高校计算机教学系列教材

数据结构教程

迟乐军
王艳春
于晓敏
袁　琪　编著

北京航空航天大学出版社
http://www.buaapress.com.cn

高等学校计算机教学系列教材

数据结构教程

迟乐军　于晓敏
王艳春　袁　琪　编

北京航空航天大学出版社
http://www.buaapress.com.cn

内容简介

计算机在各个领域的应用过程中，都会涉及到数据的组织与程序的编排等问题，都会用到各种各样的数据结构。选择最合适的数据结构和存储表示方法，以及编制相应的实现算法的方法是计算机工作者不可缺少的知识。本书全面、系统地介绍各种类型的、最常用的数据结构及其查找、排序的各种方法。对每一种数据结构除了阐述各种数据结构所涉及的逻辑关系之外，还讨论它们在计算机中的存储表示方法以及在这些数据结构上的运算（操作）和实际的执行算法，并对算法的效率进行简要的分析和讨论。概念清楚，内容丰富，详略得当，既便于开展各种层次的教学，又便于读者自学。

本书可以作为大专院校计算机及相关专业的教材，也可以供从事计算机工程与应用的科技工作者参考。

图书在版编目(CIP)数据

数据结构教程/迟乐军等编. —北京：北京航空航天大学出版社，2003.4

ISBN 7-81077-293-7

Ⅰ.数… Ⅱ.迟… Ⅲ.数据结构—教材

Ⅳ.TP311.12

中国版本图书馆 CIP 数据核字(2003)第 010330 号

数据结构教程

迟乐军 于晓敏 王艳春 袁 琪 编

责任编辑 许传安

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话：82317024 传真：82328026

http://www.buaapress.com.cn

E-mail:bhpress@263.net

北京市松源印刷有限公司印装 各地书店经销

*

开本：787×1092 1/16 印张：15.25 字数：384 千字

2003 年 4 月第 1 版 2003 年 4 月第 1 次印刷 印数：5000 册

ISBN 7-81077-293-7 定价：23.00 元

总前言

科教兴国，教育先行，在全国上下已形成共识。在教育改革过程中，出现了多渠道、多形式、多层次办学的局面。同时，政府逐年加大教育的投入力度。教育发展了，才能有效地提高全民族的文化、科学素质，使我们中华民族屹立于世界民族之林。

计算机科学与技术的发展日新月异，其应用领域迅速扩展，几乎无处不在。社会发展的需求，促使计算机教育生气蓬勃。从普通高校的系统性教学，到远距离的电视、网上教学；从全面讲述，到不同应用领域的、星罗棋布的培训班；从公办的到民办的；从纸介教材到电子教材等等，可以说计算机教学异彩纷呈。要进行教学，就必须有教材。

面对我们这么大的国家和教学形势，在保证国家教学基本要求的前提下，应当提倡教材多样化，才能满足各教学单位的需求，使他们形成各自的办学风格和特色。为此，我们组织北京工业大学、北京航空航天大学、北京理工大学、南开大学、天津工业大学等高校的有丰富教学经验的教师编写了计算机教学的系列教材，将陆续与师生见面。

系列教材包括以下各项。

（一）基础理论：离散数学。

（二）技术基础：电路基础与模拟电子技术；数字逻辑基础；计算机组成与体系结构；计算机语言（拼盘，选择使用），包括 C++程序设计基础、Visual Basic 程序设计基础、Matlab 程序设计基础、Java 程序设计基础、Delphi 语言基础、汇编语言基础等；数据结构；计算机操作系统基础；计算方法基础；微机与接口技术；数据库技术基础等。

（三）应用基础：计算机控制技术；网络技术；软件工程；多媒体技术等。

（四）技术基础扩展：编译原理与编译构造；知识工程——网络计算机环境下的知识处理。

（五）应用基础扩展：计算机辅助设计；单片机实用基础；图像处理基础；传感器与测试技术；计算机外设与接口技术。

本系列教材主要是针对计算机教学编写的，供普通高校、社会民办大学、高等职业学校、业余大学等计算机本科和专科使用。其中一部分教材也适合非计算机专业本科教学使用。在这部分教材的内容简介或前言中对使用范围均作了说明。

本系列教材在编写时，注重以下几点：（1）面对计算机科学与技术动态发展的现实，在内容上应具有前瞻性；（2）面对学以致用，既有系统的基础知识，又有应用价值的实用性；（3）具有科学性、严谨性。另外，力求排版紧凑，使有限的版面具有最大的信息量，以使读者得到实惠。

能否实现这些愿望，只有师生在教学实践中评价。我们期望得到师生的批评和指正。

高校计算机教学系列教材编委会

高校计算机系列教材编委会成员

主　　任：赵沁平

副 主 任（常务）：陈炳和

顾　　问：姜中凡

委　　员（以姓氏笔划为序）：

吕景瑜（北工大教授）

乔少杰（社长，副教授）

姜中凡（北航教授，教育部工科计算机基础教学指导委员会副主任、中专计算机教学指导委员会顾问）

苏开娜（北工大教授）

陈炳和（北工大教授）

张鸿宾（北工大博导）

郑玉明（北工大副教授）

金茂忠（北航博导）

赵沁平（北航博导，国务院学位办主任）

前　言

计算机在各个领域的应用过程中，都会涉及到数据的组织与程序的编排等问题，都会用到各种各样的数据结构，特别是针对各种特殊数据的表示，就更需要学会分析和研究计算机加工对象的特性，选择最合适的数据组织结构及其存储表示方法，以及编制相应实现算法的方法。这是计算机工作者不可缺少的知识。因此"数据结构"这门课一直是高等院校计算机专业教学中的一门主要技术基础课程。在我国当前的计算机专业教学计划中，它是主干课程之一。

本书介绍了各种最常用的数据结构，阐述各种数据结构所内涵的逻辑关系，讨论它们在计算机中的存储表示，以及在这些数据结构上的运算（操作）和实际的执行算法，并对算法的效率进行简要的分析和讨论。

全书共 10 章。前面 4 章主要介绍数据结构的基本知识和几种基本的数据结构，即线性表、栈和队列、串和数组。它们均属于线性数据结构。第 5 章和第 6 章叙述非线性数据结构。它们是树、图和广义表。第 7、8 两章分别介绍数据处理中广泛使用的技术——排序和查找。第 9 章讨论外存储器上的数据结构——文件。第 10 章给出了学生实践所需的实习与设计课题及其分析问题和解决问题的方法和思路。最后的附录概要地叙述 C 语言，便于读者理解书中用 C 语言描述的算法。

本书是在作者多年教学经验的基础上编写而成的。在编写过程当中的编写思路既注重基本原理的介绍，又重视实践能力的培养。书中还配有大量的例题和习题，供学生练习，加深对知识点的理解程度。本书讲解详细，通俗易懂，详略得当。书中算法采用类 C 语言进行描述，且所给的程序均已在计算机上运行调试，部分程序还做了较详细的注解，以便读者了解算法的实质和基本思想。书中每一章均有习题，可以检验读者的学习效果和培养程序设计的能力。

本书既可作为计算机专业的教材，亦可供从事计算机应用的工程技术人员参考，特别适合于那些使用 C 语言编程的计算机应用人员。使用本书作为本科生教材时，其内容可以讲授一个学期；若使用它作为专科生教材时，可以删去带 * 的章节；若使用它作为成人教育（面授或函授）的教材时，建议酌情再精简有关内容。

本书初稿曾经过我们的教学实践的检验，教学效果较好，学生反映本书好学易懂。本书的第 1 章、第 9 章和第 10 章由迟乐军编写，第 2 章、第 3 章和第 4 章由王艳春编写，第 5 章和第 6 章由于晓敏编写，第 7 章和第 8 章由袁琪编写，附录及部分程序的编写由张　旭完成。赵　颖、于晓坤、孙海龙、李国军和孙桂江等也参与了程序的调试和部分文档的编辑与整理工作。初稿完成后，承蒙我校计算机系软件教研室的同仁详细审阅，提出了许多宝贵意见；其他老师亦提供了许多有益的建议，作者谨此一并致以诚挚的谢意。

由于作者水平有限，殷切希望广大读者批评指正。

迟乐军　于晓敏　王艳春　袁　琪

2002 年 10 月

目　录

第1章 绪 论

自从1946年第一台计算机问世以来，信息产业的飞速发展已远远超出人们对它的预料。在某些现代化的生产线上，甚至几秒钟就能生产出一台微型计算机，产量猛增，产品价格却大幅度下降。计算机的成本低廉就使得它的应用范围迅速扩展。如今，随着计算机的应用已深入到人类社会的生产、生活等各个领域，计算机的应用已不再局限于科学计算，而更多地用于生产过程控制、信息管理及音频、视频等特殊数据处理等非数值计算的处理工作。与此相对应，计算机所加工处理的对象也由数据和相应的结构构成。这就是“数据结构”这门学科形成和发展的背景。

1.1 什么是数据结构

一般来说，当我们用计算机解决一个具体问题时，大致都需要经过下面几个步骤：首先要从具体问题中抽象出一个适当的数学模型(或数学公式)，然后设计一个描述此数学模型的算法，最后利用合适的程序设计语言来编写程序，进行测试、调整，直至最终得到满意的解答。抽象数学模型的过程实质上是分析问题，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的过程。事实上，有些问题的求解过程可以通过一定的方程进行一定的运算来获取。例如，求解梁架结构中应力的数学模型为线性方程组；预报人口增长情况的数学模型为微分方程。然而，更多的非数值计算问题却无法用数学方程加以描述。下面请看几个例子。

例1-1 图书馆的书目检索系统自动化问题。

当你想借阅一本参考书，但不知道书库中是否有该书的时候；或者，当你想找某一方面的参考书而不知图书馆内有哪些这方面的书时，都需要到图书馆去查阅图书目录卡片。在图书馆内有各种名目的卡片：有按书名编排的、有按作者名编排的，还有按分类编排的，等等。若利用计算机实现自动检索，则计算机处理的对象便是这些目录卡片上的书目信息。列在一张卡片上的一本书的书目信息可由登录号、书名、作者名、分类号、出版单位和出版时间等若干项组成。每一本书都有唯一的一个登录号，但不同的书目之间可能有相同的书名，或者有相同的作者名，或者有相同的分类号。由此，在书目自动检索系统中可以建立一张按登录号顺序排列的书目文件和三张分别按书名、作者名和分类号顺序排列的索引表，如图1-1所示。由这四张表构成的文件便是书目自动检索的数学模型。计算机的主要操作便是按照某个特定要求(如给定书名)对书目文件进行查询。诸如此类的还有查号系统自动化、仓库账目管理等。在这类文档管理的数学模型中，计算机处理的对象之间通常存在着的是一种最简单的线性关系。这类数学模型可称为线性的数据结构。

例1-2 酒店管理系统中的客房分配问题。

在酒店的客房房态管理过程中，希望同类房中各间客房的出租机会基本均等，以保证维持一个平均的磨损率。为此，分配客房采用的算法应该是“先退的房先被启用”。相应地，所有“空”的同类客房的管理模型应该是一个“队列”，即酒店前台每次接待客人入住时，从“队头”分配客房；当客人结账离开时，应将退掉的空客房排在“队尾”。由于“排队”是日常生活中经常需要的一种行为，因此队列也是这样一类活动的模拟程序中经常用的一种数学模型。

001	高等数学	樊映川	S01	……
002	大学物理	王微微	W01	……
003	化学工程	郝兵	H01	……
004	计算机网络	万树蜂	J01	……
……	……	……	……	……

高等数学	001,……
大学物理	002,……
计算机网络	004,……
……	……

樊映川	001,……
……	……

W	002,……
S	001,……
J	004,……
……	……

图 1-1　图书目录文件示例

例 1-3　铺设煤气管道问题。

假设要在某个城市的 n 个居民区之间铺设煤气管道，则在这 n 个居民之间只要铺设 n−1 条管道即可。假设任意两个居民区之间都可以架设管道，但由于地理环境的不同，所需经费也不同，则采用什么样的施工方案能使总投资尽可能少。这个问题即为“求图的最小生成树”的问题。其数学模型为如图 1-2 所示的“图”，图中“顶点”表示居民区，顶点之间的连线及其上的数值表示可以架设的管道及所需经费。求解的算法为：在可能架设的 m 条管道中选取 n − 1 条，既能连通 n 个居民区，又使总投资达到“最小”。

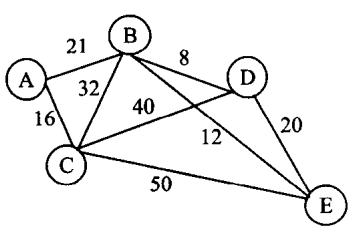


图 1-2　管道铺设问题

例 1-4　计算机和人对奕问题。

计算机之所以能和人对奕是因为有人将对奕的策略事先已存入计算机。由于对奕的过程是在一定规则下随机进行的，所以，为使计算机能灵活对奕就必须对对奕过程中所有可能发生的情况以及相应的对策都考虑周全，并且，一个“好”的棋手在对奕时不仅要看棋盘当时的状态，还应能预测棋局发展的趋势，甚至最后结局。因此，在对奕问题中，计算机操作的对象是对奕过程中可能出现的棋盘状态——称为格局，例如图 1-3(a)所示为井字棋的一个格局。格

局之间的关系是由比赛规则决定的。通常，这个关系不是线性的，因为从一个棋盘格局可以派生出几个格局。例如从图 1-3(a)所示的格局可以派生出五个格局，如图 1-3(b)所示，而从每一个新的格局又可派生出四个可能出现的格局。因此，若将从对奕开始到结束的过程中所有可能出现的格局都画在一张图上，则可得到一棵倒长的“树”。“树根”是对奕开始之前的棋盘格局，而所有的“叶子”就是可能出现的结局。对奕的过程就是从树根沿树叉到某个叶子的过程。“树”可以是某些非数值计算问题的数学模型，也是一种数据结构。

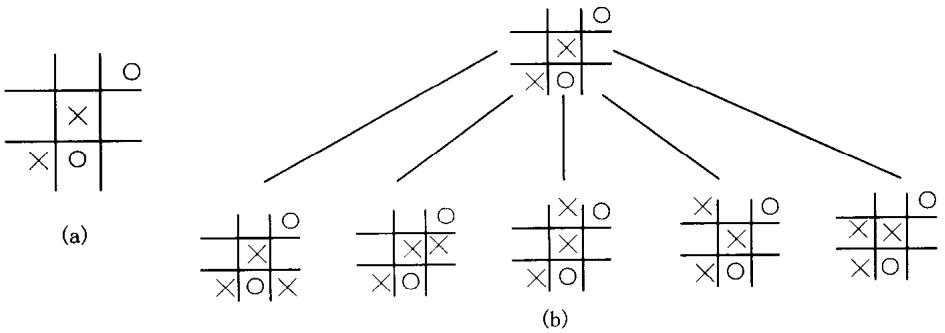


图 1-3 井字棋博弈树

综合上面几个例子可以看出，描述这类非数值计算问题的数学模型不再是数学方程，而是诸如表、树和图之类的数据结构。因此，简单说来，数据结构是一门研究非数值计算的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等的学科。

1.2 基本概念和术语

在本节中，为了与读者就某些概念取得“共识”，将对一些概念和术语赋以确定的含义。这些概念和术语将在以后的章节中多次出现。

数据(data)是信息的载体，是对客观事物的符号表示。它能够被计算机所识别、存储和加工处理，是计算机程序加工的“原料”。**数据**是指所有能输入计算机中，并被计算机程序处理的符号的总称。例如，一个利用数值分析方法解代数方程的程序，其处理对象是整数和实数；一个编译程序或文字处理程序的处理对象是字符串。因此，对计算机科学而言，数据的含义极为广泛，如图像、声音等都可以通过编码而归之于数据的范畴。

信息(information)是经过计算机加工处理的带有一定意义的结果，如方程式的解、遥感图像、视频信号等等。

数据元素(data element)是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。例如，例 1-4 中的“树”中的一个棋盘格局，例 1-3 中的“图”中的一个圆圈都被称为一个数据元素。有时，一个数据元素可由若干个数据项(data item)组成，例如，例 1-1 中一本书的书目信息为一个数据元素，而书目信息中的每一项(如书名、作者名等)为一个数据项。数据项是数据的不可分割的最小单位。

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，整数数据对象是集合 $N=\{0,\pm1,\pm2,\cdots\}$，字母字符数据对象是集合 $C=\{\text{'A'},\text{'B'},\cdots,\text{'Z'}\}$。

简单地说，**数据结构**(data structure)是相互之间存在一种或多种特定关系的数据元素的集合。通过上节的四个例子我们可以看到，在任何问题中，数据元素都不是孤立存在的，而是在它们之间存在着某种关系。这种数据元素相互之间的关系称为结构(structure)。根据数据元素之间关系的不同特性，通常有下列四类基本结构。

(1) 集合　结构中的数据元素之间除了“同属于一个集合”的关系外，别无其它关系。

(2) 线性结构　结构中的数据元素之间存在一个对一个的关系。

(3) 树形结构　结构中的数据元素之间存在一个对多个的关系。

(4) 图状结构或网状结构　结构中的元素之间存在多个对多个的关系。图 1-4 为上述四类基本结构的关系图。由于“集合”是元素之间关系极为松散的一种结构，因此也可用其它结构来表示它。

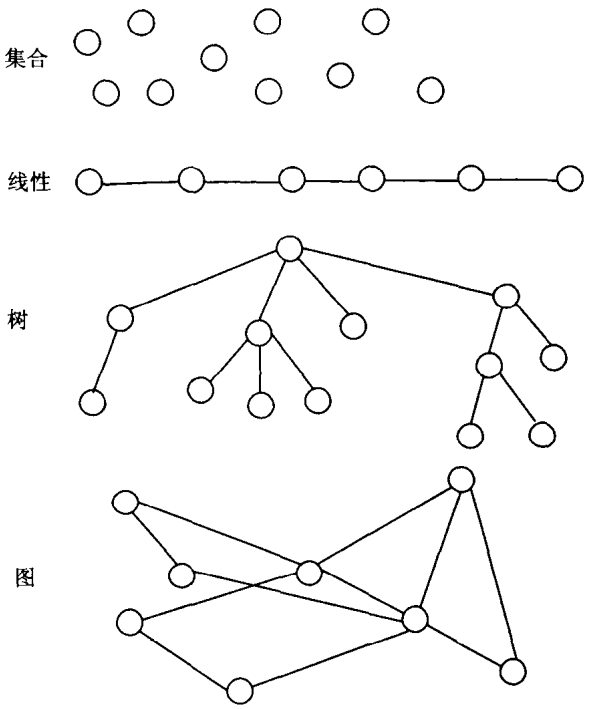


图 1-4　四种基本数据结构关系图

数据结构的形式定义为：

数据结构是一个二元组　Data-Structure=(D,S)　　(1-1)

其中：D 是数据元素的有限集，S 是 D 上关系的有限集。

下面举两个简单例子说明之。

例 1-5　在计算机科学中，复数是一种数据结构：

Complex=(C,R)　　(1-2)

其中：C 是含两个实数的集合{c1,c2}；R={P}，而 P 是定义在集合 C 上的一种关系{⟨c1,c2⟩}，其中有序偶⟨c1,c2⟩表示 c1 是复数的实部，c2 是复数的虚部。

例 1-6　假设需要编制一个事务管理的程序，管理学校科学研究课题小组的各项事务，

则首先要为程序的操作对象——课题小组设计一个数据结构。假设每个小组由一位教师、1～3名研究生及1～6名本科生组成，小组成员之间的关系是：教师指导研究生，而由每位研究生指导1～2名本科生，则可以如下定义数据结构：

$$Group=(P,R) \tag{1-3}$$

其中：$P=\{T,G_1,\cdots,G_n,S_{11},\cdots,S_{nm}\}_{1\leqslant n\leqslant 3,1\leqslant m\leqslant 2}$

$R=\{R_1,R_2\}$

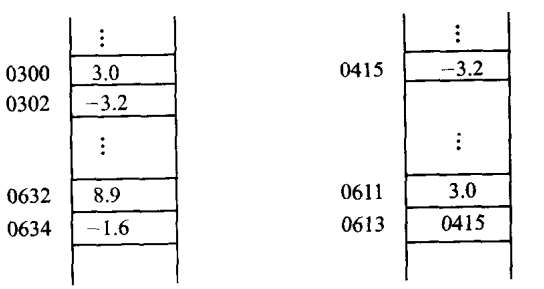
$R1=\{<T,G_i>\mid_{1\leqslant i\leqslant n,1\leqslant n\leqslant 3}\}$

$R2=\{<G_i,S_{ij}>\mid_{1\leqslant i\leqslant n,1\leqslant j\leqslant m,1\leqslant n\leqslant 3,1\leqslant m\leqslant 2}\}$

上述数据结构的定义仅是对操作对象的一种数学描述。换句话说，它是从操作对象抽象出来的数学模型。结构定义中的“关系”描述的是数据元素之间的逻辑关系，因此又称为数据的逻辑结构。然而，讨论数据结构的目的是为了在计算机中实现对它的操作，因此还需研究如何在计算机中表示它。

存储结构（又称映像）是数据结构在计算机中的表示，也称为数据的物理结构。它包括数据元素的表示和关系的表示。在计算机中表示信息的最小单位是二进制数的一位，叫做位（bit）。计算机中，可以用一个由若干位组合起来形成的一个位串表示一个数据元素（如用一个字长的位串表示一个整数，用八位二进制数表示一个字符等），通常称这个位串为**元素**（element）或**节点**（node）。当数据元素由若干数据项组成时，位串中对应于各个数据项的子位串称为**数据域**（data field）。因此，元素或节点可看成是数据元素在计算机中的映像。

数据元素之间的关系在计算机中有两种不同的表示方法：顺序映像和非顺序映像，并由此得到两种不同的存储结构：顺序存储结构和链式存储结构。顺序映像的特点是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。例如，假设用两个字长的位串表示一个实数，则可以用地址相邻的四个字长的位串表示一个复数，如图1-5(a)为表示复数z1=3.0－3.2i和z2=8.9－1.6i的顺序存储结构。



(a) 顺序存储结构　　(b) 链式存储结构

图1-5 复数存储结构示意图

非顺序映像的特点是借助指示元素存储地址的指针（pointer）表示数据元素之间的逻辑关系，如图1-5(b)为表示复数z1的链式存储结构。其中实部和虚部之间的关系用值为“0415”的指针来表示（0415是虚部的存储地址）。数据的逻辑结构和物理结构是密切相关的两个方面。在后面的章节读者会看到，任何一个算法的设计取决于所选定的数据（逻辑）结构，而算法的实现依赖于采用的存储结构。

那么，在数据结构确立之后如何描述存储结构呢？虽然存储结构涉及数据元素及其关系在存储器中的物理位置，但由于本书是在高级程序语言的层次上讨论数据结构的操作，因此不能如上所谈的那样直接以内存地址来描述存储结构。我们可以借用高级程序语言中提供的“数据类型”来描述它。例如可以用所有高级程序语言中都有的“一维数组”类型来描述顺序存储结构，以C语言提供的“指针”来描述链式存储结构。假如把C语言看成是一个执行C指令和C数据类型的虚拟处理器，那么本书中讨论的存储结构是数据结构在C虚拟处理器中的表示，不妨称它为虚拟存储结构。

数据类型(data type)是与数据结构密切相关的一个概念，用以刻画(程序)操作对象的特性。它最早出现在高级程序语言中。在用高级程序语言编写的程序中，每个变量、常量或表达式都有一个它所属的、确定的数据类型。类型明显或隐含地规定了在程序执行期间变量或表达式所有可能取值的范围，以及在这些取值上所允许进行的操作。因此数据类型是一个值的集合和定义在这个值集合上的一组操作的总称。例如C语言中的整数类型，其值集为区间[－maxint，maxint]上的整数(maxint是依赖特定的计算机的最大整数)，定义在其上的一组操作为：加、减、乘、整除和取模等。按“值”的不同特性，高级程序语言中的数据类型可分为两类：一类是非结构的原子类型，另一类是结构类型。原子类型的值是不可分解的，如C语言中的标准类型(整型、实型、字符型、布尔型和指针类型)。结构类型的值是由若干成分按某种结构组成的，因此，它是可以分解的。它的成分既可以是非结构的，也可以是结构的。例如数组的值由若干分量组成，每个分量可以是整数，也可以是数组等。在某种意义上，数据结构可以看成是“一组具有相同结构的值”，则结构类型可以看成由一种数据结构和定义在其上的一组操作组成。

实际上，在计算机中，数据类型的概念并非局限于高级语言中，每个处理器(包括计算机硬件系统、操作系统、高级语言、数据库等)都提供了一组原子类型或结构类型。例如，一个计算机硬件系统通常含有“位”、“字节”、“字”等原子类型。它们的操作通过计算机设计的一套指令系统直接由电路系统完成，而高级程序语言提供的数据类型，其操作需通过编译器或解释器转化成底层语言，即汇编语言或机器语言的数据类型来实现。引入“数据类型”的目的，从硬件的角度看，是作为解释计算机内存中信息含义的一种手段。而对使用数据类型的用户来说，实现了信息的隐蔽，即将一切用户不必了解的细节都封装在类型中。例如，用户在使用“整数”类型时，既不需要了解“整数”在计算机内部是如何表示的，也不需要知道其操作是如何实现的。如“两整数求和”，程序设计者注重的仅仅是其“数学上求和”的抽象特性，而不是其硬件的“位”操作如何进行。

抽象数据类型(abstract data type 简称 ADT)是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义仅取决于它的一组逻辑特性，而与其在计算机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。

抽象数据类型和数据类型实质上是一个概念。例如，各个计算机都拥有的“整数”类型是一个抽象数据类型。尽管它们在不同处理器上实现的方法可以不同，但由于其定义的数学特性相同，在用户看来是相同的。因此，“抽象”的意义在于数据类型的数学抽象特性。另一方面，抽象数据类型的范畴更广，它不再局限于前述各处理器中已定义，并实现的数据类型(也可称这类数据类型为固有数据类型)，还包括用户在设计软件系统时自己定义的数据类型。为了

提高软件的复用率，在近代程序设计方法学中指出，一个软件系统的框架应建立在数据之上，而不是建立在操作之上(后者是传统的软件设计方法所为)，即在构成软件系统的每个相对独立的模块上，定义一组数据和施于这些数据上的一组操作，并在模块内部给出这些数据的表示及其操作的细节，而在模块外部使用的只是抽象的数据和抽象的操作。显然，所定义的数据类型的抽象层次越高，含有该抽象数据类型的软件模块的复用程度也就越高。

一个含抽象数据类型的软件模块通常应包含定义、表示和实现三个部分。

如前所述，抽象数据类型的定义由一个值域和定义在该值域上的一组操作组成。若按其值的不同特性，可细分为下列三种类型。

原子类型(atomic data type) 属原子类型的变量的值是不可分解的。这类抽象数据类型较少，因为一般情况下，已有的固有数据类型足以满足需求。但有时也有必要定义新的原子数据类型，例如数位为100的整数。

固定聚合类型(fixed-aggregate data type) 属该类型的变量，其值由确定数目的成分按某种结构组成。例如，后面将要定义的复数是由两个实数依确定的次序关系构成的。

可变聚合类型(variable-aggregate data type) 和固定聚合类型相比较，构成可变聚合类型"值"的成分的数目不确定。例如，可定义一个"有序整数序列"的抽象数据类型，其中序列的长度是可变的。

显然，我们可以把后两种类型统称为结构类型。在抽象数据类型定义的操作中，除了和该类型相应的一组操作外，一般情况下还应有结构的创建和销毁。

抽象数据类型可通过固有数据类型来表示和实现，即利用处理中已存在的数据类型来说明新的结构；利用已经实现的操作来进行新的操作。正如前面所提到的，由于本书是在高级程序语言的层次上进行讨论，因此对抽象数据类型也只是在这虚拟层上讨论它的实现。下面以复数为例说明。

例1-7 抽象数据类型复数的定义。

在例1-5中式(1-2)的逻辑结构定义实际上是给定了复数的值域，为两个实数集的笛卡尔积，即

$$Z=R\times R=\{<c1,c2> \mid c1\in R,c2\in R\} \tag{1-4}$$

其中R表示实数集，Z表示复数集。

下面定义复数的六种操作。

(1) CREATE(x,y,z)生成一个复数。

对任何一对实数x,y(x∈R,y∈R)，必可生成一个复数域中的复数z=x+iy。

(2) ADD(z1,z2,sum)复数求和。

对复数域Z中的任意两个复数z1=x1+iy1和z2=x2+iy2，必可求得其和为

sum=(x1+x2)+i(y1+y2)

(3) SUBTRACT(z1,z2,difference)复数求差。

对复数域Z中的任意两个复数z1=x1+iy1和z2=x2+iy2，必可求得其差为