新编高等院校计算机科学与技术规划教材

数据结构

彭 波 主编

SHUJU JIEGOU

本书配有电子教案

北京邮电大学出版社
www.buptpress.com

新编高等院校计算机科学与技术规划教材

数据结构

彭 波 主 编

北京邮电大学出版社
·北京·

内 容 简 介

本书是为“数据结构”课程编写的教材，也可以作为学习数据结构及算法的C程序设计的参考教材。本书系统地介绍了数据结构基础理论知识及算法设计方法，前半部分从抽象数据类型的角度讨论了各种基本类型的数据结构及其应用，主要包括线性表、栈和队列、串、数组和广义表、树和二叉树及图；后半部分主要讨论了查找和排序的各种实现方法及其综合比较；最后一章介绍了数据结构实验目的、步骤及内容；附录给出了全书习题参考答案。全书采用类C语言作为数据结构和算法的描述语言。随书配备电子教案。

本书在内容选取上符合人才培养目标的要求及教学规律和认知规律，在组织编排上体现“先理论、后应用、理论与应用相结合”的原则，并兼顾学科的广度和深度，力求适用面广。本书结构严谨、层次清楚，概念准确、深入浅出、描述清晰。

本书可以作为计算机类专业和信息类相关专业的本科或专科教材，也可以供从事计算机工程与应用工作的科技工作者参考。

图书在版编目(CIP)数据

数据结构/彭波主编. --北京：北京邮电大学出版社，2011.1

ISBN 978-7-5635-2503-4

Ⅰ. ①数… Ⅱ. ①彭… Ⅲ. ①数据结构 Ⅳ. ①TP311.12

中国版本图书馆CIP数据核字(2010)第264231号

书　　名：数据结构

主　　编：彭　波

责任编辑：张珊珊

出版发行：北京邮电大学出版社

社　　址：北京市海淀区西土城路10号(邮编：100876)

发 行 部：电话：010-62282185　传真：010-62283578

E-mail：publish@bupt.edu.cn

经　　销：各地新华书店

印　　刷：北京忠信诚胶印厂

开　　本：787 mm×1 092 mm　1/16

印　　张：18

字　　数：468千字

印　　数：1—3 000册

版　　次：2011年1月第1版　2011年1月第1次印刷

ISBN 978-7-5635-2503-4　　定　价：33.00元

前　言

数据结构是计算机、信息类及相关专业的专业基础课。它在整个课程体系中处于承上启下的核心地位：一方面扩展和深化在离散数学、程序设计语言等课程学到的基本技术和方法；另一方面为进一步学习操作系统、编译原理、数据库等专业课知识奠定坚实的理论与实践基础。本课程在教给学生数据结构设计和算法设计的同时，培养抽象思维能力、逻辑推理能力和形式化思维方法，增强分析问题、解决问题和总结问题的能力，更重要的是培养专业兴趣，树立创新意识。本书在内容选取上符合人才培养目标的要求及教学规律和认知规律，在组织编排上体现“先理论、后应用、理论与应用相结合”的原则，并兼顾学科的广度和深度，力求适用面广。

全书共分为 10 章。第 1 章综述了数据、数据结构和抽象数据类型等基本概念，以及算法描述与分析方法；第 2 章至第 7 章主要从抽象数据类型的角度分别讨论了线性表、栈和队列、串、数组和广义表、树和二叉树、图等基本类型的数据结构及其应用；第 8 章和第 9 章讨论了查找和排序，除了介绍各种实现方法之外，还着重从时间性能、应用场合及使用范围方面进行了分析和比较；第 10 章介绍了数据结构实验目的、步骤及内容。本书对数据结构众多知识点的来龙去脉做了详细解释和说明；每章后面配有难度各异的适量习题，并在附录中给出了习题参考答案，供读者理解知识及复习提高之用。随书配备电子教案。

全书采用类 C 语言描述数据结构和操作算法。它是 C 语言的一个精选子集，同时又采用了 C++对 C 非面向对象的增强功能，使本书对各种抽象数据类型的定义和数据结构相关操作算法的描述更加简明清晰、可读性更好，既不拘泥于 C 语言的细节，又容易转换成能够上机执行的 C 程序或 C++程序。

从课程性质上讲，“数据结构”是高等院校计算机科学、信息科学及相关专业考试计划中一门专业基础课，其教学要求是：学会分析研究计算机加工的数据结构的特性，以便为应用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空分析技术。从课程学习上讲，“数据结构”的学习是复杂程序设计的训练过程，其教学目的是：着眼于原理与应用的结合，在深化理解和灵活掌握教学内容的基础上，学会把知识用于解决实际问题，书写出符合软件工程规范的文件，编写出结构清晰及正确易读的程序代码。可以说，“数据结构”比“高级程序设计语言”课程有着更高的要求，它注重培养数据抽象能力。

在本书的构思与编写过程中，得到了孙一林、邱李华、张伟娜、徐林、韩振华、王平等同志的帮助。在算法的实现与调试过程中，得到了方孝鹏、岳乐、杨志军、孙苗、王微等研究生的帮助，在此表示感谢。本书可以作为计算机类专业和信息类相关专业的本科或专科教材，也可以供从事计算机工程与应用工作的科技工作者参考。本书结构严谨、层次清楚、概念准确、深入浅出、通俗易懂、便于自学。由于作者水平有限，教材中不当之处敬请读者提出批评和建议，作者电子邮件地址：pengbo_cau@126.com。

编　者

目　录

第1章 绪 论

【本章学习要点】

- 数据结构的基本概念，以及学习数据结构的重要意义
- 算法时间及空间复杂度分析

【本章学习目标】

- 了解数据结构的基本概念
- 理解逻辑结构、存储结构及数据运算的相互关系
- 掌握算法时间及空间复杂度分析方法

使用计算机求解任何问题都离不开程序设计，而程序设计的实质就是数据表示和数据处理。数据要能被计算机处理，首先必须能够存储在计算机的内存中，这项任务称为数据表示，数据表示的核心任务是数据结构的设计；一个实际问题的求解必须满足各项处理的要求，这项任务称为数据处理，数据处理的核心任务是算法设计。因此，数据结构课程主要讨论数据表示和数据处理的基本问题。

1.1 数据结构的重要意义

数据结构起源于程序设计。随着计算机应用领域的扩大和软件及硬件的飞速发展，电子计算机的应用已远远超出了科学和工程计算的范围，被广泛地应用于情报检索、信息管理、系统工程，乃至人类社会活动的一切领域。与此同时，计算机的处理对象也从简单的纯数值性数据发展到非数值性和具有一定结构的数据，比如文本、图形、图像、音频、视频及动画等，处理的数据量也越来越大，这就给程序设计带来一个问题：即应该如何组织待处理的数据以及数据之间的关系(结构)。

1.1.1 计算机处理问题分类

计算机处理问题可以分为数值计算问题和非数值性问题。

1. 数值计算问题

众所周知，20 世纪 40 年代，电子计算机问世的直接原因是解决弹道学的计算问题。早期的电子计算机，其应用范围只局限于科学和工程计算，处理对象是纯数值性数据，通常人们把这类问题称为数值计算。

例如，线性方程求解问题涉及的运算对象是简单的整型、实型或布尔型数据。程序设计者

的主要精力集中于程序设计的技巧，不需要重视数据结构。

2. 非数值性问题

根据统计，当今处理非数值性问题占用了90%以上的机器时间，这类问题涉及到的数据结构更为复杂，数据元素之间的相互关系一般无法用数学方程式加以描述。因此，解决此类问题的关键已经不再是分析数学和计算方法，而是要设计出合适的数据结构，才能有效地解决问题。

1.1.2 非数值性问题求解

1968年，克努思(Donald. E. Knuth)教授开创了数据结构的最初体系，他所著的《计算机程序设计艺术》第一卷《基本算法》是第一本较系统地阐述数据逻辑结构和存储结构及其操作的著作。1976年，著名的瑞士计算机科学家沃思(N. Wirth)教授提出：数据结构＋算法＝程序设计。数据结构是指数据的逻辑结构和存储结构，算法是对数据运算的描述。程序设计的实质是对实际问题选择一种好的数据结构，加之设计一个好的算法，而好的算法在很大程度上取决于描述实际问题的数据结构。

例 1-1 电话号码查询问题。

编一个查询某个城市或单位的私人电话号码程序。要求对任意给出的一个姓名，如果该人有电话号码，则迅速找到其电话号码；否则指出该人没有电话号码。

(1) 要求解此问题，首先需要构造一张电话号码登记表，表中每个数据元素包括两个数据项：姓名和电话号码。

(2) 要写出好的查找算法，取决于这张表的结构及存储方式。最简单的方式是将表中数据顺序地存储在计算机中，查找时从头开始依次查对姓名，直到找出正确的姓名或是找遍整个表均没有找到为止。这种查找算法对于一个不大的单位或许是可行的，但对一个有成千上万私人电话的城市就不实用了。如果这张表是按照姓氏排列的，则可以另外造一张姓氏索引表，采用如图1-1所示的存储结构，查找时首先在索引表中查对姓氏，然后根据索引表中的地址到电话号码登记表中核查姓名，这样查找登记表时就不需要查找其他姓氏的名字了。因此，在这种新结构上产生的查找算法就更为有效。

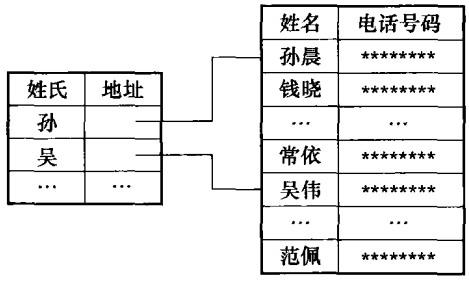


图 1-1 电话号码查询问题的索引存储

例 1-2 田径比赛时间安排问题。

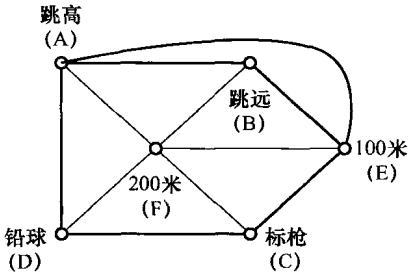
假设某学校的田径选拔赛共设6个项目比赛，即跳高(A)、跳远(B)、标枪(C)、铅球(D)、100米短跑(E)和200米短跑(F)，规定每个选手至多参加3个项目的比赛。现有五名选手报名比赛，选手所选择的项目如图1-2(a)所示。要求设计一个竞赛日程安排表，使得在尽可能短的时间内安排完比赛。

(1) 为了能够较好地解决这个问题，首先应该选择一个合适的数据结构来表示它。表示该问题的数据结构模型如图 1-2(b)所示(图中顶点代表竞赛项目，在所有的两个不能同时进行比赛的项目之间连上一条边)。显然同一个选手选择的几个项目是不能在同一时间内比赛的，因此该选手选择的项目中应该两两有边相连。

(2) 竞赛项目的时间安排问题可以抽象为对图 1-2(b)的数据结构模型图进行“着色”操作：即用尽可能少的颜色去给图中每个顶点着色，使得任意两个有边连接的相邻顶点着上不同的颜色。每一种颜色表示一个比赛时间，着上同一种颜色的顶点是可以安排在同一时间内竞赛的项目。由此可得：只要安排 4 个不同的时间竞赛即可。时间 1 内可以比赛跳高(A)和标枪(C)，时间 2 内可以比赛跳远(B)和铅球(D)，时间 3 和时间 4 内分别比赛 100 米(E)和 200 米(F)。

姓名	项目1	项目2	项目3
孙晨	跳高	跳远	100米
钱晓	标枪	铅球	—
常依	标枪	100米	200米
吴伟	铅球	200米	跳高
范佩	跳远	200米	—

(a) 参赛选手比塞项目表



(b) 安排竞赛项目的数据结构模型

图 1-2 田径比赛时间安排

对于非数值性问题的求解，首先是选取合适的数据结构表示该问题，然后才能写出有效的算法。

数据结构是计算机软件和计算机应用专业的核心课程之一，在众多的计算机系统软件和应用软件中都要用到各种数据结构。因此，仅掌握几种计算机语言是难以应付众多复杂的实际应用课题的。要想有效地使用计算机，就必须学习数据结构的有关知识。

1.2 数据结构的相关概念

在本节中，我们先对以后各章节中反复出现的数据结构相关概念赋以确定的含义，以便在后面的学习中与读者取得“共同的语言”。

1.2.1 数据概念

1. 数据

数据(Data)是信息的载体，在计算机科学中指所有能输入到计算机中并能被计算机程序识别、存储和处理的符号集合。数据是计算机程序加工的“原料”。

例如，一个利用数值分析方法求解代数方程的程序，其处理对象是整数和实数；一个编译程序或文字处理程序，其处理对象是字符串；一个影视作品制作程序，其处理对象是声音、图像、动画及视频等。对于计算机科学而言，数据的含义极为广泛，比如图像、声音、动画、视频等都可以通过编码而归之于数据的范畴。

2. 数据元素

数据元素(Data Element)是数据基本单位,也称元素(Element)、结点(Node)、顶点(Vertex)、记录(Record),在计算机程序中常作为一个整体进行考虑和处理。一个数据元素可以是不可分割的原子,称为原子项;也可以由若干个数据项组成,称为组合项。数据项是数据不可分割的最小单位。

例如,包括书名、作者名、分类号、出版单位及出版时间在内的一条书目信息在计算机图书管理程序中被作为一个数据元素;而书名、分类号等被称作数据项。数据元素具有广泛的含义,一般来说,能独立、完整地描述问题世界的一切实体都是数据元素。

3. 数据对象

数据对象(Data Object)是性质相同的数据元素的集合,是数据的一个子集。

例如,整数数据对象的集合可以表示为 $N=\{0,\pm1,\pm2,\cdots\}$;大写字母字符数据对象的集合可以表示为 $C=\{'A','B',\cdots,'Z'\}$;银行业务处理系统中的数据对象是全体储户及全体贷款客户资料;电话号码查询系统中的数据对象是全体电话用户。

4. 数据关系

数据关系(Data Relation)反映了数据对象中数据元素所固有的一种结构。在数据处理领域,通常把数据之间的这种固有关系简单地用前驱和后继来描述。

例如,在编写家庭族谱时,数据对象是家庭中的所有成员,对家族中某成员的描述就是一个数据元素,各数据元素之间存在着血缘关系;父亲是儿子的前驱,儿子是父亲的后继,小孩子没有后继。一张按照名次排列的成绩表,数据对象是全班同学,对某个同学属性(姓名、成绩等)的描述就是一个数据元素,各数据元素之间存在着名次关系;第一名没有前驱,其后继是第二名,第二名的前驱是其一名,其后继是第三名。

1.2.2 结构概念

1. 数据结构

数据结构(Data Structure)是相互之间存在一种或多种特定关系(或结构)的数据元素集合。它研究的内容包括:数据的逻辑结构、数据的物理(存储)结构、数据的运算。

2. 逻辑结构

数据的逻辑结构(Logical Structure)是从数据元素之间的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。它可以被看成是从具体问题抽象出来的数学模型,是对操作对象的一种数学描述。

根据数据元素之间逻辑关系的不同,数据的逻辑结构分为 4 类,如图 1-3 所示。

(1) 集合(Set)

结构中的数据元素之间除了"同属于一个集合"外别无其他的关系,即只有数据元素而无任何关系。

(2) 线性结构(Linear Structure)

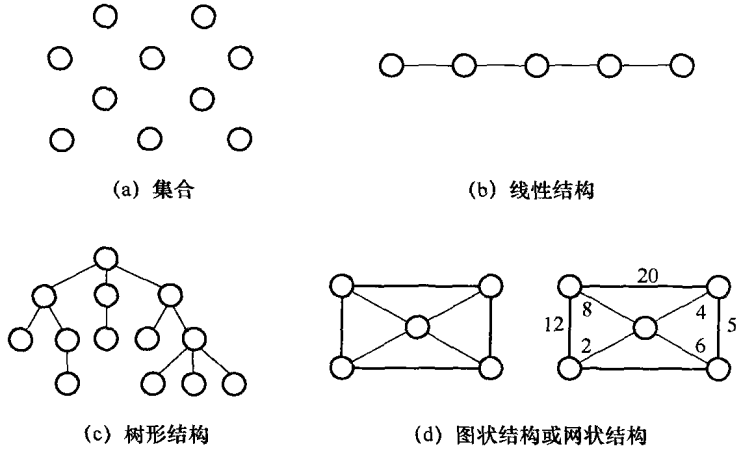
结构中的数据元素之间存在着一对一的关系,即除了第一个数据元素无前驱、最后一个数据元素无后继外,其他相邻数据元素均具有唯一的前驱和后继。

(3) 树形结构(Tree Structure)

结构中的数据元素之间存在着一对多的关系,即除了一个根数据元素外,其他各元素均有唯一的前驱,所有数据元素都可以有多个后继。

(4) 图状结构或网状结构(Graph or Net Structure)

结构中的数据元素之间存在着多对多的关系,即所有数据元素都可以有多个前驱或多个后继。



(a) 集合　(b) 线性结构　(c) 树形结构　(d) 图状结构或网状结构

图 1-3　数据的逻辑结构

由于"集合"是数据元素之间关系极为松散的一种结构,是数据结构的一种特例,因此也可以用其他结构来表示。

例 1-3　描述一周七天数据的逻辑结构。

数据元素:D = {Sun,Mon,Tue,Wed,Thu,Fri,Sat}

逻辑关系:R = {< Sun,Mon >,< Mon,Tue >,< Tue,Wed >,< Wed,Thu >,< Thu,Fri >,< Fri,Sat >}

该数据的逻辑结构如图 1-4 所示。

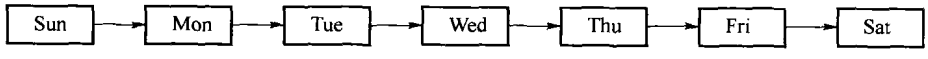


图 1-4　一周七天数据的逻辑结构

3. 物理结构

数据的物理结构(Physical Structure),又称为存储结构(Storage Structure),是数据及其逻辑结构在计算机中的表示,换言之,存储结构除了存储数据元素之外,必须隐式或显式地存储数据元素之间的逻辑关系。

数据结构在计算机中有两种不同的表示方法:顺序表示和非顺序表示。由此得出常用的两种不同的存储结构:顺序存储结构和链式存储结构。

(1) 顺序存储结构(Sequential Storage Structure)

用数据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。

例 1-4　假设一组数据的逻辑结构如下:

数据元素:D = {A,B,C,D,E}

逻辑关系:R = {< A,B >,< B,C >,< C,D >,< D,E >}

其中,每个数据元素在内存占 1 个字节,其顺序存储结构如图 1-5 所示。

	A	B	C	D	E
地址	2000	2001	2002	2003	2004

图 1-5　一组数据元素的顺序存储结构

(2) 链式存储结构(Linked Storage Structure)

在每一个数据元素中增加一个存放地址的指针，用此指针来表示数据元素之间的逻辑关系。

例 1-5 假设一组数据的逻辑结构如下：

数据元素：D = {80,75,90,85,70}

逻辑关系：R = {< 90,85 >,< 85,80 >,< 80,75 >,< 75,70 >}

其中，每个数据元素在内存占 2 个字节，其链式存储结构如图 1-6 所示。

地址	数据	指针
2000	80	2002
2002	75	2008
2004	90	2006
2006	85	2000
2008	70	^

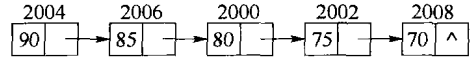
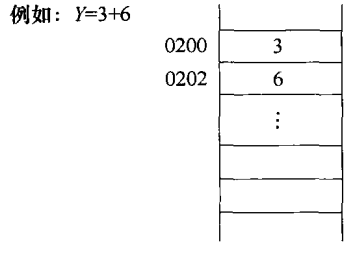


图 1-6 一组数据元素的链式存储结构

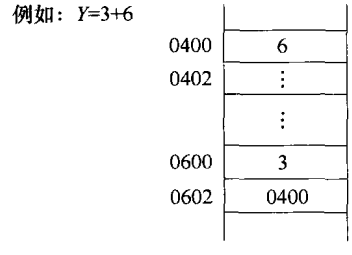
通常，在程序设计语言中，顺序存储结构借助于数组描述，链式存储结构借助于指针描述。在实际程序设计过程中，针对一种数据的逻辑结构，到底选择哪种存储结构会影响到具体算法的设计。也就是说，在设计具体算法之前，必须先确定存储结构。

4. 存储方法

数据的存储结构可以使用以下 4 种基本存储方法得到，如图 1-7 所示。



(a) 顺序存储方法



(b) 链式存储方法

数据元素信息

地址	学号	姓名	性别	年龄
101	03	丁一	男	18
109	10	王二	男	19
117	07	张三	女	17
125	05	李四	女	17
133	06	周五	男	17

附加索引

关键字	地址
03	101
05	125
06	133
07	117
10	109

(c) 索引存储方法

例如，数据元素关键字序列为{29，17，60，38}，存储地址=关键字 MOD 11。

0	1	2	3	4	5	6	7	8	9	10
					60	17	29	38		

(d) 哈希存储方法

图 1-7 数据的存储结构

(1) 顺序存储方法

该方法把逻辑上相邻的数据元素存储在物理位置上相邻的存储单元里，数据元素之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，主要应用于线性的数据结构。非线性的数据结构也可以通过某种线性化的方法实现顺序存储。

(2) 链式存储方法

该方法不要求逻辑上相邻的数据元素在物理位置上亦相邻，数据元素之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。

(3) 索引存储方法

该方法通常在存储数据元素信息的同时，还建立附加的索引表。索引表由若干索引项组成。如果每个数据元素在索引表中都有一个索引项，则该索引表称之为稠密索引(Dense Index)。如果一组数据元素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(Key)是能唯一标识一个数据元素的那些数据项。稠密索引中索引项的地址指示数据元素所在的存储位置；稀疏索引中索引项的地址指示一组数据元素的起始存储位置。

(4) 哈希存储方法

该方法的基本思想是根据数据元素的关键字直接计算出该数据元素的存储地址。

4 种基本存储方法既可以单独使用，也可以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；采用不同的存储结构，其数据处理效率往往不同。选择何种存储结构来表示相应的逻辑结构，视具体要求而定，主要考虑运算方便及算法的时空要求。

5. 数据运算

数据运算(Data Operation)是定义在数据的逻辑结构上的，每种逻辑结构都有一个运算集合。最常用的查找、插入、删除、更新、排序等运算实际上只是在抽象的数据上所施加的一系列抽象的操作。所谓抽象的操作，是指我们只知道这些操作是“做什么”，而不需要考虑“如何做”。只有确定了存储结构及编程语言之后，才考虑如何具体实现这些运算。

数据的逻辑结构、数据的存储结构及数据的运算这三方面是一个整体。孤立地去理解一个方面，而不注意它们之间的联系是不可取的。

存储结构是数据结构不可缺少的一个方面，同一逻辑结构的不同存储结构可冠以不同的数据结构名称来标识。例如，线性表是一种逻辑结构，如果采用顺序存储方法，则可以称其为顺序表；如果采用链式存储方法，则可以称其为链表；如果采用散列存储方法，则可以称为散列表。

数据运算也是数据结构不可分割的一个方面。在给定了数据的逻辑结构和存储结构之后，按定义的运算集合及其运算性质不同，也可能导致完全不同的数据结构。例如，如果对线性表上的插入运算、删除运算限制在表的一端进行，则该线性表称为栈；如果对插入运算限制在表的一端进行，而删除运算限制在表的另一端进行，则该线性表称为队列。更进一步地说，如果线性表采用顺序表或链表作为存储结构，则对插入运算和删除运算做了上述限制之后，可以分别得到顺序栈或链栈、顺序队列或链队列。

1.2.3 类型概念

1. 数据类型

数据类型(Data Type)是计算机程序中的数据对象以及定义在这个数据对象集合上的一组操作的总称。通常数据类型可以看作是程序设计语言中已实现的数据结构。

例如,C 语言中的"整型类型"是区间[MININT, MAXINT]上的整数,在这个集合上可以进行加、减、乘、除和取模等操作。

按照数据对象集合中的数据元素是否可分解,数据类型可以分为以下两类。

(1) 原子类型:数据元素不可分解,通常由语言直接提供。比如 C 语言的整型、字符型等标准类型,以及指针等简单的导出类型。

(2) 结构类型:数据元素可分解为若干个成分(或称为分量),是用户借助于语言提供的描述机制自己定义的类型。它通常是由标准类型派生的,因此它也是一种导出类型。比如 C 语言的数组、结构体等类型。

2. 抽象数据类型

抽象(Abstract)就是抽出问题本质的特征而忽略非本质的细节,是对具体事物的一个概括。比如,地图是对它所描述地域的一种抽象,中国人是对所有具有中国国籍的中国公民的一种抽象。

抽象数据类型(Abstract Data Type,ADT)是指一个数据模型以及定义在该模型上的一组操作。ADT 的定义仅取决于它的一组逻辑特性,而与其在计算机的内部如何表示和实现无关,即不论其内部结构如何变化,只要它的数学特性不变,都不影响其外部的使用。ADT 可以理解为对数据类型的进一步抽象,数据类型和 ADT 的区别仅在于: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 ADT 指的是自定义的数据类型。

随着计算机科学的不断发展,特别是面向对象的程序设计语言的研究和发展,提出了抽象数据类型的概念。为了提高软件复用率,在近代程序设计方法学中指出,一个软件系统框架应该建立在数据之上,而不是像传统软件设计方法学那样,将一个软件系统框架建立在操作之上。也就是说,在构成软件系统每个相对独立的模块中,定义一组数据和施于这些数据之上的一组操作,并在模块内部给出它们的表示和实现细节,在模块外部使用的只是抽象的数据和抽象的操作。显然,所定义数据类型的抽象层次越高,含有该抽象数据类型软件模块的复用率也就越高。

一个 ADT 的定义不涉及它的实现细节,在形式上可繁可简,本书对 ADT 的定义采用如下格式描述:

```
ADT 抽象数据类型名 {
    数据对象(Data):<数据对象的定义>
    逻辑关系(Relation):<逻辑关系的定义>
    基本操作(Operation):
        基本操作名 1:
            初始条件:<初始条件描述>
            操作结果:<操作结果描述>
        基本操作名 2:
            初始条件:<描述操作执行之前数据结构和参数应该满足的条件>
            操作结果:<说明操作正常完成之后数据结构的变化状况和应该返回的结果>
        ...
} ADT 抽象数据类型名
```

其中,基本操作有两种参数:赋值参数只为操作提供输入值;引用参数以 & 打头,除了可