

数据 结构 (Java 语言描述)

主编 库波 曹静

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

数据结构 (Java 语言描述)

主编 库波 曹勤
副主编 杨国勋 谢日星
参编 宋静 程宁
主审 王路群

内 容 提 要

本书共 10 章, 主要内容包括: 绪论、线性表、栈和队列、串、多维数组和广义表、树、图、查找、排序、文件等。各章中所涉及的数据结构与算法均给予了 Java 语言描述 (所有程序都运行通过), 以便于读者巩固和提高运用 Java 语言进行程序设计的能力与技巧。

本书在内容的选取、概念的引入、文字的叙述以及例题和习题的选择等方面, 都力求遵循面向应用、逻辑结构简明合理、由浅入深、深入浅出、循序渐进、便于自学的原则, 突出其实用性与应用性。

本书为高等院校计算机专业教材, 也适合各校非计算机专业辅修计算机专业课程的学生使用, 还可以供其他从事计算机软件开发的科技人员自学参考。

版权专有 侵权必究

图书在版编目 (CIP) 数据

数据结构: Java 语言描述/库波, 曹静主编. —北京: 北京理工大学出版社, 2012. 8

ISBN 978 - 7 - 5640 - 6428 - 0

I. ①数… II. ①库… ②曹… III. ①数据结构 - 高等学校 - 教材 ②JAVA 语言 - 程序设计 - 高等学校 - 教材 IV. ①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字 (2012) 第 179819 号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775(办公室) 68944990(批销中心) 68911084(读者服务部)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 北京高岭印刷有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 13

字 数 / 302 千字

版 次 / 2012 年 8 月第 1 版 2012 年 8 月第 1 次印刷

责任编辑 / 钟 博

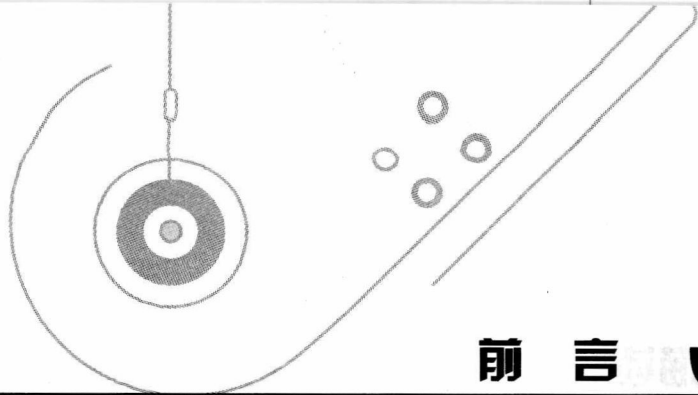
印 数 / 1 ~ 1500 册

责任校对 / 杨 露

定 价 / 41.00 元

责任印制 / 王美丽

图书出现印装质量问题, 本社负责调换



数据结构

前言

Java

语言描述

21 世纪是科学技术高速发展的信息时代,而计算机则是处理信息的主要工具,因此,人们已经认识到,计算机知识已成为人类当代文化的一个重要组成部分。

计算机科学技术以惊人的速度向前发展,它的广泛应用已从传统的数值计算领域发展到各种非数值计算领域。在非数值计算领域里,数据处理的对象已从简单的数值发展到一般的符号,进而发展到具有一定结构的数据。在这里,面临的主要问题是:针对每一种新的应用领域的处理对象,如何选择合适的数据表示(结构),如何有效地组织计算机存储,并在此基础上又如何有效地实现对象之间的“运算”关系。传统的解决数值计算的许多理论、方法和技术已不能满足解决非数值计算问题的需要,必须进行新的探索。数据结构就是研究和解决这些问题的重要基础理论。因此,“数据结构”课程已成为计算机类专业的一门重要专业基础课。

数据结构是程序设计的中级课程,主要培养学生分析数据、组织数据的能力,告诉学生如何编写效率高、结构好的程序。本书作为高职高专规划教材之一,在内容的选取、概念的引入、文字的叙述以及例题和习题的选择等方面,都力求遵循面向应用、逻辑结构简明合理、由浅入深、深入浅出、循序渐进、便于自学的原则,突出其实用性与应用性。全书共分 10 章,第 1 章主要讲述数据结构和算法的基本概念;第 2 章至第 7 章分别讲述线性表、栈和队列、串、多维数组和广义表、树和图这几种基本数据结构的特点、存储方法和基本运算,书中安排了相当的篇幅来介绍这些基本数据结构的实际应用;第 8 章和第 9 章讲述查找和排序的基本原理与方法;第 10 章围绕数据在外存上的组织方法介绍了文件的若干基本结构。各章所涉及的数据结构与算法,均给予了 Java 语言描述,以便于读



数据结构 Java 语言描述

者巩固和提高运用 Java 语言进行程序设计的能力与技巧。书中所有程序都运行通过,并可从北京理工大学出版社的网站免费下载。

本书配有电子教案,此教案用 PowerPoint 制作,可以自由修改,使用该教材的教师可以与北京理工大学出版社联系,免费获取此电子教案。

本书由库波、曹静任主编,杨国勋、谢日星任副主编,其中第 1、3、6、7 章由库波编写,第 2、4、5 章由杨国勋、谢日星编写,第 8、9 章由孙海南、曹静编写,第 10 章由宋静、程宁编写,王路群、库波统编全稿。参加本书编写大纲讨论的教师还有郭俐、余恒芳、秦培煜、肖菲、张克斌、袁晓曦等。

由于编者水平有限,书中不妥或错误之处在所难免,殷切希望广大读者批评指正。

编 者

目录

第1章 绪论	001
1.1 数据结构的基本概念	001
1.1.1 为什么要学习数据结构	001
1.1.2 什么是数据结构	002
1.1.3 基本概念和术语	003
1.1.4 数据类型与抽象数据类型	005
1.2 算法和算法分析简介	007
1.2.1 算法	007
1.2.2 算法的描述	007
1.2.3 算法评价	013
本章小结	014
习题一	014

第2章 线性表	015
2.1 线性表的逻辑结构	015
2.2 线性表的顺序存储结构	016
2.2.1 线性表的顺序存储结构	016
2.2.2 线性表在顺序存储结构下的运算	017
2.3 线性表的链式存储结构	022
2.3.1 线性链表	022
2.3.2 循环链表	027
2.3.3 双向链表	027
2.4 一元多项式的表示及相加	030
本章小结	034
习题二	034

第3章 栈和队列	035
3.1 栈	035
3.1.1 栈的定义及其运算	035
3.1.2 栈的顺序存储结构	037
3.1.3 栈的链式存储结构	038
3.2 算术表达式求值	040
3.3 队列	044
3.3.1 队列的定义及其运算	044

3.3.2 队列的顺序存储结构	046
3.3.3 队列的链式存储结构	051
3.3.4 其他队列	052
本章小结	053
习题三	054

第4章 串	055
4.1 串的基本概念	055
4.1.1 串的定义	055
4.1.2 主串和子串	055
4.2 串的存储结构	056
4.2.1 串值的存储	056
4.2.2 串名的存储映像	057
4.3 串的基本运算及其实现	058
4.3.1 串的基本运算	058
4.3.2 串的基本运算及其实现	059
4.4 文本编辑	062
本章小结	063
习题四	064

第5章 多维数组和广义表	065
5.1 多维数组	065
5.1.1 多维数组的概念	065
5.1.2 多维数组在计算机内的存放	066
5.2 多维数组的存储结构	066
5.2.1 行优先顺序	066
5.2.2 列优先顺序	066
5.3 特殊矩阵及其压缩存储	067
5.3.1 特殊矩阵	067
5.3.2 压缩存储	068
5.4 稀疏矩阵	070
5.4.1 稀疏矩阵的存储	070
5.4.2 稀疏矩阵的运算	075
5.5 广义表	082
5.5.1 基本概念	082
5.5.2 存储结构	083
5.5.3 基本运算	085

本章小结	087
习题五	088

第6章 树

6.1 树的结构定义与基本操作	090
6.1.1 树的定义及相关术语	090
6.1.2 树的存储结构	091
6.1.3 树的基本操作	092
6.2 二叉树	092
6.2.1 二叉树的定义与基本操作	092
6.2.2 二叉树的性质	094
6.2.3 二叉树的存储结构	095
6.2.4 树与二叉树的相互转换	099
6.3 遍历二叉树	100
6.3.1 先序遍历	101
6.3.2 中序遍历	102
6.3.3 后序遍历	103
6.3.4 层次遍历	104
6.4 线索二叉树	105
6.4.1 线索二叉树的基本概念	105
6.4.2 中序次序线索化算法	107
6.5 二叉排序树	111
6.5.1 二叉排序树的定义	112
6.5.2 二叉排序树的生成	112
6.5.3 删除二叉排序树上的结点	115
6.6 哈夫曼树和哈夫曼算法	117
6.6.1 哈夫曼树的定义	118
6.6.2 构造哈夫曼树——哈夫曼算法	119
6.6.3 哈夫曼树的应用	120
本章小结	122
习题六	123

第7章 图

7.1 基本定义和术语	124
7.1.1 基本定义和术语	124
7.1.2 抽象数据类型	126
7.2 图的存储结构	129

7.2.1 邻接矩阵	129
7.2.2 邻接表	130
7.3 图的遍历	132
7.3.1 深度优先搜索法	132
7.3.2 广度优先搜索法	135
7.4 最小生成树	137
7.5 最短路径	143
7.5.1 单源点最短路径	143
7.5.2 所有顶点之间的最短路径	146
7.6 拓扑排序	150
本章小结	151
习题七	152

第8章 查找	154
8.1 顺序查找	154
8.2 折半查找	155
8.3 分块查找	157
8.4 哈希法	158
8.4.1 哈希表和哈希函数的概念	159
8.4.2 哈希函数的构造方法	159
8.4.3 冲突处理	161
本章小结	164
习题八	165

第9章 排序	166
9.1 插入排序	166
9.1.1 线性插入排序	166
9.1.2 折半插入排序	168
9.2 希尔排序	169
9.3 选择排序	171
9.4 堆排序	173
9.5 快速排序	176
9.6 归并排序	180
9.7 基数排序	182
9.8 外部排序	183
9.9 各种排序方法的比较	185

本章小结	186
习题九	186

第 10 章 文件

10.1 文件的基本概念	187
10.2 顺序文件	188
10.3 索引文件	190
10.4 索引顺序文件	191
10.5 直接存取文件	192
10.6 多关键字文件	194
10.6.1 多重表文件	194
10.6.2 倒排文件	195
本章小结	196
习题十	196

参考文献

第1章

绪论

本章学习导读

本章介绍了数据结构这门学科诞生的背景、发展历史以及在计算机科学中所处的地位,重点介绍了与数据结构有关的概念和术语,读者学习本章后应能掌握数据、数据元素、逻辑结构、存储结构、数据处理、数据结构、算法设计等基本概念,并了解如何评价一个算法的优劣。



1.1 数据结构的基本概念

1.1.1 为什么要学习数据结构

众所周知,20世纪40年代,电子数字计算机问世的直接原因是解决弹道学的计算问题。早期,电子计算机的应用范围,几乎只局限于科学和工程的计算,其处理的对象是纯数值性的信息,通常,人们把这类问题称为数值计算。

近三十年来,电子计算机的发展异常迅猛,这不仅表现在计算机本身运算速度不断提高、信息存储量日益扩大、价格逐步下降,更重要的是计算机广泛地应用于情报检索、企业管理、系统工程等方面,已远远超出了数值计算的范围,渗透到人类社会活动的一切领域。与此相应,计算机的处理对象也从简单的纯数值性信息发展到非数值性的和具有一定结构的信息。

因此,再把电子数字计算机简单地看做进行数值计算的工具,把数据仅理解为纯数值性的信息,就显得太狭隘了。现代计算机科学的观点,是把计算机程序处理的一切数值的、非数值的信息,乃至程序统称为数据(Data),而电子计算机则是加工处理数据(信息)的工具。

处理对象的转变导致系统程序和应用程序的规模越来越大,结构也相当复杂,单凭程序设计人员的经验和技巧已难以设计出效率高、可靠性强的程序,数据的表示方法和组织形式已成为影响数据处理效率的关键。因此,就要求人们对计算机程序所加工的对象进行系统的研究,即研究数据的特性以及数据之间存在的关系——数据结构(Data Structure)。

数据结构是随着电子计算机的产生和发展而发展起来的一门较新的计算机学科。数据结构所讨论的有关问题,早先是为解决系统程序设计中的具体技术而出现在《编译程序》和《操作系统》之中。“数据结构”作为一门独立的课程在国外是从1968年才开始设立的。在这之前,它的某些内容曾在其他课程,如表处理语言中有所阐述。1968年在美国一些大学的计算机系的教学计划中,虽然把“数据结构”规定为一门课程,但对课程的范围仍没有作明确规定。当时,数据结构几乎和图论,特别是和表、树的理论为同义语。随后,数据结构这个概念扩充到包括网络、集合代数论、格、关系等方面,从而变成了现在称为“离散结构”的内容。然而,数据必须在计算机中进行处理,因此,不仅要考虑数据本身的数学性质,而且还必须考虑数据的存储结构,这就进一步扩大了数据结构的内容。近年来,随着数据库系统的不断发展,在数据结构课程中又增加了文件管理(特别是大型文件的组织等)的内容。

1968年美国唐·欧·克努特教授开创了数据结构的最初体系,他所著的《计算机程序设计技巧》第一卷《基本算法》是第一本较为系统地阐述数据的逻辑结构和存储结构及其操作的著作。从60年代末到70年代初,出现了大型程序,软件也开始相对独立,且结构程序设计成为程序设计方法学的主要内容,人们就越来越重视数据结构,认为程序设计的实质是对确定的问题选择一种好的结构,加上设计一种好的算法。从70年代中期到80年代初,各种版本的数据结构著作就相继出现了。

目前在我国,“数据结构”已经不仅仅是计算机专业的教学计划中的核心课程之一,而且是其他非计算机专业的主要选修课程之一。

“数据结构”在计算机科学中是一门综合性的专业基础课。数据结构的研究不仅涉及计算机硬件(特别是编码理论、存储装置和存取方法等)的研究范围,而且和计算机软件的研究有着更为密切的关系,无论是编译程序还是操作系统,都涉及数据元素在存储器中的分配问题。在研究信息检索时也必须考虑如何组织数据,以便查找和存取数据元素更为方便。因此,可以认为数据结构是介于数学、计算机硬件和计算机软件三者之间的一门核心课程。我国从1978年开始,各院校先后开设了“数据结构”课。1982年全国计算机教育学术讨论会和1983年全国大专类计算机专业教学工作讨论会都把“数据结构”确定为计算机类各专业的骨干课程之一。这是因为,在计算机科学中,“数据结构”这一门课的内容不仅是一般程序设计(特别是非数值性程序设计)的基础,而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序的重要基础。

值得注意的是,数据结构的发展并未终结,一方面,面向各专门领域中特殊问题的数据结构得到研究和发 展,如多维图形数据结构等;另一方面,从抽象数据类型的观点来讨论数据结构,已成为一种新的趋势,越来越被人们所重视。由此可见,数据结构技术的产生时间并不长,它正处于迅速发展阶段。同时,随着电子计算机的发展和更新,新的数据结构将会不断出现。

1.1.2 什么是数据结构

什么是数据结构?这是一个难以直接回答的问题。一般来说,用计算机解决一个具体问题时,大致需要经过下列几个步骤:首先要从具体问题中抽象出一个适当的数学模型,然后设计一个解此数学模型的算法(Algorithm),最后编出程序、进行测试、调整直至得到最终解答。寻求数学模型的实质是分析问题,从中提取操作的对象,并找出这些操作对象之间含有的关系,然后用数学的语言加以描述。为了说明这个问题,我们首先举一个例子,然后再给出明确的含义。

假定有一个学生通讯录,记录了某校全体学生的姓名和相应的住址,现在要写一个算法,要求是,当给定任何一个学生的姓名时,该算法能够查出该学生的住址。这样一个算法的设计,将完全依赖于通讯录中的学生姓名及相应的住址是如何结构的,以及计算机是怎样存储通讯录中的信息。

如果通讯录中的学生姓名是随意排列的,其次序没有任何规律。那么,当给定一个姓名时,则只能对通讯录从头开始逐个与给定的姓名比较,顺序查对,直至找到所给定的姓名为止。这种方法相当费时间,效率很低。

然而,若对学生通讯录进行适当的组织,按学生所在班级来排列,并且再造一个索引表,这个表用来登记每个班级学生姓名在通讯录中的起始处的位置。这样一来,情况将大为改善。这时,当要查找某学生的住址时,则可先从索引表中查到该学生所在班级的学生姓名是从何处起始的,而后,就从起始处开始查找,而不必去查看其他部分的姓名。由于采用了新的结构,于是,就可写出一个完全不同的算法。

上述的学生通讯录就是一个数据结构问题。可以看到,计算机算法与数据的结构密切相关,算法无不依附于具体的数据结构,数据结构直接关系到算法的选择和效率。

下面,再对学生通讯录作进一步讨论。当有新学生入校时,通讯录需要添加新学生的姓名和相应的住址;在学生毕业离校时,应从通讯录中删除毕业学生的姓名和住址。这就要求在已安排好的结构上进行插入(Insert)和删除(Delete)。对于一种具体的结构,如何实现插入和删除?是把要添加的学生姓名和住址插入到前头,还是末尾,或是中间某个合适的位置上,插入后对原有的数据是否有影响?有什么样的影响?删除某学生的姓名和住址后,其他的数据(学生的姓名和住址)是否需要移动?若需要移动,则应如何移动?这一系列的问题说明,为适应数据的增加和减少的需要,还必须对数据结构定义一些运算。上面只涉及两种运算,即插入和删除运算。当然,还会提出一些其他可能的运算,如学生搬家后,住址变了,为适应这种需要,就应该定义修改(Modify)运算,等等。

这些运算,显然是由计算机来完成,这就需要设计相应的插入、删除和修改的算法。也就是说,数据结构还需要给出每种结构类型所定义的各种运算的算法。

通过以上讨论,可以直观地认为:数据结构是研究程序设计中计算机操作的对象以及它们之间的关系和运算的一门学科。

1.1.3 基本概念和术语

下面来认识与数据结构相关的一些重要的基本概念和术语。

① 数据(Data)

数据是人们利用文字符号、数字符号以及其他规定的符号对现实世界的事物及其活动所做的描述。在计算机科学中,数据的含义非常广泛,我们把一切能够输入到计算机中并被计算机程序处理的信息,包括文字、表格、声音、图像等,都称为数据。例如,一个个人书库管理程序所要处理的数据可能是一张如表1-1所示的表格。

② 数据元素(Data Element)

结点也叫数据元素,它是组成数据的基本单位。在程序中通常把结点作为一个整体进行考虑和处理。例如,在表1-1所示的个人书库中,为了便于处理,把其中的每一行(代表一本书)作为一个基本单位来考虑,故该数据由10个结点构成。

一般情况下,一个结点中含有若干个字段(也叫数据项)。例如,在表1-1所示的表格数据中,每个结点都由登录号、书号、书名、作者、出版社和价格等六个字段构成。字段是构成数据的最小单位。

表1-1 个人书库

登录号	书号	书名	作者	出版社	价格
000001	TP2233	Windows NT4.0 中文版教程	赵健雅	电子工业	28.00
000002	TP1844	Authorware 5.1 速成	孙强	人民邮电	40.00
000003	TP1684	Lotus Notes 网络办公平台	赵丽萍	清华大学	16.00
000004	TP2143	Access 2000 入门与提高	张堪	清华大学	22.00
000005	TP1110	PowerBuilder 6.5 实用教程	樊金生	科学	29.00
000006	TP1397	Delphi 数据库编程技术	刘前进	人民邮电	43.00
000007	TP2711	精通 MS SQL Server 7.0	罗会涛	电子工业	35.00
000008	TP3239	Visual C++ 实用教程	郑阿奇	电子工业	30.00
000009	TP1787	电子商务万事通	赵乃真	人民邮电	26.00
000010	TP42	数据结构	江涛	中央电大	18.80

③ 数据对象 (Data Object)

数据对象或数据元素类 (Data Element Class) 是具有相同性质的数据元素的集合。在某个具体问题中, 数据元素都具有相同的性质 (元素值不一定相等), 属于同一数据对象 (数据元素类), 数据元素是数据元素类的一个实例。

④ 数据结构 (Data Structure)

数据结构研究数据元素之间抽象化的相互关系和这种关系在计算机中的存储表示 (即所谓数据的逻辑结构和物理结构), 并对这种结构定义相适应的运算, 设计出相应的算法, 而且确保经过这些运算后所得到的新结构仍然是原来的结构类型。

根据数据元素间关系的不同特性, 通常有下列四类基本结构:

(1) 集合结构。在集合结构中, 数据元素间的关系是“属于同一个集合”。集合是元素关系极为松散的一种结构。

(2) 线性结构。该结构的数据元素之间存在着一对一的关系。

(3) 树型结构。该结构的数据元素之间存在着一对多的关系。

(4) 图形结构。该结构的数据元素之间存在着多对多的关系, 图形结构也称作网状结构。

图 1.1 为上述四类基本结构的示意图。

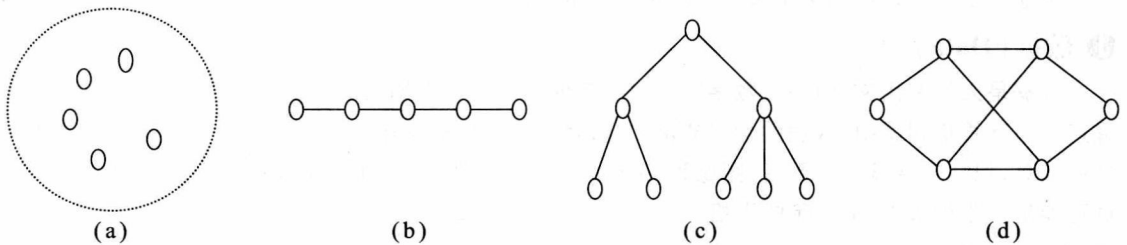


图 1-1 四类基本结构的示意图

(a) 集合结构; (b) 线性结构; (c) 树型结构; (d) 图形结构

⑤ 逻辑结构

结点和结点之间的逻辑关系称为数据的逻辑结构。

在表 1-1 所示的数据中, 各结点之间在逻辑上有一种线性关系, 它指出了 10 个结点在表中的排列顺序。根据这种线性关系, 可以看出表中第一本书是什么书, 第二本书是什么书, 等等。

⑥ 存储结构

数据及数据之间的关系在计算机中的存储表示称为数据的存储结构。

表 1-1 所示的表格数据在计算机中可以有多种存储表示, 例如, 可以表示成数组, 存放在内存中, 也可以表示成文件, 存放在磁盘上, 等等。

数据的存储结构可用以下四种基本存储方法得到:

1) 顺序存储方法

该方法把逻辑上相邻的结点存储在物理位置上相邻的存储单元里, 结点间的逻辑关系由存储单元的邻接关系来体现。

由此得到的存储表示称为顺序存储结构 (Sequential Storage Structure), 通常借助程序语言的数组描述。

该方法主要应用于线性的数据结构。非线性的数据结构也可通过某种线性化的方法来实现顺序存储。

2) 链接存储方法

该方法不要求逻辑上相邻的结点在物理位置上亦相邻,结点间的逻辑关系由附加的指针字段表示。由此得到的存储表示称为链式存储结构(Linked Storage Structure),通常借助于程序语言的指针类型描述。

3) 索引存储方法

该方法通常在存储结点信息的同时,还建立了附加的索引表。

索引表由若干索引项组成。若每个结点在索引表中都有一个索引项,则该索引表称为稠密索引(Dense Index)。若一组结点在索引表中只对应一个索引项,则该索引表称为稀疏索引(Spare Index)。索引项的一般形式是:

(关键字、地址)

关键字是能唯一标识一个结点的那些数据项。稠密索引中索引项的地址指示结点所在的存储位置;稀疏索引中索引项的地址指示一组结点的起始存储位置。

4) 散列存储方法

该方法的基本思想是:根据结点的关键字直接计算出该结点的存储地址。

以上四种基本存储方法,既可单独使用,也可组合起来对数据结构进行存储映像。

同一逻辑结构采用不同的存储方法,可以得到不同的存储结构。选择何种逻辑结构,视具体要求而定,主要考虑运算方便及算法的时空要求。

7 数据处理

数据处理是指对数据进行查找、插入、删除、合并、排序、统计以及简单计算等的操作过程。在早期,计算机主要用于科学和工程计算,进入80年代以后,计算机主要用于数据处理。据有关统计资料表明,现在计算机用于数据处理的时间比例达到80%以上,随着时间的推移和计算机应用的进一步普及,计算机用于数据处理的时间比例必将进一步增大。

1.1.4 数据类型与抽象数据类型

抽象数据类型是描述数据结构的一种理论工具。在介绍抽象数据类型之前先介绍一下数据类型的基本概念。

数据类型(Data Type)是一组性质相同的数据元素的集合以及加在这个集合上的一组操作。例如Java语言中就有许多不同的数据类型,包括数值型的数据类型、字符串、布尔型等数据类型。以Java中的int型为例,int型的数据元素的集合是 $[-2147483648, 2147483647]$ 间的整数,定义在其上的操作有加、减、乘、除四则运算,还有模运算等。

定义数据类型的一个作用是隐藏计算机硬件及其特性和差别,使硬件对于用户而言是透明的,即用户可以不关心数据类型是怎么实现的而可以使用它。定义数据类型的另一个作用是,用户能够使用数据类型定义的操作,方便地实现问题的求解。例如,用户可以使用Java定义在int型的加法操作完成两个整数的加法运算,而不用关心两个整数的加法在计算机中到底是如何实现的。这样不但加快了用户解决问题的速度,也使得用户可以在更高的层面上考虑问题。

与机器语言、汇编语言相比,高级语言的出现大大地简便了程序设计。但是要将解答问题的步骤从非形式的自然语言表达达到形式化的高级语言表达,仍然是一个复杂的过程,仍然要做很多繁杂琐碎的事情,因而仍然需要抽象。对于一个明确的问题,要解答这个问题,总是先选用该问题的一个数据模型。接着,弄清该问题所选用的数据模型在已知条件下的初始状态和要求的结果状态,以及隐含着的两个状态之间的关系。然后探索从数据模型的已知初始状态出发到达要求的结果状态

所必需的运算步骤。

在探索运算步骤时,首先应该考虑顶层的运算步骤,然后再考虑底层的运算步骤。所谓顶层的运算步骤是指定义在数据模型级上的运算步骤,或叫宏观运算。它们组成解答问题步骤的主干部分。其中涉及的数据是数据模型中的一个变量,暂时不关心它的数据结构;涉及的运算以数据模型中的数据变量作为运算对象,或作为运算结果,或二者兼而为之,简称为定义在数据模型上的运算。由于暂时不关心变量的数据结构,这些运算都带有抽象性质,不含运算的细节。所谓底层的运算步骤是指顶层抽象的运算的具体实现。它们依赖于数据模型的结构,依赖于数据模型结构的具体表示。因此,底层的运算步骤包括两部分:一是数据模型的具体表示;二是定义在该数据模型上的运算的具体实现。可以把它们理解为微观运算。于是,底层运算是顶层运算的细化,底层运算为顶层运算服务。为了将顶层算法与底层算法隔开,使二者在设计时不会互相牵制、互相影响,必须对二者的接口进行一次抽象;让底层只通过这个接口为顶层服务,顶层也只通过这个接口调用底层的运算。这个接口就是抽象数据类型。

抽象数据类型 (Abstract Data Type, ADT) 由一种数据模型和在该数据模型上的一组操作组成。

抽象数据类型包括定义和实现两个方面,其中定义是独立实现的。抽象数据类型的定义仅取决于它的逻辑特性,而与其在计算机内部的实现无关,即无论它的内部结构如何变化,只要它的逻辑特性不变,都不会影响到它的使用。其内部的变化(抽象数据类型实现的变化)只可能会对外部在使用它解决问题时的效率产生影响,因此我们的一个重要任务就是如何简单、高效地实现抽象数据类型。很明显,对于不同的运算组,为使组中所有运算的效率都尽可能地高,其相应的数据模型具体表示的选择将是不同的。在这个意义下,数据模型的具体表示又依赖于数据模型上定义的那些运算。特别是当不同运算的效率互相制约时,还必须事先将所有运算的相应使用频度排序,让所选择的数据模型的具体表示优先保证使用频度较高的运算有较高的效率。

应该看到,抽象数据类型的概念并不是全新的概念。抽象数据类型和数据类型在实质上是一个概念,只不过它是对数据类型的进一步抽象,这种抽象不仅限于各种不同的计算机处理器中已经实现的数据类型,还包括为解决更为复杂的问题而由用户自定义的复杂数据类型。例如高级语言都有的“整数”类型就是一种抽象数据类型,只不过高级语言中的整型引进实现了,并且实现的细节可能不同而已。人们没有意识到抽象数据类型的概念已经孕育在基本数据类型的概念之中,是因为人们已经习惯于在程序设计中使用基本数据类型和相关的运算,没有进一步深究。

抽象数据类型一方面使得使用它的人可以只关心它的逻辑特征,不需要了解它的实现方式。另一方面可以使人们更容易描述现实世界,可以在更高的层面上来考虑问题。例如可以使用树来描述行政区划,使用图来描述通信网络。

根据抽象数据类型的概念,对抽象数据类型进行定义就是约定抽象数据类型的名字,同时,约定在该类型上定义的一组运算的各个运算的名字,明确各个运算分别要有多少个参数,这些参数的含义和顺序以及运算的功能。一旦定义清楚,人们在使用时就可以像引用基本数据类型那样,十分简便地引用抽象数据类型;同时,抽象数据类型的实现就有了设计的依据和目标。抽象数据类型的使用和实现都与抽象数据类型的定义打交道,因此,只要严格按照定义,抽象数据类型的使用和实现就可以互相独立,互不影响,实现对它们的隔离,达到抽象的目的。

为此抽象数据类型可以使用一个三元组来表示:

$$ADT = (D, S, P)$$

其中, D 是数据对象, S 是 D 上的关系集, P 是加在 D 上的一组操作。在定义抽象数据类型时,可以使用以下格式:

```
ADT 抽象数据类型名 {  
    数据对象: <数据对象的定义 >
```

数据关系: <数据关系的定义>

基本操作: <基本操作的定义>

其中,数据对象和数据关系的定义用伪码描述,基本操作的定义格式为:

基本操作名(参数表)

初始条件: <初始条件描述>

操作结果: <操作结果描述>



1.2 算法和算法分析简介

1.2.1 算法

算法是计算机科学和技术中一个十分重要的概念。从下一章起,在讨论各种数据结构基本运算的同时,都将给出相应的算法。算法是执行特定计算的有穷过程。这个过程应有以下五个特点:

(1) 动态有穷:当执行一个算法时,不论是何种情况,在经过有限步骤后,这个算法一定要终止。

(2) 确定性:算法中的每条指令都必须是清楚的,指令无二义性。

(3) 输入:具有0个或0个以上由外界提供的量。

(4) 输出:产生1个或多个结果。

(5) 可行性:每条指令都充分基本,原则上可由人仅用笔和纸在有限的时间内也能完成。

由此可见,算法和程序是有区别的,即程序未必能满足动态有穷。例如,操作系统是个程序,这个程序永远不会终止。在本书中,只讨论满足动态有穷的程序,因此“算法”和“程序”是通用的。

1.2.2 算法的描述

一个算法可以用自然语言、数字语言或约定的符号来描述,也可以用计算机高级程序语言来描述,如C语言、Java语言或伪代码等。本书选用Java语言作为描述算法的工具。现简单说明Java语言的语法结构,具体如下:

① 常量和变量

1) 常量

在程序执行过程中其值不能改变的数据,称为常量。Java中的常量值是用文字串表示的,它区分为不同的类型,如整型常量123,实型常量1.23,字符常量'a',布尔常量true、false以及字符串常量"This is a constant string."。

Java中还可以通过关键字final将变量定义为常量。一个变量如果被定义为常量则这个常量的值不可再修改。例如:

```
final double PI=3.14; //正确
```

```
PI=3.1415926; //错误,试图修改一个常量的值
```

一般习惯将常量的标识符大写。

2) 变量

在程序执行过程中其值可以改变的数据,称为变量。每个变量都要有一个名称,这就是变量名。变量名由用户自己定义,但必须符合标识符的规定。