部变量。

3. 编码实现和静态检查

编码是将详细设计的结果进一步求精为程序设计语言程序。如何编写程序才能较快地完成调试是特别要注意的问题。程序的每行一般不要超过 60 个字符。每个过程(或函数)体一般不要超过 60 行,否则应该分割成较小的过程(或函数)。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。对函数功能和重要变量进行注释。一定要按格式书写程序,分清每条语句的层次,对齐括号,这样便于发现语法错误。

在上机之前,应该用笔在纸上写出详细的程序编码,并做认真地静态检查。多数初学者在编好程序后处于以下两种状态之一:一种是对自己的“精心作品”的正确性确信不疑;另一种是认为上机前的任务已经完成,纠查错误是上机的工作。这两种态度是极为有害的。对一般的程序设计者而言,当编写的程序长度超过 60 行时,通常会含有语法错误或逻辑错误。上机动态调试绝不能代替静态检查,否则调试效率将是极低的。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(通常应先检查单个模块);二是通过阅读或给别人讲解自己的程序而深入全面地理解程序逻辑,在这个过程中再加入一些注释。

4. 上机准备和上机调试

上机准备包括以下几个方面:

- (1) 熟悉 C 或 C++ 语言用户手册或程序设计指导书。
- (2) 注意 Turbo C、VC 与标准 C 语言之间的细微差别。
- (3) 熟悉机器的操作系统和语言集成环境的用户手册,尤其是最常用的命令操作,以便顺利进行上机的基本活动。
- (4) 掌握调试工具,考虑调试方案,设计测试数据并手工得出正确结果,学生应该熟练运用高级语言的程序调试器 DEBUG 调试程序。

上机调试程序时要带一本高级语言教材或手册。调试最好分模块进行,自底向上,即先调试低层过程或函数。必要时可以另写一个调用驱动程序。这种表面上麻烦的工作实际上可以大大降低调试所面临的复杂性,提高调试工作效率。

在调试过程中可以不断借助 DEBUG 的各种功能,提高调试效率。调试中遇到的各种异常现象往往是预料不到的,此时不应苦思冥想,而应借助系统提供的调试工具确定错误。调试正确后,认真整理源程序及其注释,印出带有完整注释的且格式良好的源程序清单和结果。

5. 总结和整理实验报告

最后是总结和整理实验报告阶段。除了必要的文档内容,还应该对开发设计的整个过程进行总结,包括应用程序的优点及不足、开发过程中的经验及体会、应用程序的后续开发设想等。值得注意的是,实验报告的各种文档资料,例如,上述中的前三部分应该在程序设计开发的过程中逐渐充实形成,而不是最后补写。

1.4 实验报告规范

实验报告的开头应给出题目、班级、姓名、学号和完成日期等,正文一般包括以下 6 方面的内容。

1. 需求分析

需求分析以无歧义的陈述说明程序设计的任务,重点强调的是程序要做什么?一般应明确规定以下内容:

- (1) 输入的形式和输入值的范围;
- (2) 输出的形式;
- (3) 程序所能达到的功能;
- (4) 测试数据:包括正确的输入及其输出结果和含有错误的输入及其输出结果。

2. 概要设计

概要设计说明本程序中用到的所有抽象数据类型的定义、各子程序（函数和过程）的功能及其调用关系以及各程序模块之间的层次（调用）关系。需要画出函数和过程的调用关系图。

3. 详细设计

详细设计实现概要设计中定义的所有数据类型，包括全局变量的定义等。对每个子程序（函数和过程）需要写出用高级程序设计语言描述的算法或伪码算法（伪码算法达到的详细程度建议为，按照伪码算法可以在计算机键盘直接输入高级程序设计语言描述的程序）；对主程序和主要算法模块应重点介绍并写出详细的高级程序设计语言描述的算法。当子函数或过程较多时，建议对函数或过程进行编号。

4. 测试分析

测试分析一般包括以下内容：

- (1) 调试过程中遇到的问题是如何解决的以及对设计与实现的回顾讨论和分析。
- (2) 算法的时空分析（包括基本操作和其他算法的时间复杂度和空间复杂度的分析）和改进设想。
- (3) 经验和体会等。
- (4) 测试功能展示。列出你的测试结果，包括输入和输出，这里的测试数据应该完整和严格。

5. 源程序清单

源程序清单是带详细注释的源程序。如果提交源程序代码，可以只列出程序文件名的清单。

6. 用户使用手册

用户使用手册告诉用户如何使用你编写的程序，详细列出每一步的操作步骤和使用规则等。

第2章 线性表及其应用

线性表是最简单、最基本、是最常用的一种线性结构。它有两种存储方法：顺序存储和链式存储，它的主要基本操作是插入、删除和检索等。本章主要目的是帮助读者熟练掌握线性表的基本操作在两种存储结构上的实现，其中以各种链表的操作和应用作为重点内容。

2.1 本章知识要点

1. 线性表的顺序存储

线性表的顺序存储指的是在内存中用一组地址连续的存储单元依次存放线性表的数据元素。用这种存储形式存储的线性表称为顺序表(Sequential List)。因为内存中的地址空间是线性的，所以，用物理位置上的相邻实现线性表中数据元素之间的逻辑相邻关系既简单又自然。

在C程序中，一维数组在内存中占用的存储空间是一组连续的存储区域，因此，用一维数组来表示顺序表是合适的。将线性表

$$A = (a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$$

存放于一维数组中，其顺序存储结构如图2-1所示。

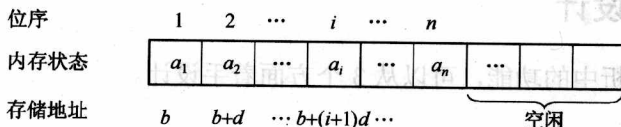


图2-1 线性表的顺序存储结构

2. 线性表的链式存储

线性表的顺序存储，内存的存储密度高，在结点等长时，可以随机地存取结点；但是，对顺序表进行插入、删除时需要通过移动数据元素来实现，影响了运行效率。线性表的链式存储可以弥补上述的不足，它适合插入、删除频繁，存储空间大小不能预先确定的线性表。

链表亦称为线性表 $(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$ 的链式存储结构。如果链表中每个结点只含一个指向后继的指针，则称其为线性链表或单链表。单链表的存储结构定义如下，结点结构如图2-2所示。

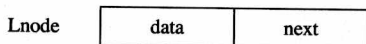


图2-2 单链表结点结构

```
typedef struct LNode
{
    ElemType    data;
    struct LNode *next;
} LNode, *LinkList;
```

2.2 “学生通讯录管理系统”的设计与实现

2.2.1 设计要求

1. 问题描述

纸质的通讯录已经不能满足大家的要求，容易丢失、查找困难等问题是纸质通讯录所不能克服的缺点。“学生通讯录管理系统”是为了帮助老师、同学，或者其他一些需要使用通讯录的人员进行管理和分析的一种应用程序。

2. 需求分析

- (1) 输入数据建立通讯录。
- (2) 查询通讯录中满足要求的信息。
- (3) 插入新的通讯录信息。
- (4) 删除不需要的通讯录信息。
- (5) 查看所有的通讯录信息。

2.2.2 概要设计

为了实现需求分析中的功能，可以从3个方面着手设计。

1. 主界面设计

为了实现学生通讯录管理系统各功能的管理，设计一个含有多个菜单项的主控菜单子程序以链接系统的各项子功能，方便用户使用本系统。本系统主控菜单运行界面如图2-3所示。

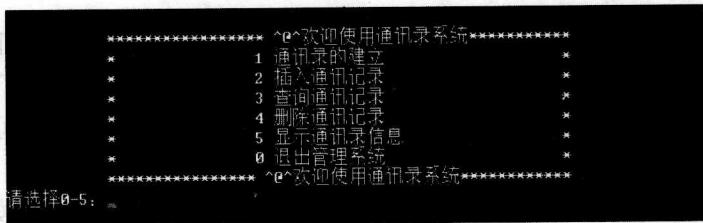


图 2-3 “学生通讯录管理系统”主菜单

2. 存储结构设计

本系统主要采用链表结构类型来表示存储在“学生通讯录管理系统”中的信息。其中，链表结点由4个分量构成：通讯录成员学号、通讯录成员姓名、通讯录成员电话号码、指向该结构体的指针。此外，本系统还设置了一个全局变量 `seat`，表示通讯录中成员的序号。

3. 系统功能设计

本系统设置了5个子功能菜单,5个子功能的设计描述如下。

(1) 建立通讯录系统。可以一次输入多个成员通讯录的信息,建立通讯录。该功能由 `creatIncreLink()` 函数实现。

(2) 插入通讯记录。每次可以插入一个成员通讯录的信息,如果要连续插入多个成员通讯录信息必须多次选择该功能。该功能由 `insertYouXu()` 函数实现。

(3) 查询通讯记录。可以按两种方式查询所需要的通讯录成员记录,一是按学号查询,二是按姓名查询。分别由 `searchNum()` 函数和 `searchName()` 函数实现。

(4) 删除通讯记录。可以对通讯录中不再需要的信息进行删除。有三种删除方式:按序号进行删除,按学号进行删除和按姓名进行删除。分别由 `deleteElem()` 函数、`delNum()` 函数和 `delName()` 函数实现。

(5) 显示通讯录记录。可以查看通讯录中所有的通讯录成员记录。该功能由 `printList()` 函数实现。

2.2.3 模块设计

1. 模块设计

本程序包含两个模块:主程序模块和链表操作模块。其调用关系如图2-4所示。

2. 系统子程序及功能设计

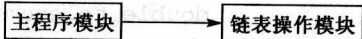


图 2-4 模块调用示意图

本系统共设置10个子程序,各程序的函数名及功能说明如下,其中大部分函数都是链表的基本操作函数。

- (1) `LinkList creatIncreLink()` //链表的创建
- (2) `deleteElem(LinkList L,int i)` //从通讯录中按序号删除第 i 个元素
- (3) `delName(LinkList L,char n[])` //按姓名删除通讯者记录
- (4) `delNum(LinkList L,int n)` //按学号删除通讯者记录
- (5) `void insertYouXu(LinkList L,LinkList Elem)` //插入一条通讯录
- (6) `printList(LinkList L)` //打印指针地址为 L 的通讯录
- (7) `prior(LinkList L,LinkList p)` //查找位于当前地址元素的前一元素的地址
- (8) `searchName(LinkList L,char n[])` //按姓名查找通讯者记录
- (9) `int searchNum(LinkList L,int n)` //按学号查找通讯者记录
- (10) `void main()` //主函数。设定界面的颜色和大小,调用链表操作模块

3. 函数主要调用关系图

本系统10个子程序之间的主要调用关系如图2-5所示。图中数字是各函数的编号。

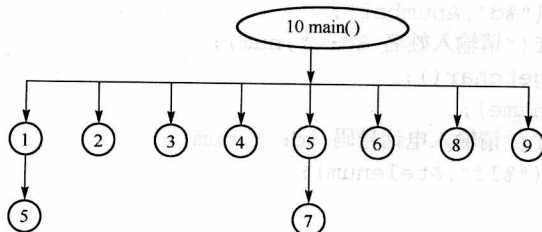


图 2-5 系统函数调用关系图

2.2.4 详细设计

1. 数据类型定义

本系统采用链式结构存储通讯录结点。结点定义如下:

```
typedef struct LNode
{
    int number;
    double telenum;
    char name[20];
    struct LNode *next;
}LNode, *LinkList;
```

2. 系统主要子程序详细设计

(1) 建立链表的函数, 主要用来建立通讯录。

```
LinkList creatIncreLink()
{//创建一个存放通讯录成员的非递减有序表, 返回头结点地址
    LinkList L=(LinkList)malloc(LEN);           //头结点
    L->next=NULL;
    LinkList p;
    int num=1, number;
    double telenum;
    char name[20];
    printf("请输入学生学号、姓名和电话号码, 建立通讯录, 以'-1'为输入结束标志\n");
    printf("请输入学号 %d: ", num);
    scanf("%d", &number);
    printf("请输入姓名 %d: ", num);
    char temp=getchar();
    gets(name);
    printf("请输入电话号码 %d: ", num);
    scanf("%lf", &telenum);
    while (number>=0)
    {
        p=(LinkList)malloc(LEN);           //新结点
        p->number=number;
        p->telenum=telenum;
        strcpy(p->name, name);
        insertYouXu(L, p);                 //有序地插入新结点
        num++;
        printf("请输入学号 %d: ", num);
        scanf("%d", &number);
        printf("请输入姓名 %d: ", num);
        temp=getchar();
        gets(name);
        printf("请输入电话号码 %d: ", num);
        scanf("%lf", &telenum);
    }
    return(L);
}
```

(2) 显示链表中所有结点的信息, 用于查看通讯录所有的记录。

```
void printList(LinkList L)
{ //打印头结点地址为L的通讯录
    printf ("\n          -----\n");
    printf ("          学号      姓名      电话号码\n");
    printf ("          -----\n");
    LinkList p=L;
    int n=1;
    if (L==NULL || L->next==NULL) printf ("该通讯录中没有元素\n");
    //判断通讯录是否为空
    else
        while(p->next !=NULL)
        {
            printf ("          %2d   %-9d", n, p->next->number);
            printf ("          %-5s   %.0f\n", p->next->name, p->next->telenum);
            p=p->next;
            n++;
        }
    printf ("\n          -----\n");
    return ;
}
```

2.2.5 测试分析

系统运行主界面如图2-3所示。各子功能测试运行结果如下。

1. 通讯录的建立

在主菜单下, 用户输入1并回车, 然后按照提示建立通讯录, 分别输入通讯录成员的学号, 姓名, 电话号码, 运行结果如图2-6所示。

2. 插入通讯记录

在主菜单下, 用户输入2并回车, 可以插入一个新的通讯录成员的信息, 依次输入学号, 姓名和电话号码, 运行结果如图2-7所示。

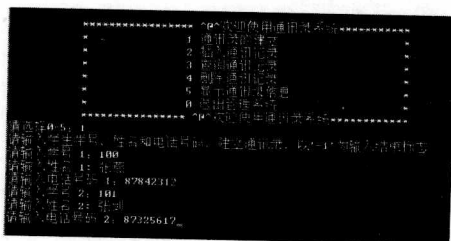


图 2-6 通讯录的建立

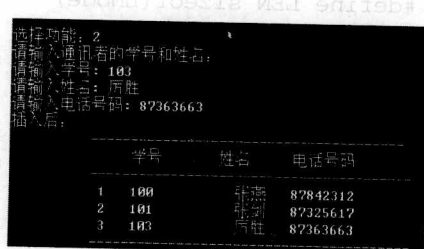


图 2-7 插入通讯记录

3. 查询通讯记录

在主菜单下, 用户输入3并回车, 可以按照两种方式查询通讯录。一种是按学号查询, 另一种方式是按姓名查询。可按照提示操作, 运行结果如图2-8所示。