



高等学校计算机科学与技术教材

数据结构实例 教程

□ 杨晓光 编著
□ 李兰友 主审



- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言精炼，实例丰富
- 可操作性强，实用性突出

清华大学出版社

● 北京交通大学出版社

高等学校计算机科学与技术教材

数据结构实例教程

杨晓光 编著

李兰友 主审

清华大学出版社

北京交通大学出版社

· 北京 ·

内 容 简 介

本书系统介绍了数据结构相关理论和基本算法。主要内容包括线性表、栈、队列、串、数组、广义表、树、二叉树、图、查找和排序。

本书从实用角度出发,安排了大量贴近生活的示例,特别是每章都给出一个解决身边问题的综合实例。为了帮助读者更好地理解数据结构,全书还提供了121道例题,430道习题,7个综合实例,13个实验,1个综合测试。

本书可作为大专院校计算机专业和电子信息专业的本、专科学生的教材和参考书,亦适合于工程技术人员参考。

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 13501256678 13801310933

图书在版编目(CIP)数据

数据结构实例教程/杨晓光编著. —北京:清华大学出版社;北京交通大学出版社,2008.12
(高等学校计算机科学与技术教材)

ISBN 978-7-81123-405-3

I. 数… II. 杨… III. 数据结构-高等学校-教材 IV. TP311.12

中国版本图书馆 CIP 数据核字(2008)第 146037 号

责任编辑:谭文芳

出版发行:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>
北京交通大学出版社 邮编:100044 电话:010-51686414 <http://press.bjtu.edu.cn>

印刷者:北京东光印刷厂

经 销:全国新华书店

开 本:185×260 印张:21.5 字数:550千字

版 次:2008年12月第1版 2008年12月第1次印刷

书 号:ISBN 978-7-81123-405-3/TP·440

印 数:1~5000册 定价:34.00元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@bjtu.edu.cn。

前 言

数据结构是计算机专业和电子信息专业的重要专业基础课程之一。在计算机课程体系中起到承上启下的作用,它是操作系统、软件工程、数据库、编译原理、计算机图形学等课程的基础。近年来,随着计算机应用领域的不断拓展,许多非计算机专业也把数据结构作为重要的选修课程之一,以期加强学生程序设计能力。

作者在多年的数据结构课程教学实践中发现,初学者普遍感觉数据结构抽象难懂,对于数据结构应用于何处、如何应用也比较迷茫。作者也一直在思考这个问题,经过深入分析以后,感到造成这一现象的原因是,现在的教科书多以抽象理论讲解为主,不考虑应用,与实际应用脱节。在本书编写过程中,尝试引用一些实用案例,这些案例都是我们身边发生的事情,既生动有趣,又能诠释数据结构理论,从而变抽象为具体。

除此之外,本书还具有以下特色。

(1) 在讲解每种新类型时,由一个贴近生活实际的小例子引入,如以奥运会门票预订为例引入队列,先给学生一个感性认识,然后再过渡到抽象的理论,使学生易于理解、易于接受。

(2) 在讲解各种数据结构与算法的同时,给出大量例题。通过例题进一步阐述数据结构理论,同时引导学生灵活运用理论解决问题,从而达到举一反三的目的。

(3) 在每一章的最后部分提供一个综合实例,从实用角度诠释如何用数据结构解决现实生活中存在的问题,如用查找和排序生成“十大流行歌手排行榜”。这样会给学生以学以致用的感受,激发他们的学习兴趣。

(4) 在每章的最后提供了大量精选习题。通过这些习题的训练,可使学生巩固所学知识,进而灵活运用所学知识。

(5) 每章还提供一至两个实验,便于学生上机练习。

(6) 附录 A 给出一个综合测试,用于检验学习效果。

(7) 书中所有算法都由 C 语言实现(均在 Visual C++6.0 下调试通过),并有详尽的注释,有些还给出测试程序,供学生验证和模仿,以加深学生对算法的理解。

全书共分九章。第 1 章介绍数据结构、抽象数据类型与算法的一些基本概念;第 2 章介绍线性表的逻辑结构和两种存储结构(顺序表和链表),以及基本操作的实现;第 3 章介绍栈和队列,讨论栈和队列特点,相应的存储结构及基本操作的实现,给出栈和队列的一些应用实例;第 4 章介绍串的基本概念、模式匹配算法,讨论串的各种存储结构,以及基本操作的实现;第 5 章介绍数组和广义表的基本概念,讨论数组和广义表的存储结构,以及特殊矩阵、稀疏矩阵和广义表的基本运算;第 6 章介绍树与二叉树,包括树与二叉树的定义与表示,讨论树与二叉树的存储结构,以及树与二叉树的遍历和相互转换;第 7 章介绍图,包括图的基本概念,讨论图的存储结构,以及图的一些应用;第 8 章介绍各种常见的查找算法及实现;第 9 章介绍各种常见的内部排序算法及实现;附录 A 为综合测试;附录 B 为部分习

题参考答案。为了便于学生学习与理解数据结构，书中给出 121 道例题，430 道习题，7 个综合实例，13 个实验，1 个综合测试。

本书可作为本科生一个学期的教学内容，参考学时为 48 ~ 64 学时。如果低于 64 学时，可适当删减带 * 的章节。本书配有教学用课件，以及书中的所有源代码，有需要的读者可通过“<http://press.bjtu.edu.cn>”网站直接下载或与出版社联系。

在本书的编写过程中，参考了大量国内外数据结构教材，其中主要参考教材列于“参考文献”中，这里对这些教材的作者一并表示感谢。

本书由杨晓光编写。李兰友教授在百忙中抽出时间审阅了全书，并提出了许多宝贵意见。参加本书编写的还有郭文平、傅岚岚、杨晓君、杨亚红、郑志荣、马延宏、杨伟心等。虽然本书作者极尽努力，但错误和不足之处仍不可避免，敬请读者批评指正。

本书可作为高等院校计算机专业和电子信息专业本、专科学生的教材和参考书，亦适合于工程技术人员参考。

编者
2008 年 10 月

目 录

第1章 数据结构概述	1
1.1 数据结构研究的内容	1
1.2 基本概念和术语	3
1.3 抽象数据类型	5
1.4 算法分析	7
1.4.1 算法及性质	7
1.4.2 算法度量及分析	8
1.5 习题	12
1.6 实验	14
第2章 线性表	16
2.1 线性表的定义及其操作	16
2.1.1 线性表的定义	16
2.1.2 线性表的抽象数据类型	17
2.2 顺序表	18
2.2.1 顺序表的定义	18
2.2.2 顺序表的基本操作	19
2.3 链表	27
2.3.1 单链表	27
2.3.2 循环链表	37
2.3.3 双向链表	38
2.3.4 静态链表	42
2.4 顺序表与链表的比较	44
2.5 综合实例——文具店的货品管理	45
2.6 习题	51
2.7 实验	54
第3章 栈和队列	56
3.1 栈的定义及其操作	56
3.1.1 栈的定义	56
3.1.2 栈的抽象数据类型	58
3.2 顺序栈	58
3.2.1 顺序栈的定义	58
3.2.2 顺序栈的基本操作	59
3.3 链栈	63

3.3.1	链栈的定义	63
3.3.2	链栈的基本操作	64
3.4	队列的定义及其操作	68
3.4.1	队列的定义	68
3.4.2	队列的抽象数据类型	69
3.5	顺序队列	70
3.5.1	顺序队列的定义	70
3.5.2	顺序队列的基本操作	73
3.6	链队列	78
3.6.1	链队列的定义	78
3.6.2	链队列的基本操作	79
3.7	栈与队列的应用	84
3.7.1	数制转换	84
*3.7.2	表达式计算	86
3.7.3	输入输出缓冲区	89
3.8	综合实例——停车场管理	89
3.9	习题	94
3.10	实验	97
第4章	串	100
4.1	串的基本概念及其操作	100
4.1.1	串的基本概念	100
4.1.2	串的抽象数据类型	101
4.2	串的顺序存储结构	101
4.2.1	串的定长顺序存储表示	102
4.2.2	串的堆存储表示	107
4.3	串的链式存储结构	110
4.4	串的模式匹配	116
4.4.1	Brute-Force 算法	116
*4.4.2	KMP 算法	118
4.5	综合实例——简易文本编辑软件	123
4.6	习题	128
4.7	实验	129
第5章	数组和广义表	131
5.1	数组的基本概念及其操作	131
5.1.1	数组的基本概念	131
5.1.2	数组的抽象数据类型	132
5.2	数组的顺序存储	132
5.3	特殊矩阵及其压缩存储	133
5.3.1	对称矩阵与三角矩阵	133

5.3.2 对角矩阵	135
5.4 稀疏矩阵	136
5.4.1 三元组顺序表表示	136
5.4.2 十字链表表示	142
5.5 广义表	147
5.5.1 广义表的基本概念	147
5.5.2 广义表的抽象数据类型	149
5.5.3 广义表的存储结构	149
5.6 综合实例—— n 阶魔方	156
5.7 习题	159
5.8 实验	162
第6章 树和二叉树	163
6.1 树	163
6.1.1 树的定义	163
6.1.2 树的表示方法	165
6.1.3 树的抽象数据类型	165
6.1.4 树的存储结构	166
6.2 二叉树	168
6.2.1 二叉树的定义	169
6.2.2 二叉树的性质	172
6.2.3 二叉树的存储结构	174
6.2.4 二叉树的遍历	179
*6.3 线索二叉树	186
6.3.1 线索二叉树的定义	186
6.3.2 线索二叉树的操作	187
6.4 森林	192
6.4.1 树、森林与二叉树的转换	192
6.4.2 树和森林的遍历	195
6.5 哈夫曼树及其应用	196
6.5.1 哈夫曼树	196
6.5.2 哈夫曼编码	198
6.6 综合实例——高校社团管理	202
6.7 习题	207
6.8 实验	211
第7章 图	212
7.1 图的基本概念	212
7.1.1 图的定义	212
7.1.2 图的抽象数据类型	215
7.2 图的存储结构	216

7.2.1	邻接矩阵	216
7.2.2	邻接表	221
*7.2.3	十字链表	225
*7.2.4	邻接多重表	225
7.3	图的遍历	226
7.3.1	深度优先搜索	226
7.3.2	广度优先搜索	228
7.4	最小生成树	231
7.4.1	普里姆算法	232
7.4.2	克鲁斯卡尔算法	234
7.5	最短路径	235
7.5.1	从某个顶点到其余顶点的最短路径	235
*7.5.2	每对顶点之间的最短路径	237
7.6	拓扑排序和关键路径	239
7.6.1	拓扑排序	239
7.6.2	关键路径	240
7.7	综合实例——故宫导游咨询	243
7.8	习题	248
7.9	实验	251
第8章	查找	252
8.1	查找的基本概念	252
8.2	顺序查找	253
8.3	折半查找	254
8.4	分块查找	255
8.5	二叉排序树	256
8.5.1	二叉排序树的定义	256
8.5.2	二叉排序树的基本操作	257
8.6	B-树	263
8.6.1	B-树的定义	263
8.6.2	B-树的基本操作	264
8.7	哈希表	269
8.7.1	哈希表的定义	269
8.7.2	哈希函数的构造方法	270
8.7.3	处理冲突的方法	272
8.7.4	哈希表的基本操作	274
8.8	综合实例——十大流行歌手排行榜	278
8.9	习题	284
8.10	实验	287
第9章	排序	288

9.1	排序的基本概念	288
9.2	插入排序	289
9.2.1	直接插入排序	289
9.2.2	折半插入排序	291
9.2.3	希尔排序	292
9.3	交换排序	294
9.3.1	冒泡排序	294
9.3.2	快速排序	295
9.4	选择排序	298
9.4.1	直接选择排序	299
9.4.2	堆排序	300
9.5	归并排序	306
9.6	基数排序	308
9.7	各种排序方法的比较	310
9.8	习题	311
9.9	实验	314
附录 A	综合测试	315
附录 B	部分习题答案	319
参考文献		334

第 1 章 数据结构概述

随着计算机技术的飞速发展,以及计算机的日益普及,计算机应用的范围越来越广泛。从最初的数值计算,发展到现在的数据处理、自动控制、信息处理、人工智能、情报检索和办公自动化等非数值计算领域。所处理的数据也从简单的数字,发展到复杂的文字、图形、图像、音频、视频和动画等具有结构的数据。因此,要想高效地处理好这些数据,必须深入研究数据本身的特性、数据之间的关系,以及如何有效地将数据组织存储在计算机内。这正是数据结构所要研究的主要问题。

本章主要介绍数据结构的基本概念,数据的逻辑结构、存储结构及其关系,抽象数据类型,算法及算法时间复杂度分析。

1.1 数据结构研究的内容

在介绍数据结构之前,我们先看几个例子。

【例 1-1】 公司员工信息管理。

某公司有“王清”、“李丽圆”、“张娟”、“张爱民”等员工。现公司想要用计算机管理其员工信息,要求能够做以下操作:

- ✎ 当招聘新员工时,能够把员工信息添加进来;
- ✎ 当有员工辞职时,能够删除该员工信息;
- ✎ 可以修改员工信息;
- ✎ 能够以某种方式检索员工信息。

分析:

通过对以上问题的描述,我们可以把公司员工信息用表 1-1 表示出来。其中,每个员工的信息由员工号、姓名、性别、年龄、住址、电话、所属部门等组成,员工数据按照一定的顺序线性排列。这就是解决该问题的模型(线性表)。有了模型以后,就可以围绕该模型设计算法,即实现员工信息的添加、修改、删除、检索等操作。

表 1-1 员工信息表

员工号	姓名	性别	年龄	住址	电话	所属部门
01002	王清	男	25	南京路 23 号	3564	财务
01003	李丽圆	女	28	甘肃路 59 号	3698	总务
01004	张娟	女	20	杭州路 2 号	2346	经理办公室
01005	张爱民	男	45	河北路 9 号	5896	销售
.....						

类似地还有学籍管理系统、飞机订票系统、图书馆书籍管理、学生选课系统等,它们都有共

同点,即被处理的对象之间具有线性关系。这就是一类数据结构——线性数据结构。

【例 1-2】 NBA 季后赛对阵形势。

在每个赛季,进入季后赛的有 16 只球队,分成东部和西部两个赛区,每赛区 8 只球队进行淘汰赛,胜者进入下一轮,这样在两个赛区中分别产生一名冠军,最后在这两名冠军之间产生总冠军。现在希望得到各队对阵形势,以及输赢情况。

分析:

通过对以上问题的描述,我们可以把各队对阵形势用图 1-1(篇幅所限只给出了 8 只球队)表示出来。这是一颗倒长的树,树根在上面,树叶在下面。树根就是总冠军,树叶就是参赛的各支球队。内部的枝杈和结点表示两支球队对阵情况和获胜一方。如果要查询凯尔特人队的比赛情况,可以从树根开始沿枝杈到达表示凯尔特人的树叶即可。树也是一种数据结构,也能表达某些非数值计算问题。

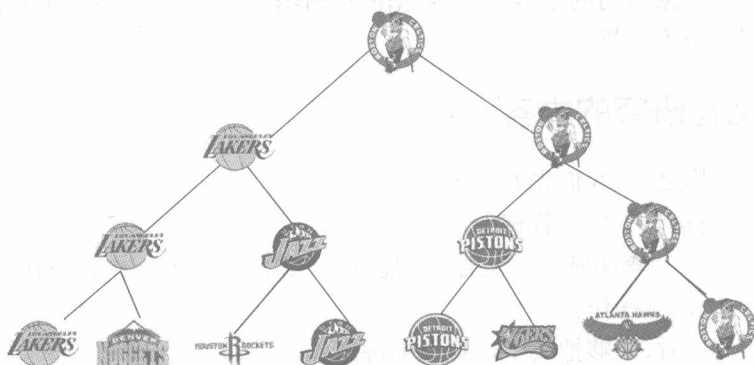


图 1-1 球队对阵形势

【例 1-3】 泰山 3 日游。

某旅行社想要开辟泰山旅游线路,为了降低成本,决定采用火车作为交通工具,但希望乘车时间越少越好,以便增加游览时间,从而吸引更多游客。

分析:

该问题可用图 1-2 所示的铁路交通图来解决,寻找到泰山的所有乘车线路中花费时间最少的线路。类似地还可以解决换乘次数最少,费用最少等问题。通常,这类问题的数学模型是图状数据结构。

从例 1-1、例 1-2 和例 1-3 可以看到,这些我们身边发生的事情,都不是数值计算问题,而是非数值计算问题。这些非数值计算问题不能通过列方程、解方程等数学方法来求解,而是用线性表、树、图等数据结构来描述。求解这类问题,人们通常的做法是:首先对问题进行抽象,获得表示实际问题的一个模型;然后围绕该模型设计求解问题的算法;最后用程序实现之。获得模型的实质就是分析问题,寻找要操作的对象,以及对象之间的关系。而数据结构正是实际问题中操作对象,以及这些对象之间关系的数学抽象。它反映了这些操作对象的内部数据构成,即数据由哪几部分构成,以什么方式构成,呈现什么样的结构,在计算机内如何存储等。因此,数据结构是一门研究非数值计算程序设计问题中计算机的操作对象以及它们之间的关系和操作等的学科。它的主要研究范围如下:

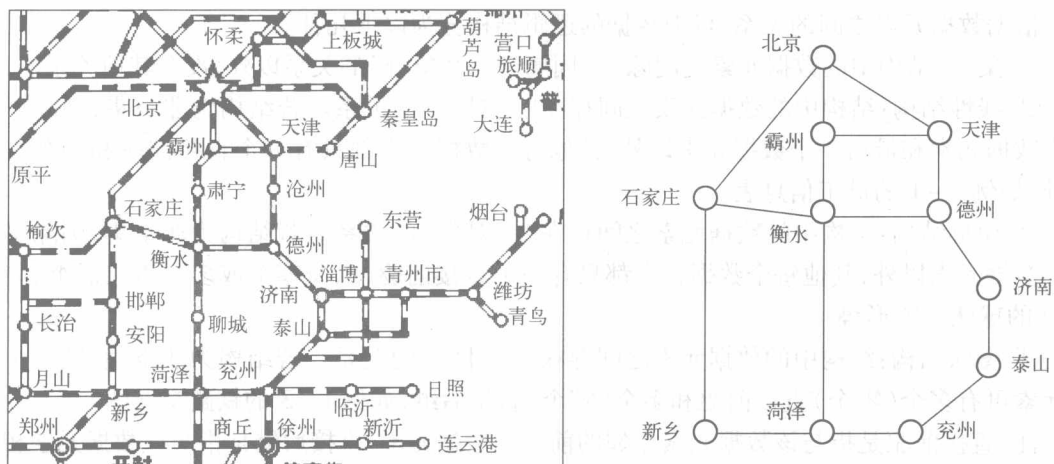


图 1-2 铁路交通图

- ◇ 数据元素之间固有的逻辑关系——数据逻辑结构；
- ◇ 数据元素及关系在计算机内的表示——数据存储结构；
- ◇ 对数据结构的操作——算法。

1.2 基本概念和术语

1. 数据

数据是用来描述现实世界的数字、字符、图像、声音,以及能够输入到计算机中并能被计算机处理的符号集合。例如,描述数值的整数、实数,描述图书馆中书目的字符串等。

2. 数据元素

数据元素是数据的基本单位,是数据这个集合中的个体,也称为元素、结点、顶点、记录。例如,图书馆书目数据中,有“数据结构”、“C 语言”、“数据库”等数据元素,整数数据中的“1, 2, …”等数据元素。

一个数据元素可以由若干个数据项组成。数据项是数据不可分割的最小标识单位,也称为字段、域、属性。例如,书目信息可由“书名”、“作者”、“出版社”等数据项组成。员工信息可由“员工号”、“姓名”、“性别”、“年龄”、“住址”、“电话”、“所属部门”等数据项组成。

3. 数据对象

数据对象是具有相同性质的数据元素的集合,是数据的一个子集。例如,书目数据对象是集合{“数据结构”,“C 语言”,“数据库”},桥牌数据对象是集合{2,3, …, 10, J, Q, K, A},英文字符数据对象是集合{a, b, …, z, A, B, …, Z}。

4. 数据结构

数据结构是相互之间存在一种或多种特定关系的数据元素的集合。

数据结构包括三方面的内容:数据的逻辑结构、数据的存储结构和数据的操作。

(1) 数据的逻辑结构

数据的逻辑结构是指数据元素之间存在的固有的逻辑关系,常简称为数据结构。

数据的逻辑结构是从逻辑关系上描述数据,与数据的存储无关,是独立于计算机的。数据的逻辑结构可以看作是从具体问题抽象出来的数学模型。

依据数据元素之间的关系,可把数据的逻辑结构分为以下几种。

① 集合:结构中的数据元素之间除了“同属于一个集合”的关系以外,没有其他关系。

② 线性结构:结构中的数据元素之间存在“一对一”的关系。若结构为非空集,则除了第一个数据元素和最后一个数据元素以外,其他每个数据元素都只有一个直接前驱和一个直接后继,如例 1-1 的员工信息表。

③ 树形结构:结构中的数据元素之间存在“一对多”的关系。若结构为非空集,则除了第一个数据元素以外,其他每个数据元素都只有一个直接前驱,以及零个或多个直接后继,如例 1-2 的球队对阵形势。

④ 图状结构:结构中的数据元素之间存在“多对多”的关系。若结构为非空集,则每个数据元素可有多个(零个)直接前驱和多个(零个)直接后继,如例 1-3 的铁路交通图。

注:直接前驱是指与该数据元素相邻的前一个数据元素;直接后继是指与该数据元素相邻的后一个数据元素。

图 1-3 为上述四种逻辑结构的例子。



图 1-3 数据逻辑结构

数据结构可以用一个二元组来表示,即

$$\text{Data_Structure} = (D, R)$$

其中,D 是某个数据对象,R 是该对象中所有数据元素之间的关系的有限集合。

【例 1-4】 例 1-1 的数据结构。

$$\text{Employee} = (N, R)$$

$$N = \{\text{王清, 李丽圆, 张娟, 张爱民}\}$$

$$R = \{ \langle \text{王清, 李丽圆} \rangle, \langle \text{李丽圆, 张娟} \rangle, \langle \text{张娟, 张爱民} \rangle \}$$

(2) 数据的存储结构

数据元素及其关系在计算机内的表示称为数据的存储结构。

要想用计算机处理数据,就必须把数据的逻辑结构映射为数据的存储结构。逻辑结构可以映射为以下 4 种存储结构。

① 顺序存储结构:把逻辑上相邻的数据元素存储在物理位置也相邻的存储单元中,借助元素在存储器中的相对位置来表示数据元素之间的逻辑关系。由此得到的存储结构称为顺序存储结构。

② 链式存储结构:借助指针表达数据元素之间的逻辑关系。不要求逻辑上相邻的数据元素在物理位置上也相邻。由此得到的存储结构称为链式存储结构。

③ 索引存储结构:在存储数据元素的同时,还建立附加的索引表。通过索引表,可以找到存储数据元素的结点。

④ 散列存储结构:根据散列函数和处理冲突的方法确定数据元素的存储位置。

例如,一个包含数据元素“王清,李丽圆,张娟,张爱民”的线性结构,其顺序存储结构和链式存储结构如图1-4所示。

顺序存储结构



链式存储结构



图1-4 数据存储结构

(3) 数据的操作

数据的操作是在数据的逻辑结构上定义的操作算法,如插入、删除、检索等。

需要注意的是,数据的逻辑结构和数据的存储结构是指一个事物的两个方面,而不是两个事物。二者相辅相成,不可分割。同时,一种数据逻辑结构可以映射为多种数据存储结构,具体映射为哪种存储结构,视具体要求而定,主要考虑运算方便及算法的时间空间效率的要求。

1.3 抽象数据类型

在C语言中定义的整型(int)是-32 768~32 767范围内所有整数的集合,以及可以对整数进行的加、减、乘、除、求余等运算。一个整数在计算机内以定点有符号二进制数补码形式存储。但是,在C语言中使用整型时,程序员并不需要知道整数在计算机内是如何表示的,其操作是如何实现的,只需要知道如何定义整数,以及如何用+、-、*、/运算符就可以了,这样就把整数的使用与实现分开了。实际上,整型就是一个抽象数据类型。

1. 抽象数据类型

抽象数据类型(Abstract Data Type, ADT)是指数据元素集合以及定义在该集合上的一组操作。

所谓“抽象”是指与具体实现无关,仅考虑在数据元素集合上能做什么,而不考虑如何做,就像C语言中的整型一样。这样做的好处是,在分析问题时只研究如何使用它,而不必过早考虑实现的细节,从而就可以将注意力集中在问题的本质上,而不必过多地考虑一些细节问题。

ADT就像电路图,它是设计师对所设计电子产品的精确描述,工人必须按照电路图的要求生产,才能制造出符合要求的电子产品。但ADT只做功能性说明,对具体的实现并不作过多的限制,数据类型、数据结构、完成操作的算法可以自由选择。

实现抽象数据类型时,要实现存储数据元素的存储结构,以及实现操作的算法。可以用面向对象语言实现抽象数据类型,也可以使用非面向对象程序设计语言实现抽象数据类型。如

果采用面向对象程序设计语言,则可以使用类来实现抽象数据类型;如果采用非面向对象程序设计语言,则可以使用结构体实现数据存储结构,使用函数实现数据操作。

2. 抽象数据类型的表示

本书按以下格式表示抽象数据类型:

ADT 抽象数据类型名
数据元素集合: 数据元素集合的定义
基本操作: 基本操作的定义

其中,数据元素集合用自然语言描述,基本操作用伪码描述,并规定基本操作的格式为:

中文名(操作名):含义

【例 1-5】 抽象数据类型“字符串”的定义。

ADT String
数据元素集合: 字符的一个有限序列。
基本操作: ✧ 求串长(StrLen):求取字符串中字符的个数 ✧ 取子串(SubStr):获取字符串中的一个连续字符序列 ✧ 定位串(Index):查找是否存在子串 ✧ 串连接(Concat):连接两个字符串形成一个新串 ✧ 串比较(StrCmp):比较两个串的大小 ✧ 判空串(StrEmpty):判断所给字符串是否为空串 ✧ 串替换(StrReplace):替换字符串中指定的所有子串

【例 1-6】 某公司经营着一个足球场,它需要一个售票的软件,以便能够查询到哪些票被售出,哪些票还未售出。

【设计思路】经过抽象以后,该问题的数据元素集合是足球场中的所有座位。操作是:哪个座位的票被售出了,则作标记 1;没有售出,则作标记 0。

ADT Ticket
数据元素集合: 由表示排和号的两个整数限制的一个固定长度的元素序列,以及一个表示球票总数的一个整数。
基本操作: ✧ 售票(Sell):销售球票 ✧ 预订(Book):预订球票或取消预订 ✧ 查询余票(Balance):查询还有多少未售出的球票 ✧ 是否售出(Index):查询指定的球票是否售出

1.4 算法分析

瑞士计算机科学家 N. Wirth 指出“程序 = 数据结构 + 算法”。它描述了计算机程序是由组织信息的数据结构和处理信息的算法组成。二者相辅相成,不可分割。

1.4.1 算法及性质

1. 算法

讲算法之前先看一个求解问题的过程。

【例 1-7】 已知 n 个整数,求 n 个整数中的最大数。

这是一个非常简单的问题,下面直接给出求解过程:

① 将第 1 个数赋值给 $\max$;

② 初始化计数变量 i 为 1;

③ 当 $i < n$ 时,执行以下内容:

 ✎ 比较 $a[i]$ 与 $\max$,若 $a[i]$ 大于 $\max$,则将 $a[i]$ 赋值给 $\max$;

 ✎ i 自增 1;

④ 返回 $\max$ 的值。

这就是一个求最大数的算法。算法就是求解问题的一系列步骤的集合。它以一组值作为输入,并产生一组值作为输出。

通常用计算机程序来实现算法,计算机执行程序中的语句,实现对问题的求解。

算法可以用伪码形式描述(如例 1-7),也可以用程序设计语言描述。

【例 1-8】 对例 1-7 的算法,用程序设计语言来描述。

```
int Max(int a[],int n)
{
    int i,max;
    max=a[0];
    for(i=1;i<=n;i++)
        if(a[i]>max)    max=a[i];
    return max;
}
```

2. 算法的性质

所有的算法都必须满足以下性质。

✎ 可行性:算法中描述的操作都是用已经实现的基本运算组成的。

✎ 有穷性:算法必须在有限步或有限的时间内完成。

✎ 确定性:算法中每一条指令必须有确切的含义,不能有二义性。

✎ 有输入:算法应该有零或多个输入量。

✎ 有输出:算法应该有一个或多个输出量。

算法的有穷性是算法和程序的分界点,程序并不要求在有限的步骤内或有限的时间内结束,比如操作系统,而算法却有这个要求。