21世纪高等院校计算机科学与工程系列教材

张 勇 杨喜权 刘君义 编 著

数据结构

中国林业出版社
China Forestry Publishing House
www.cfph.com.cn

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

21世纪高等院校计算机科学与工程系列教材

张 勇 杨喜权 刘君义 编 著

数据结构

中国林业出版社
China Forestry Publishing House
www.cfph.com.cn

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

本书主要介绍了线性表、栈结构、队列结构、数组和字符串结构、树状结构、图形结构、文件结构等数据结构的基本知识和实际应用，以及递归、查找、排序的设计方法及分析技巧，并在每章后面备有大量有针对性的习题。

本书每一个算法均先通过具体实例解释引出，然后采用 C 语言编写对应的具体程序，书中所有程序均已通过调试运行，并有非常详细的注释，使读者能迅速掌握复杂的数据结构及其算法。

本书内容丰富、语言精炼、通俗易懂，可作为高等院校计算机专业本科学生，以及非计算机学科本科学生攻读第二学位的教材，也可作为广大工程技术人员的参考书。

图书在版编目（CIP）数据

数据结构/张勇，杨喜权，刘君义编著.—北京：中国林业出版社：北京希望电子出版社，2006.8
（21 世纪高等院校计算机科学与工程系列教材）
ISBN 7-5038-4295-4

I.数... II.①张...②杨...③刘... III.数据结构－高等学校－教材 IV. TP311.12

中国版本图书馆 CIP 数据核字（2006）第 040020 号

出版： 中国林业出版社（100009 北京市西城区刘海胡同 7 号 010-66184477）
北京希望电子出版社（100085 北京市海淀区上地 3 街 9 号金隅嘉华大厦 C 座 611）
网址：www.bhp.com.cn 电话：010-82702660（发行） 010-62541992（门市）
印刷： 北京媛明印刷厂
发行： 全国新华书店经销
版次： 2006 年 8 月第 1 版
印次： 2006 年 8 月第 1 次
开本： 787mm×1092mm 1/16
印张： 19.75
字数： 453 千字
印数： 0001～3000 册
定价： 27.00 元

21世纪高等院校计算机教材编委会名单

（排名不分先后）

主　任：	陈火旺	院士	
副主任：	李仁发	教授	金茂忠 教授　陈　忠 教授　陆卫民 高工
委　员：	赵宏利	教授	装备指挥学院
	晏海华	教授	北京航空航天大学
	邵秀丽	教授	南开大学
	刘振安	教授	中国科技大学
	董玉德	副教授	合肥工业大学
	倪志伟	教授	合肥工业大学
	吕英华	教授	东北师范大学
	杨喜权	副教授	东北师范大学
	朱诗兵	副教授	装备指挥学院
	樊秀梅	副教授	北京理工大学
	徐　安	教授	上海同济大学
	赵　欢	副教授	湖南大学
	胡学钢	教授	合肥工业大学
	林福宗	教授	清华大学
	王家厥	教授	清华大学
	郑　莉	教授	清华大学
	朱淼良	教授	浙江大学
	刁成嘉	副教授	南开大学
	林和平	教授	东北师范大学
	孙铁利	教授	东北师范大学
	温子梅	讲师	广东教育学院
	吕国英	副教授	山西大学
	张广州	讲师	沈阳大学
	何新华	教授	装甲学院
	邱仲潘	副教授	厦门大学
	曾春平	副教授	第二航空学院
	姬东耀	教授	中科院计算所
	喻　飞	博士	浙江大学
	徐建华	总编	北京希望电子出版社
	郑明红	副总编	北京希望电子出版社
	韩素华	编辑室主任	北京希望电子出版社

总　序

21世纪挑战与机遇并存，没有足够的知识储备必将被时代所抛弃。中国IT教育产业竞争日趋激烈，用户需求凸现个性，行业发展更需要理性。未来5年IT行业将以18%的速度连续增长，将引发IT产业新的发展高潮。实现信息产业大国的目标，应该依赖教育，要圆信息产业强国的梦想，依然要寄托于教育，IT教育事业任重道远，其产业也正面临着机遇与挑战。

我国的计算机教学长久以来一直重原理、轻应用。高等院校的计算机教学机制和教材对计算机本身的认识都存在误区。要改革高校计算机教学，教材改革是重要方面，用计算机教材的改革促进基础教育的改革势在必行。

一本好书，是人生前进的阶梯；一套好教材，是教学成功的保证。为缓解计算机技术飞速发展与计算机教材滞后落伍的矛盾，我们通过调查多所院校的师生，并多次研讨，根据读者认识规律，开创出一种全新的方式，打破过去介绍原理——理论推导——举例说明的样式，增加实用操作性，通过上机实验与课上内容结合来增强可读性，用通俗易懂的语言和例子说明复杂概念。

本套教材的特点一是“精”，精选教学内容；二是“新”，捕捉最新资讯；三是“特”，配备电子课件，力争达到基础性、先进性、全面性、典型性和可操作性的最大统一。

为保证教材质量，我们同时聘请了一批学术水平较高的知名专家、教授作为本套教材的主审和编委。全套教材包括必修课教材20多种，选修课教材和学习配套用书10余种，基本上涵盖了目前高等院校（含高等职业技术学院、高等专科学校、成人高等学校）计算机科学与技术专业所必修或选修的内容。各种教材编写时既注意到内容上的连贯性，又保证了教学上的相对独立性。

本套教材在内容的组织上，大胆汲取当今计算机领域最新技术，摒弃了传统教材中陈旧过时的内容。这些变化在各本教材中都得到了不同程度的体现。本套教材编写时既参照了教育部有关计算机科学与技术专业的教学要求，又参考了“程序员考试大纲”和“全国计算机水平等级考试大纲”的内容，因此既适合作为高等学校计算机科学与技术专业教材，也可作为计算机等级考试学习用书。

考虑到各校教学特点和计算机设备条件，我们本着“学以致用”的理念，在本套教材编写中自始至终贯彻“由浅入深，理论联系实际”的原则，以阐明要义为主，辅之以必要的例题、习题和上机实习，能够使学生尽快领悟和掌握。

在本套教材编写过程中，作者们付出了艰辛的劳动，教材编委会的各位专家、教授对本套教材进行了认真的审定和悉心地指导。书中参考、借鉴了国内外同类教材和专著，在此一并表示感谢。

我们希望更多的优秀教师参与到教材建设中来，真诚希望广大教师、学生与读者朋友在使用本套教材过程中提出宝贵意见和建议。

若有投稿或建议，请发至本丛书出版者电子邮件：textbook@bhp.com.cn

21世纪高等院校计算机教材编委会

前　言

目前，各行业各领域都与计算机建立了紧密的联系，并由此产生开发各种应用软件的需求。有关数据结构和算法方面的研究是计算机科学与工程的基础性研究之一，掌握该领域的知识对于我们利用计算机资源高效地开发计算机程序非常重要。因为一个高效的程序既需要“编程小技巧”，更需要合理的数据组织和清晰高效的算法，这也正是计算机科学领域里数据结构与算法设计所研究的主要内容。通过对数据结构与算法设计的系统学习与研究，理解和掌握算法设计的主要方法，培养对算法的计算复杂性进行正确分析的能力，这对系统软件和应用软件研究与开发的科技工作者来说是非常重要和必不可少的。

为了适应21世纪对计算机类人才的需要，结合我国高等学校教育工作的现状，立足于更新教学内容和教学方法，我们编写了本书。书中以基本数据结构和算法设计策略为知识单元，系统地介绍了数据结构的知识与应用、计算机算法的设计方法与分析技巧，同时为了减少读者在学习“数据结构”时的恐惧心理，每个算法均附有详细的注释和程序清单。

本书共12章。第1章主要介绍了数据结构、抽象数据类型和算法的一些基本概念以及算法复杂性分析；第2章至第6章分别介绍了线性表、复杂的高级链表、栈、队列、数组、广义表、串等简单的数据结构；第7章介绍了一些递归的技巧和方法以及经典的递归问题的解决方法和策略；第8章和第9章介绍了复杂的数据结构、树和二叉树以及图；第10章和第11章介绍了常见的查找和排序算法；第12章介绍了常见的几种文件结构。

本书第1章、第12章由杨喜权老师编写，其余章节由张勇老师编写。参与本书编写工作的还有刘君义老师。

由于作者水平有限，书中错误在所难免，恳请广大读者给予批评指正。

编　者

目　录

第1章 概 述

1.1 数据结构的概念

自从1946年第一台电子计算机诞生以来，程序设计一直是计算机科技工作者研究的主题。数据结构正是伴随着程序设计的发展而逐渐兴起的一门学科。然而在可视化程序设计平台广泛流行和应用的今天，程序设计已经不再是一件神秘和专业性的工作。很多非计算机专业的人员都可以亲自动手设计应用程序，这似乎给人一种错觉,只要掌握了一门可视化程序设计语言，人人都可以成为编程高手。但实际上并非如此，要想成为一个熟练的专业化程序设计人员，至少应该具备3个条件：一是能够熟练选择和设计数据结构和算法；二是能够熟练掌握一门程序设计语言；三是熟知应用领域的相关知识。其中后两个条件通过努力比较容易实现，而第一个条件则需要花费相当的时间和精力才能达到，它也是区分一个程序设计人员水平高低的重要标志。

这样看来，数据结构看上去好像是一门深奥的学科。实际上，在学习这门课程之前学习程序设计的过程中，在不知不觉中已经运用一些数据结构的知识来解决了一些问题，只不过那时用到的数据结构非常简单。如下面的例子：

试用C语言设计求10个数的平均数的程序：98,100,126,50,78,134,147,111,89,73。

请看下面的两个程序：

```
void main()
{   int s0,s1,s2,s3,s4,s5,s6,s7,s8,s9;
    int sum,average;
    s0=98;s1=100;s2=126;s3=50;s4=78;
    s5=134;s6=147;s7=111;s8=89;s9=73;
    average=(s0+s1+s2+s3+s4+s5+s6+s7+s8+s9)/10;
    printf("输出平均数:%d\n",average);
}
```

这是一个非常简单的C语言程序，每个读者都能不费吹灰之力读懂，但是若你所想到的设计方法如此，那么你可要加强一下程序设计的能力。因为上面的程序使用了10个内存变量来存储10个数，这种设计方法的可扩充性不好。事实上，完全可以用数组来保存这10个数，程序如下：

```
void main()
{   int s[10]={98,100,126,50,78,134,147,111,89,73};
    int i,average;
    for(i=0;i<10;i++)
    sum+=s[i];
    average=sum/10;
    printf("输出平均数：%d\n",average);
}
```

在上面的程序 1 和程序 2 中，读者会认为程序 2 比较好。事实上，这是因为在程序 2 中使用比较好的数据结构。虽然两个程序都可以正确地解决实际问题，但是采用了不同的数据结构就会使程序具有不同的效率，而且在现今绝大多数应用程序都需要使用各种各样的数据结构及算法，缺少数据结构和算法的深厚功底，很难设计出高水平的具有专业水准的应用程序。这就是为何需要学习数据结构的重要原因。

那么究竟什么是数据结构呢？在回答这个问题之前，先阐述一下与数据结构相关的基本概念。

（1）数据　数据是描述客观事物的信息的符号，是计算机系统可加工处理的“原材料”。数据是一个广义的概念，既可以指普通的数据（可参加算术运算），例如整数和实数，也可以指符号（源程序，商品名称等）或数字化了的声音、图像、图形等。

（2）数据元素及数据项　数据元素也称为记录，指能独立完整地描述问题世界中的最小数据单位。而构成数据元素的不可分割的数据单位称为数据项。例如：在学生学籍系统中，每个学生记录就是一个数据元素，而学生的姓名、年龄等项目是数据项。数据元素是讨论数据结构时涉及的最小数据单位，其中的数据项一般不予参考。

（3）数据对象　性质相同的数据元素的集合称为数据对象。例如：所有学生记录的集合是该问题世界的一个数据对象。

（4）数据类型　现实世界中的数据是多种多样的，不同的用途需要不同种类的数据，而不同种类的数据对应不同种类的使用方法（操作），因此有必要对数据进行分类。在计算机中以操作种类划分数据类型，即将数据类型定义为：一个值的集合和定义在这个值上的一组操作的总称。

（5）数据结构　上面描述的数据元素往往不是孤立的，而是存在着一定的关系，把数据元素之间的这种关系称为结构。进一步地，把相互之间存在着一定关系的数据元素的集合及定义在其上的基本操作（运算）称为数据结构。

根据数据元素间关系的不同特性、通常有下面 4 种结构：

（1）集合结构　在集合结构中，数据元素的关系是“属于同一集合”。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。

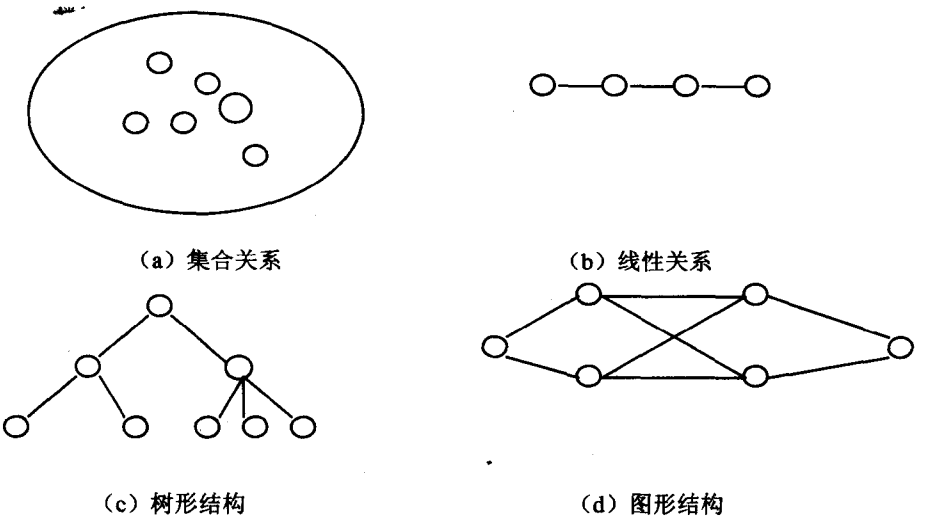
（3）树形结构　该结构的数据元素之间存在着“一对多”的关系。

（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，4 种结构如图 1-1 所示。

从上面所介绍的数据结构的概念可知，一个数据结构有两个要素：一是数据元素的集合，另一个是关系的集合。如果不考虑定义在数据结构上的操作，则数据结构也可以借助集合论术语定义为：数据结构是一个二元组

Data_structure=(D,S)

其中，D 是数据元素的有限集，S 是 D 上关系的有限集。



（a）集合关系　　（b）线性关系

（c）树形结构　　（d）图形结构

图 1-1　数据结构的分类

在这个定义中，数据元素之间的关系采用集合论中关系的形式化描述方法来定义，采用形式化方法表示关系的优点是描述精确，易于使用数学方法研究，但这种方法有时显得很繁琐，不利于被初学者理解和掌握。

实际上，上面介绍的数据结构的概念指的是数据的逻辑结构。在这里，基本操作刻画了数据元素之间的逻辑关系，反映了它们的性质和功能，这种逻辑关系可以看作是从具体问题中抽象出来的数学模型，它与实际的存储无关。而实际上，任何数据均要存储到计算机内存中，而数据在计算机内的存储安排，称为数据的物理结构或称为数据的存储结构。具体内容将在下一节介绍。

1.2　数据结构的存储

1.2.1　存储器表示

归根到底，数据要存储到计算机的存储器中。当代计算机的存储器，从使用方式来讲，分为内部存储器（简称内存或主存）和外部存储器（简称外存或辅存）。内存由多个存储单元构成，每个存储单元为一个基本的存储单位。存储单元用地址标识，每个存储单元都有一个惟一的地址，访问存储单元时，必须指明要访问的存储单元的地址。存储单元是最小的访问单位。各存储单元是连续编址的。若干地址连续的存储单元称为一个存储区。内存数据是通过 CPU 指令按存储单元的地址直接被访问的。

外存是内存的辅助，CPU 指令一般不直接控制外存，而将其作为外部设备访问，外存的数据一般是通过外设访问指令装入后接受 CPU 的操作。

本书数据结构的存储问题除特殊声明外，都针对内存。

1.2.2　数据结构的映像

数据元素之间的关系在计算机中的存储方式，即是所谓数据结构的映像，是数据结构

的另一种表示方法，也称为数据结构的存储方式或物理结构。前面介绍了数据结构的逻辑结构，那么有的读者会想：为什么要有逻辑结构和物理结构之分呢？实际上，逻辑结构反映的是数据元素逻辑上的关系。例如，在学生档案系统中，一个学生记录与另一个学生记录在进行程序设计（例如，输入时）时相邻的，即逻辑上是相邻的。而实际上，这两个学生记录在物理位置上不一定相邻，这取决于你采用什么样的存储结构。不同的存储结构对具有同样的逻辑结构的数据来说，各种操作的执行效率可能是不一样的。因此说，数据的逻辑结构和物理结构是密切相关的两个概念。作为数据结构的初学者，首先应该弄清楚这两个关键的概念，这对后面内容的学习是非常有帮助的。

但这里需要强调的是，对内存的使用，最直接的方式是通过具体机器的机器指令来实现，但不采用这种方式，原因主要有两种：一是在实际的计算机应用中，大多数不直接使用机器语言，而主要使用高级程序设计语言（如 C，C++，Java，PASCAL，BASIC 等）；二是机器语言相当繁琐，且可移植性差，会使人陷入到一些不必要的细节中。因此本书在高级语言的层次上讨论数据结构的操作，但是不能直接以内存地址来描述存储结构，好在可以借助高级语言中提供的数据类型来描述它（例如，数组、指针类型等）。但不管用什么数据类型来描述，要将一个逻辑上的数据结构存储到计算机中必须满足以下两点：

（1）内容存储　数据结构中的各数据元素的内容，都分别存储在一个独立的可访问的存储区中。

（2）关系存储　数据元素的存放方式方法，必须能显式或隐式地体现数据元素间的逻辑关系。

这两点是存储映像所应有的必要条件。在满足上述必要条件的基础上，存储映像还应考虑存储使用效率及数据结构的操作实现的方便性等。

基本的存储结构有顺序方法、链接方法、散列方法、索引方法。实际的存储映像方法，也可能是这些基本方法的复杂组合。

1.2.3　数据结构的几种常见存储方式

1. 顺序映像

这是一种面向线性关系的存储方法。对于线性数据结构，可将其数据元素按相应的线性关系的前后次序，存储在物理存储器中，使得数据元素在此线性关系下的逻辑次序与它们在存储器中的存放次序一致。这种存储方式称为顺序存储。

由于顺序存储方式下数据元素的逻辑次序与物理次序一致，所以数据元素间的线性关系由它们的存储次序体现，而不需要专门存储。这种方法主要面向线性结构，但不局限于此。若某结构中存在一种线性关系，而通过此线性关系就可以确定元素关系，则可使用这种方法。如多维数组、顺序二叉树等结构的存储。为了能使存储次序表达逻辑次序，在存储器中，任意相邻两数据元素之间的存储单元数目应相等。

2. 链式存储

链式存储的每个数据元素的存储区分为两部分：第一部分为数据区，存储元素的内容；第二部分为指针区，存放该数据元素与其他数据元素之间的关系信息，这种关系信息一般为地址。对于线性结构，指针区中可以只设一个地址；对于非线性结构，可能需要多个地

址。数据元素的存储区之间，可以是连续的，也可以是不连续的。

链式方法具有很大的灵活性，适合于大多数的数据结构。与顺序方法相比，它的存储空间开销较大（增加了指针区），但由于各数据元素的存储区不要求是连续排列的，故对内存空闲区分布的要求不高，很适合动态存储管理。

3. 索引结构

索引结构主要针对集合和线性结构，面向检索操作。它主要是在数据结构的存储区（称数据区）外，增加一个或若干个索引区。

索引区本身也是个线性表或其他数据结构，其中每个元素用于记录数据区中一个（对于稠密索引）或一组（对于非稠密索引）元素的存储位置。

索引存储并不强调对关系的存储，而主要对数据内容，所以，一般只适合集合结构或线性结构。

4. 散列存储

散列存储（也称杂凑法）是一种按元素内容存储的方法。其基本思想如下：

（1）设置一个散列函数，其形式为“元素内容->地址”；规定元素内容到存储地址的映射。

（2）存储时，通过散列函数求出元素的存储地址，按此地址存储。

（3）读取时，通过散列函数求出元素的存储地址，按此地址读取。

与索引存储类似，散列存储也是面向内容的存储，不适合存储复杂数据结构。以上几种存储方式的细节问题将要在后面章节介绍。

1.3　数据结构课程研究的内容

数据结构与数学、计算机硬件和软件有十分密切的关系。数据结构是介于数学、计算机硬件和计算机软件之间的一门计算机科学与技术专业的核心课程，是高级程序设计语言、编译原理、操作系统、数据库、人工智能等课程的基础，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。

数据结构课程集中讨论软件开发过程中的设计阶段，同时设计编码和分析阶段的若干基本问题。此外，为了构造出好的数据结构及其实现，还需考虑数据结构及其实现的评价和选择。因此，数据结构的内容包括两个层次的 5 个“要素”，如图 1-2 所示。

数据结构的核心技术是分解和抽象。通过对问题的抽象，舍弃数据元素的具体内容，就得到逻辑结构。类似地，通过分解，将处理要求划分成各种功能，再通过抽象舍弃实际细节，就得到运算的定义。上述两个方面将问题变换成数据结构。这是一个从具体(即具体问题)到抽象（即数据结构）的过程。然后，通过增加对实现细节的考虑进一步得到存储结构和实现运算，从而完成设计任务。这是一个从抽象（即数据结构）到具体（即具体实现）的过程。熟练地掌握这两个过程是数据结构课程在专业技能培养方面的基本目标。

层次 \ 方面	数据表示	数据处理
抽象	逻辑结构	基本运算
实现	存储结构	算法
评价	不同数据结构的比较及算法分析	

图 1-2　数据结构的课程体系

数据结构作为一门独立的课程在国外是从 1968 年才开始的。但在此之前其有关内容已散见于编译原理及操作系统课程之中。20 世纪 60 年代中期，美国的一些大学开始设立有关课程，但当时的课程名称并不叫数据结构。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》是第一本较系统地阐述数据的逻辑结构及存储结构及其操作的著作。从 20 世纪 60 年代末到 70 年代初，出现了大型程序，软件也相对独立。结构程序设计成为程序设计方法学的主要内容，人们越来越重视数据结构，从 20 世纪 70 年代中期到 80 年代，各种版本的数据结构著作相继出现。目前，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，从抽象数据类型和面向对象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。

1.4　C 语言与数据结构

1.4.1　数据类型及抽象数据类型

数据类型是与数据结构密切相关的概念，它最早出现在高级程序语言中，用以刻画操作对象的特性。同一类数据的全体称为一个数据类型。在高级程序设计语言中，数据类型用来说明数据在数据分类中的归属。例如，高级程序设计语言中，每个变量、常量或表达式都有一个它所属的数据类型。数据类型显式或隐式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，以及在这些值上的操作。因此数据类型是一个值的集合和定义在这个值集上的一组操作的总称。为了解决问题需要，高级程序设计语言定义了一系列的数据类型，不同的高级语言所定义的数据类型不尽相同。基本上可分为两种类型：一是原子类型，原子类型的值是不可分解的。例如，C 语言中的基本类型（整型、实型、字符型和枚举型）、指针类型和空类型。另一种是构造类型，在构造数据类型中，允许成分数据本身是可以分解的。例如，C 语言中的结构体类型。

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，而高级语言提供的数据类型，其操作通过

编译器或解释器转化成低层，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，即将一切用户不必要了解的细节封闭在类型中。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，也不需要了解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。

那么数据类型与要讨论的数据结构有什么关系呢？前面介绍过，可以借用高级程序语言中提供的数据类型来描述存储结构。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，其他的象图、树型数据结构也可以用复杂的构造数据类型来描述，这些内容将在后面章节介绍。

抽象数据类型是对数据类型的进一步抽象，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。

另一方面，抽象数据类型的范围很广，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 3 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 3 种类型：

（1）原子类型　属于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，已有的固有类型足已满足需求。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。

（2）固定聚合类型　属于该类型的变量，其值由确定数目的成分按某种结构组成，例如，复数是由两个实数依确定的次序关系构成。

（3）可变聚合类型　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，后两种类型可统称为结构类型。

和数据结构的形式定义相对应，抽象数据类型可用以下三元组表示：

（D，S，P）

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。

1.4.2 C语言的数据类型

由于本书在高级程序设计语言的虚拟层次上讨论数据结构及其算法，主要是面向读者，故采用读者非常熟悉的C语言作为描述工具，有时也用一些伪码描述含抽象操作的抽象算法。这使得数据结构与算法的描述和讨论更加精确和清晰。下面先对C语言的数据类型作一下总结。

把整型和实型合称为“数值型”，把数值型和字符型合称为“基本数据类型”。C语言根据数据加工处理的特征，还设有其他复杂的数据类型，具体如图1-3所示。

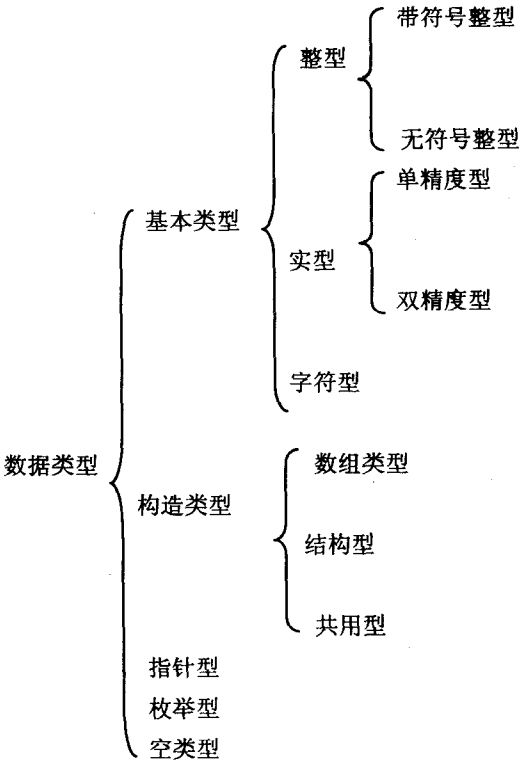


图1-3 C语言的数据类型

构造类型指由若干个相关的基本类型数据组合在一起形成的一种复杂的数据类型。数组是由相同类型数据组合而成的，例如，若干个人的年龄组合在一起，就是一个整型数组；若干个人的基本工资、职务工资、奖金组合在一起，就是实型数组。结构型是由不同数据类型组合而成的，例如，一个人的姓名（字符串）、性别（字符型）、年龄（整型）、基本工资（实型）也可以组合在一起，构成一个结构型数据。如果若干个数据不同时使用，就可以让它们占用相同的内存区域，以便节省内存，这些数据组合在一起就是“共用型”，共用型中的数据可以是相同类型的，也可以是不同类型的。

指针型是一种简单的数据类型，它是用来表示内存地址的。指针类型的数据可以表示基本类型数据的地址，也可以表示结构类型数据的首地址和其中某个具体数据的地址。例如，存放一个人年龄的内存的地址、存放某数组的首地址等都可以用指针型数据来表示。