

21世纪高等学校规划教材 | 计算机科学与技术



数据结构

——Java语言描述（第2版）

朱战立 编著

清华大学出版社

21世纪高等计算机教材 | 计算机科学与技术



数据结构

——Java语言描述（第2版）

朱战立 编著

清华大学出版社
北京

内 容 简 介

本书内容符合 2009 年公布的研究生新的入学统考大纲要求,内容主要包括线性表、堆栈、队列、串、数组、向量、集合、矩阵、树、二叉树、图、排序、查找、哈希表以及递归,对于每一种类型的数据结构,都详细阐述了基本概念、各种不同的存储结构和不同存储结构上一些主要操作的实现算法,并给出了许多设计实例帮助读者理解。面向对象方法是目前软件设计的主流方法,本书用面向对象思想组织全部材料,采用 Java 语言作为算法描述语言。

本书既可作为大专院校计算机等专业的教科书,也可作为从事计算机应用的工程技术人员的自学参考。

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:Java 语言描述/朱战立编著.--2 版.--北京:清华大学出版社,2016

21 世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-42332-4

I. ①数… II. ①朱… III. ①数据结构—高等学校—教材 ②C 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(2015)第 287099 号

责任编辑:郑寅堃 薛 阳

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

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

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市中晟雅豪印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.25 彩 插:1 字 数:470 千字

版 次:2005 年 12 月第 1 版 2016 年 1 月第 2 版 印 次:2016 年 1 月第 1 次印刷

印 数:18501~20500

定 价:39.50 元

产品编号:061703-01

第2版前言

“数据结构”是计算机学科各专业中重要的专业基础课,也是其他计算机相关专业的一门必修课或选修课。本书讨论的内容主要包括线性表、堆栈、队列、串、数组、向量、集合、矩阵、树、二叉树、图、排序、查找、哈希表以及递归。其中,线性表、堆栈、队列、串、数组、向量、集合和矩阵属于线性结构,树和二叉树属于树状结构,图属于图结构,排序和查找是两个应用广泛的算法设计问题,哈希表是一种特殊类型的存储结构。树、二叉树和图的算法设计问题经常需要使用递归算法,考虑到有些读者对递归的概念和设计方法不太了解,因此专设一章讨论了递归。在讨论上述课程内容时,本书结合理论讨论,给出了十多个类的设计代码,并详细讨论了这些类的设计方法。另外,本书还给出了许多简单应用系统的完整程序设计实例。本书的所有程序代码都在 JCreator 4.50LE 中调试通过。

2005 年出版的本教材第 1 版,经过十年的使用,以及 2009 年公布的研究生新的入学统考大纲,作者认为有必要进行修订。

本次修订的内容主要包括以下几个。

(1) 根据新的研究生入学统考大纲,在树和二叉树一章中增加了等价问题一节,在图一章中增加了拓扑排序和关键路径两节,从而使本教材完全符合 2009 年公布的研究生统考大纲。

(2) 重写了原书中作者认为不够准确或完善的内容,改正了新发现的原书中的错误。

(3) 删除了原书第 0 章的 Java 语言基础一章。

本书具有如下特点。

(1) 内容丰富,难度适中,文字简洁准确,图文并茂。

(2) 本书的所有类设计和应用举例都调试通过。

(3) 习题全面,覆盖面广,并在附录 C 中给出了部分习题解答。

本书作者二十多年来一直从事数据结构课程的教学工作,曾编著过若干本采用不同算法描述语言的数据结构教材。本书是在经过长期使用的以前出版的教材基础上,参照新的研究生入学统考大纲,通过进一步修改、补充和完善完成的。

根据作者的经验,使用本教材授课约需 54~80 课时,其中包括约 10 课时的课内上机实习。

尽管作者在写作过程中非常认真和努力,但是错误和不足之处仍在所难免,敬请读者批评指正。作者的 E-mail 地址是: zhlzhu@xsyu.edu.cn。

朱战立

2015.8

第1版前言

“数据结构”是计算机专业以及其他一些和计算机技术关系密切专业的一门核心课程。“数据结构”课程主要讨论现实世界中数据的各种逻辑结构、在计算机中的存储结构以及各种算法的设计问题。“数据结构”课程的教学目的,是使学生掌握如何组织数据、如何存储数据和如何处理数据的基本概念和软件设计的基本方法,从而为进一步学习后续专业课程打下坚实的基础。

计算机软件开发方法是不断发展的,“数据结构”课程内容也应随着软件开发方法的不断发展而发展。目前,面向对象的软件分析和设计技术已发展成为软件开发的主流方法。因此,用面向对象的思想组织数据结构课程的材料,用面向对象的程序设计语言描述数据结构问题,就成为“数据结构”课程内容改革的必然。

用面向对象的程序设计语言描述算法时,首先涉及选用什么样的高级程序设计语言问题。Java语言是一种以对象为核心的纯粹面向对象的高级程序设计语言。和C++语言相比,Java语言取消了指针,也不允许多重继承。但用对象引用的方法不仅完全可以实现指针的功能,而且更具安全性;在单重继承基础上的接口方法,不仅可以实现多重继承,而且效率更高。另外,Java语言具有类库丰富、网络编程方便快捷、兼容多种应用环境等特点,是其他高级程序设计语言无法相比的。目前,国外大学的“数据结构”课程大部分都采用Java语言作为算法描述语言,国内大学虽然在这门课程的教学内容改革和教材建设方面起步较晚,但预计改革速度将会很快。因此,本书采用Java语言作为描述语言。

“数据结构”课程是一门理论和实践结合密切的课程。本书采用理论叙述简洁明了、实践应用例子丰富完整的方法编写,希望达到理论和实践密切结合的教学目的。

本书讨论的内容主要包括线性表、堆栈、队列、串、数组、向量、集合、矩阵、树、二叉树、图、排序、查找、哈希表以及递归。其中,线性表、堆栈、队列、串、数组、向量、集合和矩阵属于线性结构,树和二叉树属于树状结构,图属于图结构,排序和查找是两个应用广泛的算法设计问题,哈希表是一种特殊类型的存储结构。树、二叉树和图的算法设计问题经常需要使用递归算法,考虑到有些读者对递归的概念和设计方法不太了解,因此专设一章讨论了递归。

本书给出了十多个类的设计代码,并详细讨论了这些类的设计方法。另外,本书还给出了许多简单应用系统的完整程序设计实例。本书的所有程序代码都在JCreator 3.50LE中调试通过。

为了帮助教师使用本书作为教材,以及帮助读者学习理解本书讨论的内容,本书出版后,所有类的设计代码和所有应用实例的设计代码都可在清华大学出版社网站相关网页的资料下载区中得到。

习题的选择和设计是教材编写的一个重要方面。作者在习题选择和设计上考虑了习题的完整性、系统性和典型性,并把所有习题按类型分成基本概念题、复杂概念题、算法设计题和上机实习题4大类。基本概念题是每章基本内容的要点,读者学习完各章后,应以这部分

习题为提纲,透彻理解各章的基本内容以及相关的基本概念;复杂概念题和算法设计题既是教材内容的延伸,也是对读者学习理解程度的检查,读者要花时间认真完成这部分习题;“数据结构”课程要安排一定课时的上机实习,上机实习题是专为此目的设计的习题。习题中前面标有*号的,表示难度较大的习题。考虑到学生学习和教师教学的方便,作者在附录B中给出了上机实习报告书写规范和一个上机实习报告书写实例,在附录C中给出了部分典型习题的解答。

本书作者近年来一直从事“数据结构”课程的教学工作,讲授的“数据结构”课程被学校评为一类课程,编写的C++语言的数据结构教材被教育部定为普通高等教育“十五”国家级规划教材。本书是作者多年教学实践的结晶。

根据作者的经验,使用本教材授课约需54~70课时,其中包括10个左右课时的课内上机实习。前面标有*号的小节在课时少时可不讲授。

本书的最后成书得到了许多人的帮助。韩家新和作者多次讨论了书中的一些重要概念问题,杨锦锋、李红雷、杨明、杨萌、秦金祥协助作者做了许多程序设计和调试,以及电子图稿的制作工作。在本书完成之际,在此表示诚挚的感谢。另外,感谢在“数据结构”课程教学和教材编写方面富有经验的叶核亚不辞辛劳,认真审阅了全书,指出了书中的一些不妥之处,并提出了许多建设性的意见。

尽管作者在写作过程中非常认真和努力,但是疏漏和不足之处仍在所难免,敬请读者批评指正。

朱战立

2005.3

目 录

第 1 章 绪论	1
1.1 数据结构的基本概念	1
1.2 抽象数据类型	4
1.3 算法和算法的时间复杂度	5
1.3.1 算法.....	5
1.3.2 算法设计目标.....	6
1.3.3 算法的时间复杂度分析.....	7
1.4 算法的空间复杂度分析.....	11
1.5 Java 语言的工具包	11
习题	12
第 2 章 线性表	14
2.1 线性表概述.....	14
2.1.1 线性表的定义	14
2.1.2 线性表抽象数据类型	15
2.2 顺序表.....	16
2.2.1 顺序表存储结构	16
2.2.2 顺序表类	16
2.2.3 顺序表的效率分析	20
2.2.4 顺序表类应用举例	21
2.3 单链表.....	24
2.3.1 单链表的结构	24
2.3.2 结点类	26
2.3.3 单链表类	26
2.3.4 单链表的效率分析	30
2.3.5 顺序表和单链表的比较	30
2.3.6 单链表应用举例	30
2.4 循环单链表.....	31
2.5 双向链表.....	32
2.6 仿真链表.....	33
2.7 面向对象的软件设计方法.....	34
2.8 设计举例.....	34

2.8.1	顺序表算法设计举例	34
2.8.2	单链表算法设计举例	35
	习题	39
第3章	堆栈和队列	42
3.1	堆栈	42
3.1.1	堆栈的基本概念	42
3.1.2	堆栈抽象数据类型	44
3.1.3	顺序堆栈	44
3.1.4	链式堆栈	46
3.2	堆栈应用	48
3.2.1	括号匹配问题	48
3.2.2	表达式计算问题	50
3.3	队列	55
3.3.1	队列的基本概念	55
3.3.2	队列抽象数据类型	55
3.3.3	顺序队列	56
3.3.4	顺序循环队列类	58
3.3.5	链式队列	59
3.3.6	队列的应用	61
3.4	优先级队列	62
3.4.1	顺序优先级队列类	62
3.4.2	优先级队列的应用	64
	习题	66
第4章	串	68
4.1	串概述	68
4.1.1	串的基本概念	68
4.1.2	串的抽象数据类型	69
4.2	串的存储结构	70
4.3	串类	71
4.3.1	MyString 类	71
4.3.2	MyString 类的测试	75
4.3.3	MyStringBuffer 类	76
4.3.4	MyStringBuffer 类的测试	77
4.4	串的模式匹配算法	78
4.4.1	Brute-Force 算法	78
4.4.2	KMP 算法	80
4.4.3	Brute-Force 算法和 KMP 算法的运行效率比较	85

习题	87
第 5 章 数组、集合和矩阵	89
5.1 数组	89
5.1.1 数组的定义	89
5.1.2 数组的实现机制	90
5.1.3 数组抽象数据类型	91
5.1.4 Java 语言支持的数组功能	91
5.2 向量类	93
5.3 集合	96
5.3.1 集合的概念	96
5.3.2 集合抽象数据类型	97
5.3.3 集合类	97
5.4 矩阵类	101
5.5 特殊矩阵	104
5.5.1 特殊矩阵的压缩存储	104
5.5.2 n 阶对称矩阵类	106
5.6 稀疏矩阵	108
5.6.1 稀疏矩阵的压缩存储	108
5.6.2 数组结构的稀疏矩阵类	108
5.6.3 三元组链表	112
习题	113
第 6 章 递归算法	116
6.1 递归的概念	116
6.2 递归算法的执行过程	117
6.3 递归算法的设计方法	120
6.4 递归过程和运行时栈	122
6.5 递归算法的效率分析	124
6.6 递归算法到非递归算法的转换	126
6.7 设计举例	127
6.7.1 一般递归函数设计举例	127
6.7.2 回溯法及设计举例	129
习题	133
第 7 章 树和二叉树	135
7.1 树	135
7.1.1 树的定义	135
7.1.2 树的表示方法	137

7.1.3	树的抽象数据类型	137
7.1.4	树的存储结构	138
7.2	二叉树	140
7.2.1	二叉树的定义	140
7.2.2	二叉树抽象数据类型	141
7.2.3	二叉树的性质	142
7.2.4	二叉树的存储结构	143
7.3	以结点类为基础的二叉树设计	146
7.3.1	二叉树结点类	146
7.3.2	二叉树的遍历	147
7.3.3	二叉树遍历的应用	150
7.3.4	应用举例	151
7.3.5	非递归的二叉树遍历算法	154
7.4	二叉树类	155
7.5	二叉树的分步遍历	157
7.5.1	二叉树游标类	158
7.5.2	二叉树中序游标类	159
7.5.3	二叉树层序游标类	162
7.6	线索二叉树	164
7.7	哈夫曼树	166
7.7.1	哈夫曼树的基本概念	166
7.7.2	哈夫曼编码问题	167
7.7.3	哈夫曼编码的软件设计	168
7.8	等价问题	173
7.9	树与二叉树的转换	177
7.10	树的遍历	178
	习题	179
第8章	图	182
8.1	图概述	182
8.1.1	图的基本概念	182
8.1.2	图的抽象数据类型	185
8.2	图的存储结构	185
8.2.1	图的邻接矩阵存储结构	185
8.2.2	图的邻接表存储结构	187
8.3	邻接矩阵图类	188
8.4	图的遍历	191
8.4.1	图的深度和广度优先遍历算法	191
8.4.2	图的深度和广度优先遍历成员函数设计	193

8.5	最小生成树	196
8.5.1	最小生成树的基本概念	196
8.5.2	普里姆算法	197
8.5.3	克鲁斯卡尔算法	202
8.6	最短路径	203
8.6.1	最短路径的基本概念	203
8.6.2	从一个结点到其余各结点的最短路径	203
8.6.3	每对结点之间的最短路径	208
8.7	拓扑排序	208
8.8	关键路径	211
	习题	216
第 9 章	排序	218
9.1	排序的基本概念	218
9.2	插入排序	220
9.2.1	直接插入排序	220
9.2.2	希尔排序	222
9.3	选择排序	224
9.3.1	直接选择排序	224
9.3.2	堆排序	225
9.4	交换排序	230
9.4.1	冒泡排序	230
9.4.2	快速排序	231
9.5	归并排序	234
9.6	基数排序	236
9.7	各种排序算法的性能比较	239
	习题	239
第 10 章	查找	241
10.1	查找的基本概念	241
10.2	静态查找	242
10.2.1	在无序序列中查找	242
10.2.2	在有序序列中查找	243
10.2.3	索引	244
10.3	动态查找	247
10.3.1	二叉排序树	247
10.3.2	B_树	258
	习题	262

第 11 章 哈希表	264
11.1 哈希表的基本概念	264
11.1.1 哈希表的基本构造方法	265
11.1.2 建立哈希表的关键问题	266
11.2 哈希函数构造方法	267
11.3 哈希冲突解决方法	268
11.3.1 开放定址法	268
11.3.2 链表法	269
11.4 哈希表类设计	270
11.4.1 哈希表项类	270
11.4.2 哈希表类	271
11.4.3 应用程序设计举例	273
习题	274
附录 A Java 语言工具包实现的常用数据结构	275
附录 B 上机实习内容规范和实习报告范例	278
B.1 上机实习内容规范	278
B.2 上机实习报告范例——约瑟夫环问题	278
附录 C 部分习题解答	283
参考文献	296

计算机是对各种各样数据进行处理机器。要对数据进行处理,首先就要对数据进行组织。因此,在计算机中如何有效地组织数据和处理数据,既是计算机科学研究的基本内容,也是学生继续深入学习后续课程的基础。本章主要对“数据结构”课程学习中将遇到的基本概念进行概括性的叙述,这些内容将贯穿数据结构课程的整个学习过程中。

本章主要知识点

- 数据结构的基本概念,包括数据、数据元素、抽象数据元素、数据的逻辑结构、数据的存储结构、数据的操作;
- 类型、数据类型和抽象数据类型的概念;
- 算法的基本概念,包括算法的定义、性质和算法的设计目标;
- 算法的时间复杂度分析,包括 $O()$ 函数的定义,几种典型算法的时间复杂度分析方法。

1.1 数据结构的基本概念

1. 数据和数据元素

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抽象描述。

例如,今天天气最高温度为 4°C ,最低温度为 -4°C 就是关于今天天气情况的描述数据。又例如,班上甲同学叫张三,乙同学叫李四就是关于班上同学姓名的描述数据。

表示一个事物的一组数据称作一个**数据元素**;构成数据元素的数据称作该数据元素的数据项。

例如,要设计学生类,学生的数据元素一般应包括:学号、姓名、性别、年龄等数据。学生的学号、姓名、性别、年龄等数据就构成学生情况描述的数据项;包括学号、姓名、性别、年龄等数据项的一组数据就构成学生信息的一个数据元素。表 1-1 是一个有三个数据元素的学生信息表。

表 1-1 学生信息表

学号	姓名	性别	年龄
2000001	张三	男	20
2000002	李四	男	21
2000003	王五	女	22

在讨论数据结构时,关于数据元素、数据项的描述都使用某种高级程序设计语言来描述,本书采用 Java 语言描述。Java 语言表示数据元素时,是把数据元素表示成类的成员变量。类的成员变量的访问权限通常设计成私有访问权限。这样,学生类 Student 的成员变量部分就是:

```
public class Student
{
    private long number;
    private char name[10];
    private char sex[3];
    private int age;
    ...
};
```

有了 Student 类,就可以定义和创建 Student 类的对象。

上述学生情况数据元素是给出了具体数据元素类型的例子。但是,像数学一样,数据结构课程在讨论一种类型的数据结构问题时,通常说的是抽象意义上的数据元素,是没有实际含义的。没有实际含义的数据元素称作**抽象数据元素**。

2. 数据的逻辑结构

数据元素之间的相互联系方式称为**数据的逻辑结构**。

按照数据元素之间的相互联系方式,数据的逻辑结构可分为线性结构、树状结构和图结构。

线性结构的一般定义是:除第一个和最后一个数据元素外每个数据元素只有一个前驱数据元素和一个后继数据元素。表 1-1 中的数据元素就是一个线性结构的例子。又例如,对于数据元素 A、B、C、D,若数据的逻辑结构符合如图 1-1(a)所示的线性形式,则称这样的数据结构为线性结构。

树状结构的一般定义是:除根结点外每个数据元素只有一个前驱数据元素,可有零个或若干个后继数据元素。图 1-1(b)是一个树状结构的例子。对于数据元素 A、B、C、D、E、F、G,数据元素 A 没有前驱数据元素,有 B、C 两个后继数据元素;数据元素 B 的前驱数据元素为 A,后继数据元素为 D 和 E;数据元素 C 的前驱数据元素为 A,没有后继数据元素;如此等等。

图结构的一般定义是:每个数据元素可有零个或若干个前驱数据元素和零个或若干个后继数据元素。图 1-1(c)是一个图结构的例子。对于数据元素 A、B、C、D、E、F、G,数据元素 E 有 B 和 C 两个前驱数据元素,有 F 和 G 两个后继数据元素。

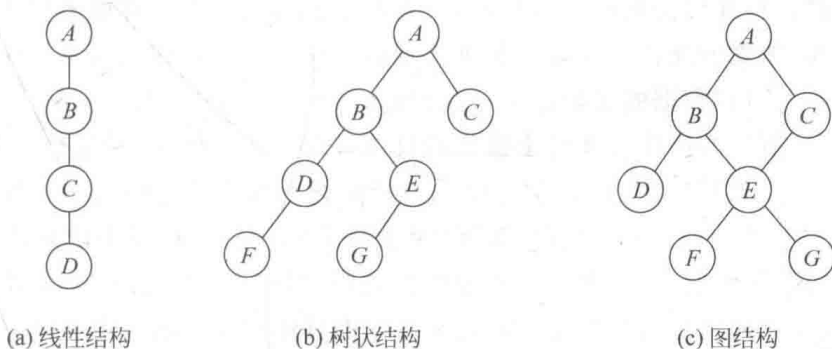


图 1-1 基本的数据逻辑结构形式

3. 数据的存储结构

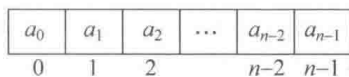
任何需要计算机进行管理和处理的数据元素都必须首先按某种方式存储在计算机中。数据元素在计算机中的存储方式称为**数据的存储结构**。

数据存储结构的基本形式有两种：一种是顺序存储结构，另一种是链式存储结构。

顺序存储结构是把数据元素存储在一块连续地址空间的内存中，程序设计方法是使用数组。顺序存储结构的特点是，逻辑上相邻的数据元素在物理上也相邻，数据间的逻辑关系表现在数据元素的存储位置关系上。图 1-2(a)是包含数据元素 a_0, a_1, \dots, a_{n-1} 的数组 a 内存存储结构示意图。其中， $0, 1, 2, \dots, n-1$ 为数据元素 a_0, a_1, \dots, a_{n-1} 在数组中的下标。

指针是指向一个内存单元的地址。一个数据元素和一个指针称为一个**结点**。

链式存储结构是用指针把相互直接关联的结点(即直接前驱结点或直接后继结点)链接起来。链式存储结构的特点是，逻辑上相邻的数据元素在物理上(即内存存储位置上)不一定相邻，数据间的逻辑关系表现在结点的链接关系上。图 1-2(b)是包含数据元素 a_0, a_1, \dots, a_{n-1} 的线性结构的链式存储结构示意图。其中，上一个结点到下一个结点的箭头表示上一个结点中指针域所指向的下一个结点对象。指针 $head$ 指向链式存储结构中的第一个结点对象。对象引用是一个对象的标识。Java 语言用 `表示一个对象的对象引用来实现指针。`



(a) 顺序存储结构



(b) 链式存储结构

图 1-2 基本存储结构形式

顺序存储结构和链式存储结构是两种最基本、最常用的存储结构。除此之外，利用顺序存储结构和链式存储结构进行组合，还可以有一些更复杂的存储结构。

4. 数据的操作

对一种类型的数据进行的某种方法的处理称作**数据的操作**，一种类型的数据所有的操作集合称作**数据的操作集合**。

“数据结构”课程在讨论数据的操作时,一般从功能和实现两个方面进行讨论。

在功能方面,数据的操作主要讨论某种类型数据应包含的操作和每种操作的逻辑功能。功能方面的操作一般和数据的逻辑结构一起讨论。

在实现方面,数据的操作主要讨论操作的具体实现方法。例如,若某软件要对表 1-1 的学生信息进行处理,在功能方面,对学生信息可能进行的操作有:插入一条数据元素,删除一条数据元素,列出所有数据元素的值,等等;在实现方面,表 1-1 的学生信息若采用图 1-2(a)的顺序存储结构存储数据元素,此时就要考虑在顺序存储结构下,插入一个数据元素、删除一个数据元素,以及列出所有数据元素的值等操作的具体实现方法。可见,操作的实现必须在数据的存储结构确定后才能进行。面向对象程序设计方法(包括 Java 语言)实现具体操作的方法是把操作设计为相应类的成员函数。

5. 数据结构课程讨论的内容

数据结构课程主要讨论线性表、堆栈、队列、串、数组、树、二叉树、图等典型的常用数据结构,在讨论这些典型的常用数据结构时,主要从它们的逻辑结构、存储结构和数据操作三个方面进行分析讨论。例如,在第 2 章讨论线性表时,2.1 节讨论线性表的抽象数据类型(即线性表的逻辑结构和逻辑结构意义下的操作功能),2.2 节讨论线性表的顺序存储结构和顺序存储结构下顺序表类的定义和各成员函数的具体实现方法,2.3 节讨论线性表的链式存储结构和单链表类的定义和各成员函数的具体实现方法。其他各章中对堆栈、队列、串、数组、树、二叉树、图等进行讨论的章节安排次序和第 2 章的类同。

1.2 抽象数据类型

类型是一组值的集合。

例如,整数类型就是具体计算机所能表示的整数数值的集合,通常整数类型的范围是 $-32\ 767 \sim 32\ 768$ 。又例如,布尔类型就是数值 true 和 false 组成的集合。

数据类型是指一个类型和定义在这个类型上的操作集合。

例如,当我们说计算机中的整数数据类型时,就不仅指计算机所能表示的整数数值的集合,而且指能对这个整数类型进行的加(+)、减(-)、乘(*)、除(/)和求模(%)操作。

在“数据结构”课程中,通常把在已有的数据类型基础上设计新的数据类型的过程称作数据结构设计,在这里,数据结构的含义和数据类型的含义相同。

抽象数据类型(Abstract Data Type, ADT)是指一个逻辑概念上的类型和这个类型上的操作集合。

从定义看,数据类型和抽象数据类型的定义基本相同。数据类型和抽象数据类型的不同之处仅仅在于:数据类型指的是高级程序设计语言支持的基本数据类型,而抽象数据类型指的是在基本数据类型支持下用户新设计的数据类型。“数据结构”课程主要讨论线性表、堆栈、队列、串、数组、树、二叉树、图等典型的常用数据结构,这些典型的常用数据结构就是一个个不同的抽象数据类型。

抽象数据类型是软件设计的初期使用的概念,也可以把它看作软件模块(教材中讨论的都是可重复使用的通用软件模块)开发时的用户需求。在用 Java 语言具体进行软件开发

时,抽象数据类型通常用接口或抽象类表示。

1.3 算法和算法的时间复杂度

1.3.1 算法

1. 算法的定义和算法的表示方法

算法是描述求解问题方法的操作步骤集合。

算法要用某种语言来描述。描述算法的语言主要有三种形式:文字形式、伪码形式和程序设计语言形式。文字形式是用中文或英文这样的文字来描述算法。伪码形式是用一种仿程序设计语言的语言(因这样的描述语言不是真正的程序设计语言,所以称作伪码)来描述算法。程序设计语言形式是用某种程序设计语言描述算法。用程序设计语言描述算法的优点是,这样的算法不用修改就可直接在计算机上运行。本书采用 Java 语言描述算法。

下面给出算法设计的两个例子。从这两个例子,读者可体味到用高级程序设计语言描述算法的基本方法。

例 1-1 设计一个把存储在数组 a 中的一组整数类型数据元素逆置后保存在数组 b 的算法。所谓逆置是指数据元素排列次序相反。

设计:为方便起见,这里把算法设计成独立于类的 static 属性的函数,下面类同。该算法设计问题要求有两个参数,一个是原数组 a ,另一个是逆置后的数组 b 。

算法设计如下。

```
static void reversel(int[] a, int[] b)
{
    int n = a.length;           //取数组 a 的长度值

    for(int i = 0; i < n; i++)
        b[i] = a[n - 1 - i];
}
```

该算法的实现方法图示见图 1-3。图中的 a_0 表示数组 a 中下标为 0 的数据元素, b_0 表示数组 b 中下标为 0 的数据元素,其余类推。

例 1-2 设计一个把数组 a 中的整数类型数据元素就地逆置的算法,所谓就地逆置是指数组 a 中的数据元素反序存放。

设计:这是一个不同于例 1-1 的算法设计问题。该算法要求只有一个参数,即数组 a 。

算法设计如下。

```
static void reverse2(int[] a)
{
    int n = a.length;
    int m = n/2;
```

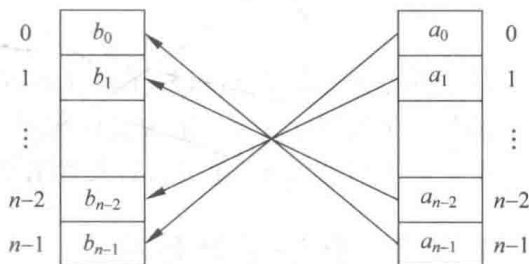


图 1-3 例 1-1 算法实现方法图示