



普通高等教育“十一五”精品规划教材

实用数据结构与 算法设计

SHIYONG SHUJU JIEGOU YU SUANFA SHEJI

主编 庄晋林 杨彬



中国水利水电出版社
www.waterpub.com.cn

普通高等教育“十一五”精品规划教材

实用数据结构与 算法设计



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书共分9章, 主要内容包括绪论、基本线性结构、拓展线性结构、树结构、图结构、排序与查找、文件、实验与课程设计。

本书从实用的角度, 在注重理论基础的前提下强调了实践的重要性。本书介绍了各种常用数据结构的存储以及相应的实现算法, 并配有丰富的算法应用实例, 这些算法结构化程度高、可读性好, 并具有较强的实用性。同时本书突出了基本技能的训练, 使读者在掌握理论知识的基础上, 进一步培养分析、思考和解决问题的能力。

本书适用于计算机科学与技术及相关专业应用型本科层次的学生, 亦可作为相关专业教师或工程技术人员的参考书。

图书在版编目(CIP)数据

实用数据结构与算法设计 / 庄晋林, 杨彬主编. --
北京: 中国水利水电出版社, 2009

普通高等教育“十一五”精品规划教材
ISBN 978-7-5084-6806-8

I. ①实… II. ①庄… ②杨… III. ①数据结构—高等学校—教材②算法分析—高等学校—教材 IV.
①TP311.12

中国版本图书馆CIP数据核字(2009)第152999号

书 名	普通高等教育“十一五”精品规划教材 实用数据结构与算法设计
作 者	主编 庄晋林 杨彬
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)
经 售	北京科水图书销售中心(零售) 电话: (010) 88383994、63202643 全国各地新华书店和相关出版物销售网点
排 版	中国水利水电出版社微机排版中心
印 刷	北京纪元彩艺印刷有限公司
规 格	184mm×260mm 16开本 23.5印张 557千字
版 次	2009年8月第1版 2009年8月第1次印刷
印 数	0001—3000册
定 价	45.00元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

前言

自从 20 世纪 60 年代末 Donald E. Knuth 开创了数据结构的最初体系以来，数据结构课程的内容已经变得日益完善。由权威机构 IEEE - CS/ACM (The Computer Society of the Institute for Electrical and Electronic Engineers/The Association for Computing Machinery) 推出的一系列课程体系中，都把数据结构作为其核心体系的组成部分。在“中国计算机科学与技术学科教程 2002” (China Computing Curricula 2002, 简称 CCC2002) 以及由教育部高等学校计算机科学与技术教学指导委员会 2007 年推出的“高等学校计算机科学与技术专业公共核心知识体系与课程”中，也都把数据结构作为计算机专业主要的核心课程。可见，数据结构在整个计算机专业课程体系中处于举足轻重的地位。该课程属于专业技术基础课，它与计算机软件、硬件和数学密切相关，是设计与实现操作系统、数据库系统、编译程序及其他系统程序和大型应用程序的重要基础。课程内容包括软件设计中常用的数据结构及其相应的存储结构和算法设计，涉及多种查找和排序技术，并对算法的性能进行分析和评价。

2009 年硕士研究生入学考试将计算机科学与技术学科专业基础综合考试纳入全国统一命题的范围，并制订了计算机学科专业基础综合考试大纲。该大纲涵盖数据结构、计算机组成原理、操作系统和计算机网络 4 方面的内容，其中数据结构比例占 30%，处于优先地位。本书参照该考试大纲“数据结构”科目的主要核心内容，同时兼顾应用型本科层次学生的实际需要，以培养创新精神和提高实践能力为目标，为改变读者在学习该课程时普遍感到抽象、繁琐、偏难的现状，在内容编排与撰写上，力求深入浅出、表述严谨、突出重点、通俗易懂，以使读者在掌握基本概念、基本知识的基础上，便于自学和巩固。

本书知识点全面，注重理论与实践的结合。除了全面阐述基本概念、基本原理和方法外，还列举了大量的应用实例，帮助读者理解各种数据结构及其算法的设计与实现。在此基础上，针对课程重点内容，给出了实验举例，安排

了适当的实验选题与课程设计科目，读者可进一步熟悉数据结构的实际应用，并通过实战训练，强化动手实践和解决实际问题能力。

本书分为9章，可以按照60学时左右安排理论教学。内容包括绪论、基本线性结构、拓展线性结构、树结构、图结构、排序与查找、文件、实验与课程设计。

全书编写工作分工如下：庄晋林编写第1、9章；杨彬编写第4章；布辉编写第6章1~10节；刘冉编写第6章11~13节、第7章及第8章第8节；刘素华编写第5章；朱国仲编写第2章及第8章1~4节；常呈果编写第3章及第8章5~7节。全书由庄晋林、杨彬负责统稿。

在本书的编写过程中，得到了华北水利水电学院“数据结构”精品课程项目的支持。同时，在编辑出版过程中得到了中国水利水电出版社编辑同志的热情帮助和指导，对此，我们表示诚挚的感谢。

由于时间仓促及作者水平所限，书中难免存在一些错误和偏颇，恳请读者批评指正。

编者

2009年7月

目录

前言

第 1 章 绪论	1
1.1 数据结构的发展史及地位	1
1.2 数据结构的定义	3
1.3 数据类型	7
1.4 算法及算法分析	10
1.5 程序设计技术简介	17
1.6 小结	25
习题 1	25
参考文献	30
第 2 章 基本线性结构	31
2.1 线性表	31
2.2 线性表的顺序存储结构与运算	33
2.3 线性表的链式存储结构与运算	38
2.4 栈	51
2.5 队列	57
2.6 串	65
2.7 应用实例	85
2.8 小结	101
习题 2	101
参考文献	104
第 3 章 拓展线性结构	105
3.1 数组的定义	105
3.2 数组基本运算的实现	107
3.3 矩阵的压缩存储与运算	111
3.4 广义表	128
3.5 小结	148
习题 3	149

参考文献	150
第 4 章 树结构	151
4.1 树的概念与运算	151
4.2 二叉树	156
4.3 二叉树遍历及线索二叉树	164
4.4 树和森林	175
4.5 应用实例	184
4.6 小结	197
习题 4	197
第 5 章 图结构	201
5.1 图的概念	201
5.2 图的存储结构	206
5.3 图的遍历	213
5.4 生成树	218
5.5 应用实例	226
5.6 小结	241
习题 5	241
参考文献	245
第 6 章 排序与查找	246
6.1 排序概述	246
6.2 插入排序	248
6.3 交换排序	253
6.4 选择排序	256
6.5 归并排序	261
6.6 基数排序	262
6.7 各种内部排序方法的比较与选择	266
6.8 查找概述	267
6.9 线性表的查找	268
6.10 树表上的查找	275
6.11 哈希表查找	287
6.12 应用实例	295
6.13 小结	299
习题 6	299
参考文献	301
第 7 章 文件	302
7.1 文件的概念	302
7.2 顺序文件	305

7.3 索引文件	307
7.4 索引顺序文件	310
7.5 散列文件	315
7.6 多关键字文件——多重表文件	316
7.7 小结	319
习题 7	319
参考文献	320
第 8 章 实验	321
8.1 顺序表的操作	321
8.2 链表的操作	328
8.3 栈的操作	336
8.4 队列的应用	341
8.5 二叉树的操作	345
8.6 图的操作	350
8.7 二叉排序树的操作	354
8.8 哈希表的操作	358
第 9 章 课程设计	363
9.1 课程设计概述	363
9.2 课程设计选题	364

第 1 章 绪 论

在计算机广泛普及的今天，其应用几乎涵盖了人类社会的所有领域，而且在航空航天、军事、科学计算、信息检索、生产线控制等一些关键领域已经高度依赖计算机系统。由于受莫尔（Gordon Moore，英特尔公司创始人之一）定律的驱动，计算机硬件系统的性能及其可靠性迅速提高。在硬件系统日趋完善的基础上，如何极大地发挥计算机系统的效能，则取决于软件系统的功能及性能。因此，在计算机应用领域中，设计高效而可靠的算法已成为充分发挥计算机效能的基本前提。

虽然电子计算机经历了 60 余年的发展历史，但是主流计算机的系统结构始终无法摆脱冯·诺依曼体系的束缚，算法设计和数据存储仍然是计算机软件设计者必须面对的两大问题。计算机本质上是一个数据处理系统，而数据是计算机化的信息，它是计算机可以直接处理的最基本和最重要的对象。数据结构这门课程研究的主要内容就是解决如何有效地存储数据以及设计正确的算法对其进行加工处理。

1.1 数据结构的发展史及地位

1.1.1 数据结构发展简史

1968 年，美国斯坦福大学教授、图灵奖获得者 Donald E. Knuth 开始出版在计算机领域具有划时代意义的名著《计算机程序设计艺术》（The Art of Computer Programming）。该书曾与牛顿的《自然哲学的数学原理》等书一起，被评为“世界历史上最伟大的十种科学著作”之一。Bill Gates 曾经说过：“如果你认为你确实是一个好的程序员，读一读 Knuth 的《计算机程序设计艺术》吧，要是你真把它读通了，你就可以给我递简历了”。这套丛书的第一卷《基本算法》（Fundamental Algorithms）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，可以认为这本书开创了数据结构的最初体系。此外，Mary d'Imperio 的文章《Data Structures and their Representation in Storage, Annual Review in Automatic Programming 5》（Oxford: Pergamon Press, 1969）给出了关于数据结构与程序设计的一般关系的论述，这篇论文对于这一课题给出了历史性的指南，它包括了 12 个线性表处理系统和字符串处理系统所用结构的详细分析。从此，数据结构作为一门独立的课程开始进入大学课堂，但当时并没有明确规定课程的范围。数据结构的研究范畴起初主要包括线性表、数组、树、图、排序等内容，随着计算机科学与技术不断发展，数据结构的内容也随之更新。比如，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。

在我国，自 1978 年美籍华裔学者冀中田在国内首次开设这门课程以来，经历了 30 余

年的发展，这门课程已成为各大学计算机科学与技术专业学生的核心课程，同时也是信息技术类相关专业学生和研究生学习计算机的主要选修课程。

1.1.2 数据结构在计算机学科中的地位

在计算机发展的早期，计算机的主要作用是处理数值计算问题。所涉及的主要运算对象是简单的整型、实型或布尔型数据，因此，程序设计者往往把主要精力集中于程序设计的技巧上，而无须重视数据结构。随着计算机软、硬件的发展和应用领域的不断扩大，计算机处理的对象越来越复杂，涉及的问题诸如文字编辑、语言翻译、表格、图形图像、音频、视频处理等。对这些对象的处理主要是非数值型的计算问题，这类计算在计算机应用中占的比例逐年增加，已经远远超过单纯的数值计算。而且，这类问题涉及的数据结构更为复杂，这是由于数据元素之间的相互关系通常不能用数学方程式加以描述。因此，解决这类问题的关键不再是数学分析和计算方法，而是要设计出合适的数据结构以及相应的算法，才能有效地解决问题。

由于数据结构是研究数据之间的逻辑关系以及在计算机中如何存储和使用这些数据等相关问题，它涉及数据的组织、存储及其相关的运算。因此，与数据结构相关的学科涵盖了计算机硬件、软件及数学，而且和计算机软件的研究有着更密切的关系，它是三者之间的一个交集。

在计算机软件开发中，几乎所有的计算机系统软件和应用软件都要用到各种类型的数据结构。因而，数据结构不仅是设计一般程序（特别是非数值计算的程序设计）的基础，而且还是计算机学科中操作系统、编译原理、数据库管理系统、软件工程和人工智能等一系列与软件相关的专业课程的先修课，起着承上启下的关键作用。同时，数据结构技术也广泛应用于信息科学、系统工程、应用数学以及各种工程技术领域。打好“数据结构”这门课程的扎实基础，对于学习计算机专业的其他课程非常重要。“数据结构”课程属于计算机科学与技术专业的重要的专业基础课，而且，教育部决定，从 2009 年起，对全国硕士研究生入学统一考试计算机科学与技术学科的初试科目进行调整，把计算机学科专业基础综合（考试内容包括数据结构、计算机组成原理、操作系统和计算机网络）作为全国统一命题的计算机科学与技术学科联考科目。其中数据结构所占比例达 30%，处于优先位置。

数据结构与算法就像孪生兄弟，相互依存、密不可分。著名计算机科学家沃思（Niklaus Wirth）提出的关于算法与数据结构关系的著名公式：算法+数据结构=程序，直观地表达了三者之间的关系。

算法与数据结构属于计算学科的一个重要研究分支。所谓计算学科是对描述和变换信息的算法过程的系统研究，其研究范围包括它的理论、分析、设计、效率、实现和应用。计算学科来源于对算法理论、数理逻辑、计算模型、自动计算机器等的研究，它与存储程序式电子计算机的发明一起形成于 20 世纪 40 年代初。计算学科要回答的基本问题是“什么能被（有效地）自动进行？”

在计算机学科（计算学科）教育方面最有代表性和影响力的机构 IEEE-CS/ACM（The Computer Society of the Institute for Electrical and Electronic Engineers/The Association for Computing Machinery）推出的一系列课程体系，先后包括 CC1991（Computing Curricula 1991）、CC2001 及 CC2004，都把数据结构作为其核心体系的组成部分。在参照 CC2001 制定

的适合中国国情的“计算机科学与技术学科教程 2002”（China Computing Curricula 2002，简称 CCC2002）以及由教育部高等学校计算机科学与技术教学指导委员会 2007 年推出的“高等学校计算机科学与技术专业公共核心知识体系与课程”中，也都把数据结构作为计算机专业主要的核心课程。由此可见，数据结构在整个计算机专业教学体系中处于举足轻重的地位。

学习数据结构的目的是为了掌握计算机处理对象的特性，将实际问题中所涉及的处理对象在计算机中表示出来并对它们进行处理。与此同时，通过算法训练来提高学生的逻辑思维能力，通过程序设计的技能训练来促进学生的综合应用能力和专业素质的提高。

1.2 数据结构的定义

1.2.1 数据结构概念

在计算机科学界对于数据结构并没有统一的标准定义。可以列举一些不同的描述性定义。比如：Sartaj Sahni 在他的《数据结构、算法与应用》一书中称：“数据结构是数据对象，以及存在于该对象的实例和组成实例的数据元素之间的各种联系。这些联系可以通过定义相关的函数来给出。”他将数据对象（data object）定义为“一个数据对象是实例或值的集合”；Clifford A. Shaffer 在《数据结构与算法分析》一书中的定义是：“数据结构是 ADT（Abstract Data Type，即抽象数据类型）的物理实现”；Lobert L.Kruse 在《数据结构与程序设计》一书中，将一个数据结构的设计过程分成抽象层、数据结构层和实现层。其中，抽象层是指抽象数据类型层，它讨论数据的逻辑结构及其运算，数据结构层和实现层，讨论一个数据结构的表示和在计算机内的存储细节以及运算的实现。

一般认为，一个数据结构是由数据元素依据某种逻辑联系组织起来的。对数据元素间逻辑关系的描述称为数据的逻辑结构；数据必须在计算机内存储，数据的存储结构是数据结构的实现形式，是其在计算机内的表示；此外在讨论一个数据结构时必须同讨论在该类数据上执行的运算才有意义。

下面通过实例引入数据结构的概念。

【例 1-1】 高校运动会成绩管理。田径运动是高校体育活动的重要组成部分，绝大多数学校每年都要举办田径运动会。与其他单项竞赛（如球类比赛）相比，田径运动会项目多、参赛运动员多、筹备工作复杂、竞赛时间集中，所有信息都要求迅速、准确地进行处理。用计算机系统完成田径运动会报名、日程编排、成绩统计以及各类信息处理工作，对于提高工作效率和管理水平，实现体育运动会管理的现代化和规范化具有现实意义。下面以某个小型运动会成绩单为例说明什么是线性数据结构。

表 1-1 和表 1-2 分别表示两个单项比赛的最终成绩；表 1-3 表示各代表队的总成绩。可以对表格中的数据进行查询、浏览、插入、修改、删除与统计等操作。

表 1-1 男子 100 米成绩单

名次	运动员编号	姓名	年龄	所属代表队	成绩（秒）	是否破纪录	积分	备注
1	621	杜军永	19	信息工程学院	11.7	是	8+10	破纪录加分
2	133	赵成龙	18	水利学院	12.3	否	7	
3	357	宋智军	21	交通学院	12.5	否	6	

续表

名次	运动员编号	姓名	年龄	所属代表队	成绩(秒)	是否破纪录	积分	备注
4	463	林雨森	20	人文学院	12.7	否	5	
5	519	陈启	19	建筑学院	12.8	否	4	
6	145	于明亮	21	环境工程学院	12.9	否	3	
7	343	王子超	22	电力学院	13.1	否	2	
8	452	袁培良	20	机械学院	13.2	否	1	

表 1-2 女子跳远成绩单

名次	运动员编号	姓名	年龄	所属代表队	成绩(米)	是否破纪录	积分	备注
1	162	尹晶晶	18	水利学院	3.82	是	8+10	破纪录加分
2	322	黄文静	20	电力学院	3.78	是	7+10	破纪录加分
3	671	王晓娜	21	信息工程学院	3.67	否	6	
4	156	史珍珍	17	环境工程学院	3.62	否	5	
5	422	张琼兰	19	人文学院	3.59	否	4	
6	428	王倩	21	机械学院	3.53	否	3	
7	177	曹振婷	22	水利学院	3.50	否	2	
8	546	袁霞	18	建筑学院	3.39	否	1	

表 1-3 各代表队总成绩(得分表)

名次	代表队	男子 100米	女子 100米	男子 1500米	女子 1500米	男子 铅球	女子 铅球	男子 跳远	女子 跳远	总积分
1	水利学院	7	6	18	13	11	4	0	20	79
2	信息工程学院	18	7	3	5	5	8	11	6	63
3	电力学院	2	3	5	8	6	7	10	17	58
4	建筑学院	4	5	7	0	18	0	2	1	37
5	交通学院	6	18	2	2	0	3	0	0	31
6	人文学院	5	4	1	3	4	7	1	4	29
7	机械学院	1	3	6	4	0	2	4	3	23
7	环境工程学院	3	0	4	1	5	5	0	5	23

以上表格内的数据前后均按名次排序,它们之间存在着线性关系,属于线性数据结构。

【例 1-2】 人类家族继承关系。表 1-4 以三国时期蜀汉国缔造者刘备的家族关系为例来说明“树”这种数据结构。

表 1-4 刘 备 家 谱

第一代	刘 备											
第二代	刘 禅							刘永	刘 理			刘封 (养子)
第三代	刘睿	刘瑶	刘琮	刘瓚	刘谌	刘恂	刘虔		刘胤	刘辑	刘承	刘林

刘备家谱的继承关系可以表示为树型结构,如图 1-1 所示,它像一棵倒长的树。在实际中这样的结构很多,比如:政府管理部门、企事业单位组织机构等表示上下级从属关系的结构都是树型结构。

个基本单位进行访问和处理的。

数据对象 (data object): 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, 数据元素是数据对象的一个实例。例如, 在交通咨询系统的交通网中, 所有的顶点是一个数据元素类, 顶点 A 和顶点 B 各自代表一个城市, 是该数据对象的两个实例, 其数据元素的值分别为 A 和 B。

数据结构 (data structure): 是指互相之间存在着一种或多种关系的数据元素的集合。在任何问题中, 数据元素之间不会是孤立的, 在它们之间都存在着这样或那样的关系, 这种数据元素之间的关系称为结构。根据数据元素间关系的不同特性, 通常有下列 4 类基本的结构:

(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”。集合是一种松散的关系结构。

(2) 线性结构。该结构的数据元素之间存在着一对一的关系。通常元素之间有先后关系, 后面的元素相对于前面的元素称为后继; 反之, 称为前驱。

(3) 树型结构。该结构的数据元素之间存在着一对多的关系。通常元素之间存在“上下级”的层次关系, 也称之为“前驱”和“后继”。

(4) 图型结构。该结构的数据元素之间存在着多对多的关系, 图型结构也称为网状结构。图型结构中的元素之间可能存在着各种各样的复杂关系。

图 1-3 所示为表示上述 4 类基本结构的示意图。

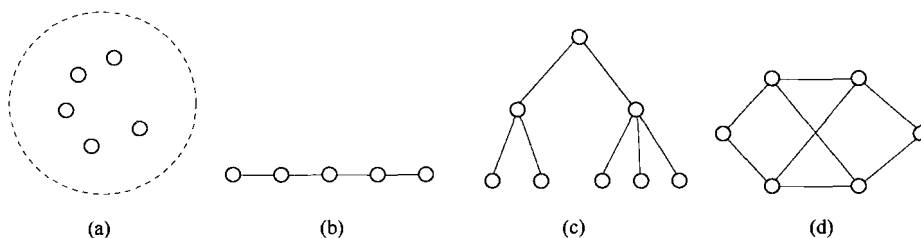


图 1-3 4 类基本结构的示意图

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

由于集合是数据元素之间关系极为松散的一种结构, 因此也可用其他结构来表示它。

从上面所介绍的数据结构的概念中可以知道, 一个数据结构有两个要素。一个是数据元素的集合, 另一个是关系的集合。在形式上, 数据结构通常可以采用一个二元组来表示。

数据结构的定义形式为: 数据结构是一个二元组

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

其中, D 是数据元素的有限集; R 是 D 上关系的有限集。

如前所述, 数据结构包括数据的逻辑结构和数据的物理结构。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, 它与数据的存储无关。研究数据结构的目的是为了在计算机中实现对它的操作, 为此还需要研究如何在计算机中表示一个数据结构。数据结构在计算机中的表示 (又称映像) 称为数据的物理结构, 或称存储结构。它所研究的是数据结构在计算机中的实现方法, 包括数据结构中元素的表示及元素间关系的表示。

数据的存储结构可采用顺序存储或链式存储的方法。

顺序存储方法是把逻辑上相邻的元素存储在物理位置相邻的存储单元中，由此得到的存储表示称为顺序存储结构。顺序存储结构是一种最基本的存储表示方法，通常借助于程序设计语言中的数组来实现。

链式存储方法对逻辑上相邻的元素不要求其物理位置相邻，元素间的逻辑关系通过预设的指针字段来表示，由此得到的存储表示称为链式存储结构，链式存储结构通常借助于程序设计语言中的指针类型来实现。

1.3 数据类型

目前标准的讲授数据结构方法是：先给出抽象数据类型的定义，然后再根据具体的存储结构给出实现。这种方法第一次出现在 A. V. Aho、J. E. Hopcroft 和 J. D. Ulman 所著的《数据结构和算法》(Data Structure and Algorithms. Reading, Mass: Addison-Wesley, 1983) 中。

下面首先回顾一下在程序设计语言中出现的数据类型概念。

1.3.1 数据类型的概念

数据类型 (data type) 是高级程序设计语言中的基本概念，用它来刻画程序中操作对象的特性。在高级语言编写的程序中，每个变量、常量或表达式都有一个确定的数据类型。数据类型可以通过显式地或隐含地进行说明，它强制性地规定了在程序执行期间变量或表达式所有可能的取值范围，以及在这些值上允许进行的操作。比如在标准 C 语言中说明的 int 型变量，它的取值范围是 $-32768 \sim +32767$ ，并可进行一元运算 +、- 以及二元运算 +、-、*、/、%、mod 等。因此，数据类型是一个值的集合和定义在该值集上的一组操作的总称。实质上，数据类型就是高级程序设计语言中已经实现的一些基本数据结构。

在高级程序设计语言中，数据类型可分为两类：一类是原子类型；另一类则是结构类型。原子类型的值是不可分解的。如 C 语言中整型、字符型、浮点型、双精度型等基本类型，分别用保留字 int、char、float、double 标识。而结构类型的值是由若干成分按某种结构组成的，因此是可分解的，并且它的成分可以是非结构的，也可以是结构的。例如，一维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数据”，而数据类型则可被看成是由一种数据结构和定义在其上的一组操作所组成的。

1.3.2 抽象数据类型

所谓抽象是指从特定的实例中抽取共同的性质以形成一般化概念的过程。对一个系统进行的抽象描述称为对它的规范说明，而对抽象的解释称为它的实现。抽象可以分为不同的层次，高层次的抽象是把它低层次抽象看作其一种实现。抽象是人类理解复杂现象和求解复杂问题时使用的重要手段。

数据抽象是指对数据以及操作该数据的算法的抽象。它体现了模块化和信息隐蔽两种思想。模块化思想提供了一个将复杂系统分解为若干子系统（或模块）的手段，每个模块包含了系统中与该模块相关的所有信息。模块化使系统便于理解与维护；信息隐蔽则使得

一个模块的实现细节对用户隐蔽起来，而模块的外部特性（操作和数据）由它的接口来体现，用户或外部模块对它的访问只能通过接口来进行。信息隐蔽使模块出现错误时，其影响被限制在自身范围内，有利于提高系统的可靠性。

在高级程序设计语言中的数据类型的就是数据抽象的体现。抽象数据类型则是数据类型概念的进一步延伸与扩展。

抽象数据类型（Abstract Data Type, 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。

可见抽象数据类型和数据类型实质上是一个概念。例如，各种高级程序设计语言都拥有的整数类型就是一个抽象数据类型，尽管它们在不同处理器上的实现方法可以不同，但由于其定义的数学特性相同，在用户看来都是相同的。因此，“抽象”的意义在于数据类型的数学抽象特性，只是抽象数据类型比传统意义上高级语言中已定义并实现的数据类型的范畴更广，它可以包括用户在设计软件系统时自己定义的数据类型。

抽象数据类型的定义可以由一种数据结构和定义在其上的一组操作组成，而数据结构又包括数据元素及元素间的关系，因此抽象数据类型实质上可以由元素、关系及操作三要素来定义，即抽象数据类型可用 (D, S, P) 三元组表示，其中， D 是数据对象， S 是 D 上的关系集， P 是对 D 的基本操作集。

此外，在设计抽象数据类型时，通常把类型的定义与其实现分离开来，从而达到信息隐蔽的目的。以下是两个抽象数据类型的实例。

【例 1-4】 对于矩阵及其相关运算，可以定义以下矩阵的抽象数据类型：

ADT Matrix{

数据对象： $D = \{a_{ij} \mid i=1, 2, \dots, m; j=1, 2, \dots, n\}$;

$a_{ij} \in \text{ElemSet}$, m 和 n 分别为矩阵的行数和列数}

数据关系： $R_1 = \{\text{Row}, \text{Col}\}$

$\text{Row} = \{ \langle a_{ij}, a_{i,j+1} \rangle \mid 1 \leq i \leq m, 1 \leq j \leq n-1 \}$

$\text{Col} = \{ \langle a_{ij}, a_{i+1,j} \rangle \mid 1 \leq i \leq m-1, 1 \leq j \leq n \}$

基本操作：

MatrixCreate (&M) ;

操作结果：创建矩阵 M 。

Transpose (M, &T) ;

初始条件：矩阵 M 存在。

操作结果：求矩阵 M 的转置矩阵 T 。

Add (M, N, &S) ;

初始条件：矩阵 M 与 N 的行、列数分别对应相等。

操作结果：求矩阵 M 与 N 的和 $S=M+N$ 。

Sub (M , N , & S) ;

初始条件: 矩阵 M 与 N 的行、列数分别对应相等。

操作结果: 求矩阵 M 与 N 的差 $S=M-N$ 。

Multi (M , N , & S) ;

初始条件: 矩阵 M 的列数与 N 的行数相等。

操作结果: 求矩阵 M 与 N 的积 $S=M \cdot N$ 。

Copy (M , & N) ;

初始条件: 矩阵 M 存在。

操作结果: 将矩阵 M 复制到矩阵 N 。

Inversion (M , & N) ;

初始条件: 矩阵 M 存在。

操作结果: 求矩阵 M 的逆矩阵 N 。

Print (M) ;

初始条件: 矩阵 M 存在。

操作结果: 输出矩阵 M 的内容。

} ADT Matrix

【例 1-5】 复数及其相关运算的抽象数据类型定义:

ADT Complex {

数据对象: $D=\{e_1, e_2 \mid e_1, e_2 \in \text{RealSet} \}$

数据关系: $R_1=\{ \langle e_1, e_2 \rangle \mid e_1 \text{ 是复数的实数部分, } e_2 \text{ 是复数的虚数部分} \}$

基本操作:

ComplexInit (& Z , v_1 , v_2)

操作结果: 构造复数 Z , 其实部和虚部分别被赋予参数 v_1 和 v_2 的值。

ComplexDestroy (& Z)

操作结果: 复数 Z 被销毁。

GetReal (Z , &realPart)

初始条件: 复数 Z 已存在。

操作结果: 用 realPart 返回复数 Z 的实部值。

GetImag (Z , &ImagPart)

初始条件: 复数 Z 已存在。

操作结果: 用 ImagPart 返回复数 Z 的虚部值。

Add (Z_1 , Z_2 , &sum)

初始条件: Z_1 , Z_2 是复数。

操作结果: 用 sum 返回两个复数 Z_1 、 Z_2 的和值。

} ADT Complex