

高职高专计算机教育规划教材

数据结构

刘喜勋 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机教育规划教材

数据结构

刘喜勋 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

全书共 10 章。第 1 章叙述了数据结构、抽象数据类型的概念,并对算法的描述方法、时间复杂度和空间复杂度通过实例进行了讲述。第 2 章~第 6 章分别讲述了线性表、栈、队列、串和数组线性结构及其应用。第 7 章和第 8 章讲述了树和图非线性结构及其应用。第 9 章和第 10 章讲述了实用的排序和查找方法。对每一种数据结构以实例为切入点,详细叙述了基本概念、逻辑结构、存储结构和常用算法。

本书专为高等职业技术学院计算机类专业学生学习数据结构课程而编写,从数据结构的教學要求出发,以培养学生的基本专业素质及综合应用能力为目标,充分体现内容的先进性、适用性、实用性及可读性。基础理论的阐述由浅入深,通俗易懂。内容的组织和编排以够用为度,突出应用。

图书在版编目(CIP)数据

数据结构/刘喜勋编著. —北京:中国铁道出版社,
2007.1

高职高专计算机教育规划教材
ISBN 978-7-113-07660-3

I. 数… II. 刘… III. 数据结构—高等学校:技术学校—
教材 IV. TP311.12

中国版本图书馆 CIP 数据核字(2007)第 018659 号

书 名: 数据结构

作 者: 刘喜勋

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟 秦绪好

责任编辑: 苏 茜 陈 宏 贾 星

封面设计: 付 巍

封面制作: 白 雪

责任校对: 郑 楠

印 刷: 三河市国英印务有限公司

开 本: 787×1092 1/16 印张: 15.75 字数: 358 千

版 本: 2007 年 4 月第 1 版 2007 年 4 月第 1 次印刷

印 数: 1~5 000 册

书 号: ISBN 978-7-113-07660-3/TP·2200

定 价: 21.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

高职高专计算机教育规划教材

顾 问：冯博琴

主 任：张晓云

副主任：陈建铎 李伟华 王海春 范启岭

王 津 杨俊清 孟繁增 崔永红

委 员：（按姓氏字母先后为序）

白延丽 董少明 韩文智 韩银锋 黄伟敏

李培金 李秀疆 刘省贤 刘喜勋 梅创社

沈久福 王 可 王 坤 吴晓葵 熊永福

杨卫社 杨学全 张 勇 张 宇 钟生海

我国经济建设和发展取得了举世瞩目的成就,随着经济建设发展的需求,教育事业也得到了发展,特别是我国高职高专教育实现了跨越式的发展。依据教育部公布的统计年报,2000年全国高等院校共有1813所,到2005年全国普通高等院校和成人高等院校共有2273所,增长25%;2000年普通高等院校共招本科、高职(专科)学生464.21万人,校均规模达5289人,2005年达504.46万人,校均规模达7666人,增长8%;2005年全国各类高等院校总规模超过2300万人,高等教育入学率达到21%。

党的十六大报告指出,21世纪头20年经济建设和改革的主要任务是:完善社会主义市场经济体制,推动经济结构战略性调整,基本实现工业化,大力推进信息化,加快建设现代化。报告中指出,要坚持以信息化带动工业化,以工业化促进信息化,走出一条科技含量高、经济效益好、资源消耗低、环境污染少、人力资源优势得到充分发挥的新型工业化路子。报告中还指出,要形成以高新技术产业为先导、基础产业和制造业为支撑、服务业全面发展的产业格局。高职高专的专业建设格局也以十六大报告的思路为依据不断调整。例如,陕西省2005年高职高专院校中,根据专业排名前10位的统计,有44所学校开设了“计算机应用技术”课程,排名第一;有26所学校开设了“计算机网络技术”课程,排名第五;有24所学校开设了“计算机信息管理”课程,排名第六。2005年,这3个专业的毕业生人数共计5199人。除了信息产业外,具有信息化知识的复合人才也是信息产业人才需求的另一部分,随着社会信息化程度的日益提高,需要一大批既懂计算机技术,又懂经营管理的信息化人才;随着网络技术的发展,网络管理、网络安全、网页制作等方面的人才需求也将日益增加。

随着高校扩招、用人需求扩大、专业建设不断发展壮大,优质教学资源短缺成为高职高专教育的热点问题。作为信息化基础的计算机技术及应用方向的教学体系与课程建设,在教学思想、教学方法、教学手段不断改革的过程中,积累了大量可推广的经验。一套好的教材是优质教师队伍通过长时间教学实践积累的产物,是教学改革经验与成果的有效推广载体与手段,是教学改革经验与成果推广应用的有效途径,因此,教材建设工作是整个高职高专教育教学工作中的重要组成部分,积极推动教材建设工作是解决优质教学资源短缺、实现优质资源共享的有效方式。中国铁道出版社正是认识到了计算机技术教育发展与信息化建设的关联,因而积极推广教学改革经验与成果,协助高职高专院校实现优质资源共享,并为此推出了“高职高专计算机教育规划教材”丛书。

本丛书本着以服务为宗旨，以就业为导向，面向社会、面向市场、面向职业岗位能力，积极围绕职业岗位人才需求的总目标和职业能力需求，根据不同课程在课程体系中的地位及不同作用，采取不同的教学及教材编写方法。如以知识讲授为主体的围绕问题中心的教学和教材编写方法；以基础能力训练为核心的围绕基础训练任务的教学和教材编写方法；以岗位综合能力训练为核心的以任务为中心的教学和教材编写方法等。

国家兴盛，人才为本；人才培养，教育为本。信息化是我国加快实现工业化和现代化的必然选择，高职高专教育应抓住机遇，乘势而上，培养数以千万计的高技术应用专门人才，为经济建设发展提供保障，以求在贯彻和服务于全面落实科学发展观的过程中能更好地发展。

张 晓 云

数据结构是计算机专业的一门专业基础课。通过对数据结构课程的学习,能使學生掌握现实世界数据的逻辑结构和在计算机中的存储结构等知识,以及算法的实现和算法分析能力。

本教材从数据结构的教學要求出发,以培养學生的基本专业素质及綜合应用能力为目标,充分体现內容的先进性、适用性、实用性和可读性。本教材具有以下特色:

(1) 本教材选用 C 语言作为算法的描述语言,以貼近高职高专的教學实际。又引入抽象数据类型概念,给出了每种数据结构的简单 ADT 定义,为學生学习面向对象程序设计打下基础。

(2) 本教材对每一种数据结构,都通过实例引入,提出待解决的问题,以加快学生的理解,激发学生的学习兴趣。对于基础理论的阐述由浅入深,通俗易懂。內容的组织和编排以“必须、够用”为度,突出应用,略去了一些理论推导和数学证明过程。

(3) 本教材重视实践环节,每章都结合该章內容,设计了习题和上机实验。通过习题和上机,使學生加深对所学知识的理解,积累编程和调试程序经验,提高解决实际问题的能力。

(4) 本教材中每一种数据结构都给出了规范的 C 语言类型定义,算法描述给出了完整的 C 语言程序,并通过算法分析、实例验证和程序注释来帮助学生理解算法。以便使學生学会分析研究计算机加工的数据结构的特征,能为涉及到的数据选择适当的逻辑结构、存储结构及相应的算法。同时,每一个程序都在 Turbo C 2.0 中运行验证,确保和数据结构上机环境相一致。通过上机实验范例可把算法的设计落到实处,使學生得到复杂程序设计的训练。

全书共 10 章。第 1 章叙述了数据结构的概念以及抽象数据类型的概念,并对算法的描述方法、时间复杂度和空间复杂度通过实例给予了讲述。第 2 章~第 6 章分别讲述了线性表、栈、队列、串和数组线性结构及其应用。第 7 章和第 8 章讲述了树和图非线性结构及其应用。第 9 章和第 10 章讲述了实用的排序和查找方法。此外,每章的上机实验范例给出了一个完整的 C 语言程序,以供读者参考。

在编写本书时,编者阅读了大量的国内外同类教材,借鉴了其中的观点方法,博采众长,在此对参考文献中提到和未提到的作者表示感谢。

本书的作者虽是多年从事本课程教學的教师,但由于水平和经验所限,书中难免会有疏漏之处,敬请广大读者批评指正。

编者

2007 年 2 月

目录

CONTENTS

第 1 章 绪论	1
1.1 数据结构的概念及分类	1
1.1.1 数据与数据结构	1
1.1.2 数据结构的分类	2
1.2 抽象数据类型	4
1.2.1 数据类型	4
1.2.2 数据抽象与抽象数据类型	5
1.2.3 用于描述数据结构的语言	6
1.3 算法定义	6
1.4 算法性能分析与度量	7
1.4.1 算法的性能标准	8
1.4.2 算法的后期测试	8
1.4.3 算法的事前估计	9
1.4.4 渐进的时间复杂度	12
1.4.5 渐进的空间复杂度	15
本章小结	16
实验 1 程序的执行时间	17
习 题	18
第 2 章 线性表	20
2.1 线性表实例及概念	20
2.2 线性表的存储方式	21
2.2.1 线性表的顺序存储结构	21
2.2.2 线性表的链式存储结构	23
2.3 线性表的有关操作	26
2.3.1 顺序表的操作实现	26
2.3.2 单链表的操作实现	30
2.3.3 双向循环链表的操作实现	34
2.4 线性表的 ADT 定义	37
2.5 线性表的应用——多项式相加问题	38
2.5.1 存储结构的选取	38
2.5.2 一元多项加法运算的实现	38
本章小结	39
实验 2 线性表的综合运算	42
习 题	45
第 3 章 栈	48
3.1 栈的应用实例及概念	48

3.2 栈的存储方式	49
3.2.1 栈的顺序存储结构	49
3.2.2 栈的链式存储结构	51
3.3 栈的有关操作	51
3.3.1 顺序栈的操作实现	51
3.3.2 链栈的操作实现	53
3.4 栈的 ADT 定义	54
3.5 栈的应用实例——算术表达式的求值	55
3.5.1 表达式的构成	55
3.5.2 运算符的优先关系	55
3.5.3 算法思路	56
本章小结	57
实验 3 栈的操作	58
习 题	60
第 4 章 队列	62
4.1 队列的应用实例及概念	62
4.2 队列的存储方式	63
4.2.1 队列的链式存储结构	63
4.2.2 队列的顺序存储结构	65
4.3 队列的有关操作	68
4.3.1 循环队列的操作实现	68
4.3.2 链队列的操作实现	70
4.4 队列的 ADT 定义	73
4.5 顺序循环队列的应用	74
本章小结	75
实验 4 队列的操作	76
习 题	79
第 5 章 串	82
5.1 串的应用实例及基本概念	82
5.2 串的存储结构	83
5.2.1 串的顺序存储	83
5.2.2 串的链式存储	84
5.3 串运算的实现	85
5.4 串的 ADT 定义	93
本章小结	94
实验 5 串的操作	95
习 题	98
第 6 章 数组	100
6.1 二维数组应用实例及概念	100
6.2 数组的顺序存储和实现	101

6.2.1 数组的顺序存储	101
6.2.2 数组的实现	102
6.3 特殊矩阵的压缩存储	104
6.3.1 三角矩阵	105
6.3.2 稀疏矩阵	106
6.4 数组应用实例	111
本章小结	113
实验 6 数组的定义与实现	114
习 题	116
第 7 章 树	117
7.1 树的实例和基本概念	117
7.1.1 树的实例	117
7.1.2 树的基本概念	118
7.1.3 树的常用术语	119
7.1.4 树的表示方法	120
7.2 二叉树	120
7.2.1 二叉树的定义	120
7.2.2 二叉树的重要性质	121
7.2.3 二叉树的存储结构	122
7.2.4 二叉树二叉链表的一个生成算法	124
7.3 二叉树的遍历	125
7.3.1 二叉树遍历的定义	125
7.3.2 遍历的递归方法	125
7.3.3 二叉树遍历的非递归实现	127
7.4 二叉树其他运算的实现	130
7.5 线索二叉树	131
7.5.1 线索二叉树的基本概念	131
7.5.2 线索二叉树的逻辑表示图	133
7.5.3 中序线索化算法	133
7.5.4 在中序线索树上检索某结点的前驱或后继	134
7.5.5 在中序线索树上遍历二叉树	135
7.6 树与森林	135
7.6.1 树的存储结构	135
7.6.2 树、森林和二叉树的转换	137
7.6.3 一般树或森林的遍历	139
7.7 哈夫曼树及其应用	140
7.7.1 哈夫曼树的基本概念	140
7.7.2 哈夫曼树的构造及其算法	141
7.7.3 哈夫曼树的应用	143
7.8 二叉树的 ADT 定义	145

本章小结.....	146
实验 7 二叉树的建立与遍历.....	148
习 题.....	150
第 8 章 图.....	153
8.1 图的实例及概念.....	153
8.1.1 实例.....	154
8.1.2 图的定义和基本概念.....	154
8.2 图的存储结构及实现.....	157
8.2.1 邻接矩阵.....	157
8.2.2 邻接链表.....	164
8.2.3 图的 ADT 定义.....	172
8.3 遍历图.....	172
8.3.1 深度优先遍历.....	173
8.3.2 广度优先遍历.....	174
8.4 最小生成树.....	176
8.4.1 最小生成树的基本概念.....	176
8.4.2 普里姆算法.....	177
8.4.3 克鲁斯卡尔算法.....	180
8.5 最短路径.....	181
8.5.1 从某个源点到其他各顶点的最短路径.....	182
8.5.2 求每一对顶点之间的最短路径.....	185
本章小结.....	187
实验 8 图的基本算法实现.....	190
习 题.....	193
第 9 章 排序.....	197
9.1 排序的基本概念.....	197
9.2 插入排序.....	198
9.2.1 直接插入排序.....	198
9.2.2 希尔排序.....	200
9.3 交换排序.....	201
9.3.1 冒泡排序.....	201
9.3.2 快速排序.....	203
9.4 选择排序.....	205
9.4.1 直接选择排序.....	205
9.4.2 堆排序.....	205
本章小结.....	210
实验 9 各种排序方法实现.....	211
习 题.....	212
第 10 章 查找.....	214
10.1 静态查找表.....	214

10.1.1 无序顺序表的查找	214
10.1.2 有序顺序表的查找	216
10.1.3 索引顺序表的查找	218
10.2 动态查找表	220
10.2.1 二叉排序树	221
10.2.2 平衡二叉树	225
10.3 哈希表及其查找	226
10.3.1 哈希表与哈希函数	226
10.3.2 构造哈希函数的常用方法	227
10.3.3 解决冲突的主要方法	228
本章小结	230
实验 10 查找方法实现	230
习 题	233
参考文献	235

第 1 章 // 绪论

建议学时: 6 学时

总体要求

- 理解什么是数据、数据对象、数据元素、数据结构
- 理解数据的逻辑结构与物理结构及逻辑结构与物理结构间的关系
- 理解什么是数据类型、抽象数据类型
- 理解算法的定义、特性、时间代价和空间代价
- 熟悉用 C 语言描述算法的方法, 能够使用 C 语言编写程序

核心技能点

- 具有抽象数据的能力
- 具有 C 语言编程的能力
- 具有算法的时间代价和空间代价静态分析能力

扩展技能点

- 抽象数据类型的应用能力
- 算法的时间代价和空间代价方法的实际应用能力

相关知识点

- C 语言的基本语句
- C 语言函数的编写格式及功能
- C 语言标识符的命名规则
- C 语言类型定义
- 数学极限的知识

学习重点

- 掌握数据的逻辑结构与物理结构和逻辑结构与物理结构间的关系
- 熟练掌握算法的定义、特性、时间代价和空间代价分析
- 熟悉用 C 语言描述算法的方法

1.1 数据结构的概念及分类

1.1.1 数据与数据结构

人们在日常生活中会遇到各种信息, 如用语言交流思想, 银行与商店的商业交易, 战争中用于传递命令的旗语等。这些信息必须转换成数据才能在计算机中进行处理。因此, 数据的定义是信息的载体, 是描述客观事物的数、字符以及所有能输入到计算机中并被计算机程序识别和处理的符号的集合。数据大致可分为两类, 一类是数值性数据, 包括整数、浮点数、复数、双精度数等, 主要用于工程和科学计算, 以及商业事务处理; 另一类是非数值数据, 主要包括字符和字符串, 以及文字、图形、图像、语音等数据。

从传统的观点来看,在解决实际问题时,总把数据按其性质归类到一些称之为数据对象(Data Object)的集合中。数据对象中的所有数据成员即为数据元素,都具有相同的性质,它们是数据的子集。例如,整数数据对象可以是集合 $N=\{0,\pm 1,\pm 2,\pm 3,\dots\}$ 。英文字母数据对象可以是集合 $LETTER=\{A,B,\dots,Z\}$ 。

学校中的学生是更复杂的数据对象,它的每一个数据元素就是一个学生记录,主要包括学生的学号、姓名、性别、籍贯、出生年月、成绩等数据成员或数据项,以表明学生在某一方面的属性。这些数据项可以分为两种:一种叫做初等项,如学生的性别、籍贯等,这些数据项是数据处理时不能再分割的最小单位;另一种叫做组合项,如学生的成绩,它可以再划分为物理、化学等课程更小的项。通常,在解决实际问题时是把每个学生记录当作一个基本单位进行访问和处理的。

在学生选课系统中,可能以一个班级的学生记录作为学生数据对象,也可能以一个年级的学生记录作为学生数据对象。所以,如何选择数据对象将依据要求而定。

一个数据对象中所有数据成员之间一定存在某种关系。例如,招生考试时把所有考生按考试成绩从高到低排序。这样,所有考生记录都处在一个有序的序列中。例如,在 n 个网站之间建立通信网络,要求以最小的代价将 n 个网站连通,如图 1-1 (a) 所示。这样,在所有网站之间形成一种树形关系;如果要求网络中任一网站出现故障,整个网络仍然保持畅通,则在所有网站之间形成一种网状关系,如图 1-1 (b) 所示。若综合考虑数据对象及所有数据成员之间的关系,就可得到数据结构的定义:

数据结构由某一数据对象及该对象中所有数据成员之间的关系组成。记为:

$$\text{Data-Structure}=\{D,R\}$$

其中, D 是某一数据对象, R 是该对象中所有数据成员之间关系的有限集合。

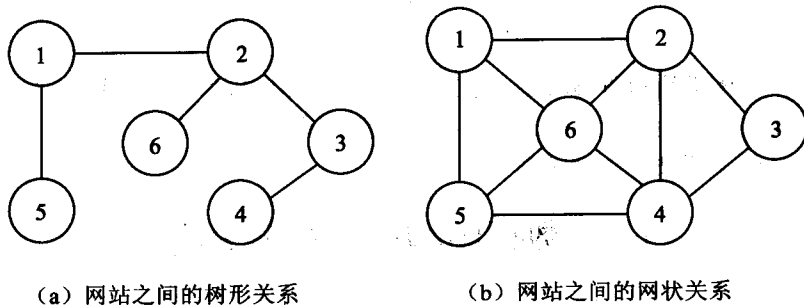


图 1-1 n 个网站之间的连通关系

1.1.2 数据结构的分类

依据数据成员之间关系的不同,数据结构分为两大类:线性结构和非线性结构。线性结构也称为线性表,在这种结构中所有数据成员(也称为数据元素)都按某种次序排列在一个序列中,如图 1-2 所示。线性结构中除第一个元素外,其他每一个元素都有一个且仅有一个直接前驱,第一个数据元素没有直接前驱;除最后一个元素外,其他每一个元素都有一个且仅有一个直接后继。

学 籍 表

序号	学号	姓名	性别	英语	数学	物理
01	20030301	李明	男	86	91	80
02	20030302	马琳	男	76	83	85
⋮	⋮	⋮	⋮	⋮	⋮	⋮
50	20030350	刘薇薇	女	88	93	90

图 1-2 线性结构中各数据成员之间的线性关系

根据线性结构中数据元素的存取方法的不同,又可分为直接存取结构、顺序存取结构和词典(广义索引)结构。对于直接存取结构,可以直接存取某一指定项而不需先访问其前驱。像数组、记录和文件都属于这一类。可以按给定下标直接存取数组中某一数组元素;可以按记录号直接检索记录集合或文件中的某一记录。对于顺序存取结构,只能从序列中第一个数据元素起,按顺序逐个访问,直到指定的元素。例如,在第2章中讨论的线性表就是这种情况。另外,一些限制存取位置在表的一端或两端的表(如栈、队列和优先级队列等)也是这种情况。词典与数组有类似之处,但数组是通过整数下标进行存取,而词典是通过关键码(key)进行存取的。设定数据记录中某一数据项或某一组合数据项为关键码,通过关键码来识别记录。例如,对于学生记录,可设定学生的学号为关键码,用它来识别是哪一位学生的记录。索引表和散列表都属于这一类。

在非线性结构中,各个数据成员不再保持在一个线性序列中,每个数据成员可能与零个或多个其他数据成员发生联系。根据关系的不同,可分为层次结构和群结构。层次结构是按层次划分数据元素的集合,指定层次上的元素,可以有零个或多个处于下一个层次上的、直接的所属下层元素。在第7章将重点讨论树形结构,它是典型的层次结构。树中的元素叫做结点。树可以为空,也可以不为空。若树不为空,它有一个叫做根的结点,其他结点都是从它派生出来的。除根以外,每一个结点都有一个处于该结点直接上层的结点。树的结构如图1-3所示。

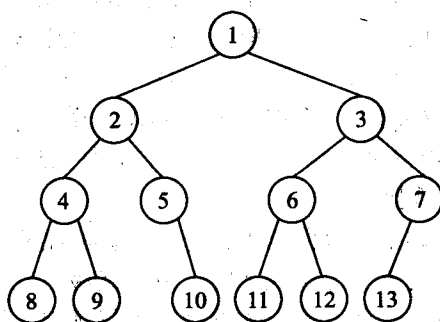


图 1-3 树形结构

群结构中所有元素之间无顺序关系。集合就是一种群结构,在本教材中不讨论。集合中没有重复的元素。另一种群结构就是图结构,如图1-4(a)所示。它是由图的顶点集合和连接顶点的边集合组成的。还有一种图的特殊形式,即网络结构。它给每条边赋予一个权值,这个权值指明了在遍历图时经过此边时的耗费。例如,在图1-4(b)中,顶点代表城市,赋予边的权值表示两个城市之间的距离。

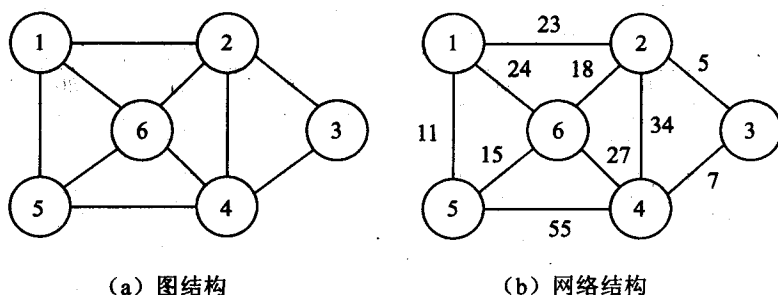


图 1-4 群聚集

此外, 数据结构依据视点的不同, 可分为数据的逻辑结构和物理结构。数据的逻辑结构简称数据结构, 是指从解决问题的需要出发, 为实现必要的功能所建立的数据结构。它属于用户的视图, 是面向问题的。如在招生系统中建立的按考分排列的考生记录的有序表格。数据的物理结构是指数据应该如何在计算机中存放, 是数据逻辑结构的物理存储方式, 属于具体实现的视图, 是面向计算机的。数据的逻辑结构根据问题所要实现的功能建立, 数据的物理结构根据问题所要求的响应速度、处理时间、修改时间、存储空间和单位时间的处理量等建立, 是逻辑数据的存储映像。

通常讨论数据结构, 不但要讨论各种在解决问题时可能遇到的典型的逻辑结构, 还要讨论这些逻辑结构的存储映像, 此外, 还要讨论这种数据结构的相关操作及其实现。

1.2 抽象数据类型

1.2.1 数据类型

在程序设计语言中有各种数据类型。以 C 语言为例, 有 5 种基本的数据类型: 字符型、整型、浮点型、双精度型和无值, 分别用保留字 `char`、`int`、`float`、`double` 和 `void` 标识。这些数据类型不但规定了使用该类型时的取值范围, 而且还规定了该类型可以用的一组操作。例如, 与整型相关的操作有 `+`、`-`、`*`、`/`, 与浮点型相关的操作也有 `+`、`-`、`*`、`/`。然而整型的 `/` 操作与浮点型的 `/` 操作虽然形式一样, 却是两个不同的操作。如整型运算 $3/2$ 的运算结果为 1, 浮点型运算 $3/2$ 的运算结果为 1.5。因为操作 `/` 可用于多个数据类型, 故称它是一种重载操作。

事实上, 数据类型是对数据的取值范围、每一数据的结构以及允许执行操作的一种描述。例如, 在高级程序设计语言中已实现了的, 或非高级语言直接支持的数据结构即为数据类型。在程序设计语言中, 一个变量的数据类型不仅规定了这个变量的取值范围, 而且定义了这个变量可用的操作。例如, 一个变量 `n` 定义为整型, 则它可能的取值范围是 $-32\,767 \sim 32\,767$, 可用的操作有双目运算符 `+`、`-`、`*`、`/`、`%`, 单目运算符 `+`、`-`, 关系运算符 `>`、`<`、`>=`、`<=`、`=`、`!=` 及赋值运算符 `=` 等。在 C 语言中, 不但规定了一些基本的数据类型, 还提供了—些构造复杂类型 (如数组型、结构体、共用体、枚举型和文件型等) 的规则, 程序员可以利用这些规则, 自行定义解决应用问题所必需的数据类型, 它是确切地描述数据对象并正确地进行相关计算的有效工具。

1.2.2 数据抽象与抽象数据类型

在软件设计时,常提到“抽象”和“信息隐蔽”的概念。所谓抽象,就是抽取反映问题本质的东西,忽略非本质的细节。以数据抽象为例,在计算机中使用二进制定点数和浮点数实现数据的存储和运算,而在汇编语言中则给出了各种数据的自然表示,如 15.5、1.3E10、10 等,它们是二进制数据的抽象,用户在编程时可以直接使用它们,不必考虑实现的细节。到了高级语言,给出了更高一级的数据抽象,出现了整型、实型、字符型、双精度型等。待到抽象数据类型 (Abstract Data Type, ADT) 出现,可以进一步定义出更高级的数据抽象,如各种表、队列和图,甚至窗口、管理等。这种数据抽象的层次为设计者提供了有力的手段,使得设计者可以从抽象的概念出发,从整体上进行考虑,然后逐步展开,最后得到所需的结果。

抽象数据类型通常是指由用户定义,用以表示应用问题的数据模型。例如,线性表数据类型的 ADT 定义由数据元素、结构及操作 3 部分组成。

- 元素: a_i 同属于一个数据对象, $i=0, 1, \dots, n-1$ ($n \geq 0$)。
- 结构: 对所有的数据元素 a_i ($i=0, 1, \dots, n-1$) 存在次序关系 $\langle a_i, a_{i+1} \rangle$, a_1 无前驱, a_n 无后继。
- 操作: 对线性表可执行的基本操作如下。

- (1) ListInitiate(L): 初始化操作。可设定一个空的线性表 L。
- (2) ListLength(L): 求长度函数。函数值为给定线性表 L 中数据元素的个数。
- (3) Listempty(L): 判断空表函数。若 L 为空表,则返回 1,否则返回 0。
- (4) ListClear(L): 表清空操作。操作的结果是使 L 成为空表。
- (5) ListGet(L,i): 取元素函数。若 $0 \leq i \leq \text{ListLength}(L)-1$,则函数值为给定线性表 L 中第 i 个数据元素,否则为空元素 NULL。称 i 为该数据元素在线性表中的位序。
- (6) ListLocate(L,x): 定位函数。若线性表 L 中存在其值与 x 相等的数据元素,则函数值为该数据元素在线性表中的位序,否则为 -1。若线性表中与 x 相等的数据元素不止一个,则函数值为这些元素在线性表中的位序的最小值。
- (7) ListPrior(L,elem): 求前驱函数。已知 elem 为线性表 L 中的一个数据元素,若它的位序大于 0,则函数值为 elem 的前驱,否则为空元素。
- (8) ListNext(L,elem): 求后继函数。已知 elem 为线性表 L 中的一个数据元素,若它的位序小于 ListLength(L)-1,则函数值为 elem 的后继,否则为空元素。
- (9) ListInsert(L,i,b): 插入操作 (前插)。在线性表 L 的第 i 号元素之前插入一个新元素 b。此操作仅在 $0 \leq i \leq \text{ListLength}(L)$ 时才可行。
- (10) ListDelete(L,i): 删除操作。若 $0 \leq i \leq \text{ListLength}(L)-1$,则删除线性表 L 中的第 i 个元素,否则此操作无意义。

对于一个其数据成员完全相同的数据类型,如果给它赋予不同的语义,即定义具有不同功能的一组数据,则可形成不同的抽象数据类型。例如,队列和优先级队列,它们可能都是相同的顺序表结构,但其语义不同,队列是先进先出,优先级队列是优先级高的先出,具有各不相同的服务,因此,是不同的抽象数据类型。

抽象数据类型的特征是使用与实现分离,实行封装和信息隐蔽。就是说,在抽象数据类型设计时,把类型的声明与其实现分离开来。