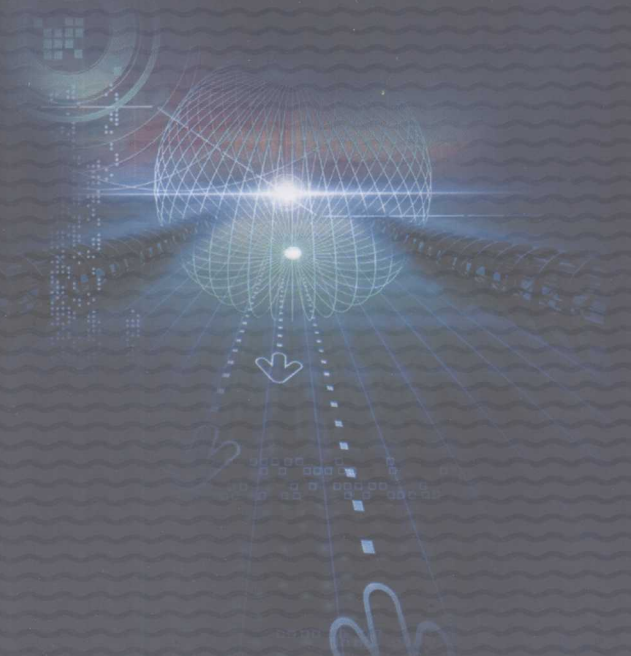
免费提供电子教案

高等院校规划教材

计算机科学与技术系列

数据结构（C++版）

吴小平　马桂媛　编著



机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材·计算机科学与技术系列

数据结构（C++版）

吴小平　马桂媛　编著

机械工业出版社

本书共分 11 章。第 1 章介绍数据结构和算法的概念及相关术语。第 2～5 章介绍线性结构。第 6～8 章介绍非线性结构。第 9 章和第 10 章分别介绍了查找和排序。第 11 章介绍了一些常用文件。各章内容都有相对独立的部分，以便针对不同专业或不同层次的需要组织教学。

本书可作为高等院校计算机类专业和相关专业数据结构课程的教材，也可以供从事计算机应用工作的工程技术人员参考。

图书在版编目（CIP）数据

数据结构（C++版）/ 吴小平，马桂媛编著. —北京：机械工业出版社，2009.8
（高等院校规划教材·计算机科学与技术系列）
ISBN 978-7-111-27794-1

Ⅰ. 数… Ⅱ. ①吴…②马… 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. TP311.12 TP312

中国版本图书馆 CIP 数据核字（2009）第 122186 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：陈 皓 常建丽
责任印制：洪汉军

北京瑞德印刷有限公司印刷（三河市胜利装订厂装订）

2009 年 8 月·第 1 版第 1 次印刷
184mm×260mm·17.25 印张·424 千字
0001－3000 册
标准书号：ISBN 978-7-111-27794-1
定价：31.00 元

凡购本书，如有缺页，倒页，脱页，由本社发行部调换
销售服务热线电话：（010）68326294 68993821
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）88379753 88379739
封面无防伪标均为盗版

出 版 说 明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21 世纪高等院校计算机教材系列”、“21 世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21 世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。另外，本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前　言

“数据结构”是计算机及相关专业的一门重要的专业基础课，也是部分理工科专业的选修课程。数据结构课程的主要任务是，研究现实世界中各种数据对象的逻辑结构在计算机中的存储表示以及在不同存储结构上的相应算法，初步掌握算法的时间分析技术和空间分析技术，并通过实际编程训练为开发复杂程序打下良好基础。

数据结构广泛应用于计算机的各个领域。在计算机操作系统、计算机图形学、软件工程、多媒体技术、计算机网络、编译原理、数据库原理和技术、计算机辅助设计、人工智能等课程中普遍使用数据结构的理论和方法来描述和解决问题。因此，学好数据结构课程是学好其他后续计算机课程，特别是计算机软件方面课程的基础，同时也为今后走上工作岗位从事大型计算机软件开发奠定了基础。

本书共分 11 章。第 1 章主要介绍数据结构的基本概念和相关术语，并简单介绍了进行算法描述和算法分析的基本方法；第 2～5 章介绍线性结构（线性表、栈、队列、数组、串）的逻辑特征，存储表示方法和基本操作的实现算法，以及一些应用实例；第 6～8 章介绍非线性结构（广义表、树、图）的逻辑特征，存储表示方法和基本操作的实现算法，以及应用实例；第 9 章和第 10 章分别介绍非数值计算领域中的两种非常重要的操作，即查找和排序；第 11 章介绍一些常用文件的逻辑结构和物理结构特征。各章内容都有相对独立的部分，以便针对不同专业或不同层次的需要组织教学。

本书采用简化的 C++程序设计语言作为算法的描述语言，与目前多数学校采用 C++语言作为教学语言相衔接。但考虑到本书的学习对象主要是低年级学生，因此除了使用类模板描述数据类型以外，只使用了 C++相对简单的内容来描述算法，以便于学生理解和接受。

本书的所有算法都已在 VC++ 6.0 环境下上机调试通过，但为了节省篇幅，所有算法都以函数形式给出。若读者要运行这些算法，除了需要编写主函数来调用它们以外，还必须包含头文件和添加必要的变量说明。本书选用的例题都针对特定的数据结构，便于学生学习使用数据结构原理解决实际问题的方法。

本书建议理论讲授 50～60 个学时，上机实践 20～30 个学时。在实际教学过程中，授课教师可以根据学生的专业特点和不同层次进行适当增删。

由于作者水平有限，书中难免有不妥之处，敬请读者批评指正。

编　者

目　录

第1章 绪 论

随着计算机产业的发展，特别是计算机技术的高速发展和微型计算机的日益普及，计算机系统无论在硬件方面，还是在软件方面都远远超过了人们对它的预料，它已广泛渗透到人类社会的各个领域。现在的计算机已不再局限于处理纯数值计算问题，而更多地用于控制、管理以及数据处理等领域。与此相对应，计算机处理的对象也由纯粹的数值发展到诸如字符、表格、声音、图像、视频等复杂且具有结构的非数值数据。因此，在相应程序的设计过程中，必须研究数据的特性和数据之间存在的内在关系，才能设计出优良的程序，这正是学习本课程的基本目的。“数据结构”是一门综合性的计算机专业基础课，是介于数学、计算机硬件和计算机软件之间的一门核心课程，其内容不仅是一般程序设计（特别是非数值计算程序设计）的基础，而且是设计编译程序、操作系统、数据库系统以及其他复杂程序的重要基础。

1.1 数据结构的研究内容

一般来说，使用计算机解决问题大致需要以下几个步骤：首先从具体问题中抽象出一个适当的数学模型，然后设计一个解此数学模型的算法，最后编写出程序，进行测试和修改，直至得到最终解答。在解决问题的过程中，寻求数学模型的实质是通过分析，从问题中提取操作的对象，并找到这些对象之间的关系，然后用数学语言加以描述。然而，对非数值计算问题，往往很难用一个或几个数学方程来描述对象之间的关系，只能采用数据结构方法进行描述。

【例 1-1】 学生管理系统。

学生管理系统中记录了学生姓名、学号、性别、年龄、民族、专业、班级等基本信息。对于学生的这些基本信息，最方便的管理方法是把它们组织成一个表（见表 1-1），其中每个学生的基本信息构成一个记录，占一行。学生管理系统中其他信息之间的关系亦用表来进行描述，如学生成绩表 1-2。显然，这些类似的表就构成了学生管理系统最基本的数学模型。

表 1-1 学生基本情况

姓 名	学 号	性 别	年 龄	民 族	专 业	班 级
张勇	20050401	男	20	汉族	计算机应用	200504
何力	20050403	男	21	汉族	计算机应用	200504
唐开宇	20050404	男	19	汉族	计算机应用	200504
陈力伟	20060407	男	20	回族	计算机应用	200504
…	…	…	…	…	…	…

观察表 1-1 和表 1-2，可发现它们有以下特征：

1）行的数量是有限的。

2）行之间构成先后顺序关系，除第一行外，每一行都有唯一的直接前驱，除最后一行外，每一行都有唯一的直接后继。

表中行之间的这种关系称为线性关系。这类数学模型可称为线性的数据结构。它是一种重要的数据结构。

表 1-2　学生成绩

姓　名	课 程 名	学　期	成　绩
唐开宇	高等数学	1	80
唐开宇	线性代数	3	75
何力	高等数学	1	77
何力	线性代数	3	69
…	…	…	…

【例 1-2】 计算机和人对弈的问题。

计算机之所以能和人对弈是因为人们事先将对弈策略输入了计算机。对弈过程在一定规律下随机进行。为了使计算机能灵活对弈就必须将对弈过程中可能发生的情况和相应的对策考虑周全。一个“好”的棋手在对弈时不仅要掌握棋盘的当前状态，还应能预测和主导棋局的发展趋势。在对弈问题中，计算机操作的对象是称为格局的棋盘状态。图 1-1 为一个井字棋格局，格局之间的关系由比赛规则决定，通常这个关系不是线性的，这是因为由一个棋盘格局可以派生出几个格局。例如，图 1-1 所示的格局随棋手落子位置的不同可以派生出 5 个格局，而从一个新的格局又可派生出 4 个格局……格局之间的关系表现为一种层次关系。这种层次关系称为树形关系，它是某些非数值计算问题的数学模型，也是一种重要的数据结构。

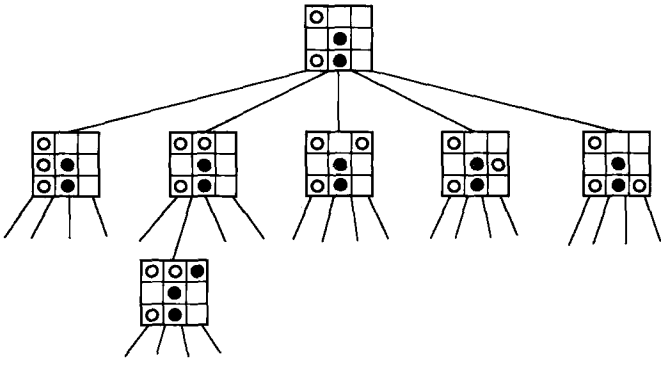


图 1-1　井字棋格局

【例 1-3】 城市间的交通网络。

图 1-2 示意了我国一些主要城市之间的交通网络情况。其中，椭圆代表城市，称为顶点，连接两椭圆之间的线段代表两城市之间的交通线路，线段上的数字代表对应交通线路的长度。从一个城市到另一个城市往往可以通过多条线路抵达。该图顶点之间的关系显然既不是线性关系，也不是树形关系，而呈现出一种网状的关系。我们将这种关系称为图状结构或网状结构，它是许多非数值计算问题的数学模型，也是一种重要的数据结构。

以上 3 个例子表明，描述现实世界中许多问题的数学模型往往不是数学方程，而是诸如

表、树和图之类的数据结构。“数据结构”正是以研究和解决这些问题为主要目的。因此，简单说来，“数据结构”是一门研究在程序设计问题中计算机的操作对象、它们之间的关系以及操作实现方法等的学科。

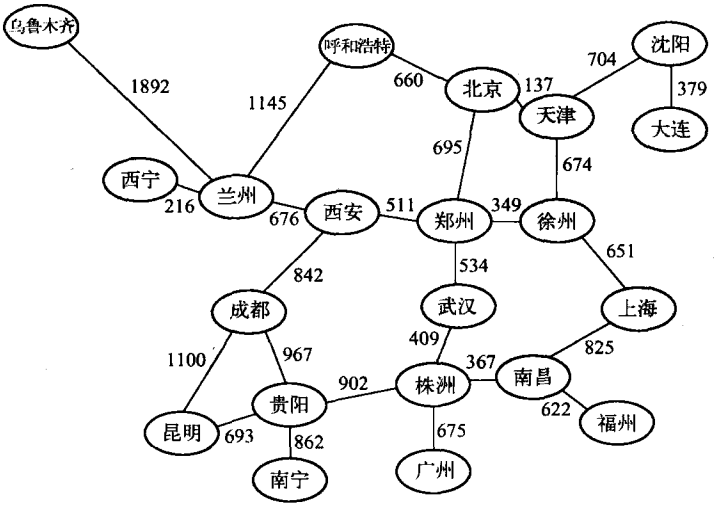


图 1-2　我国一些主要城市之间的交通网络示意图

1.2　数据结构的基本概念和相关术语

数据结构包含数据的逻辑结构、数据的存储结构和对数据进行的操作 3 方面基本内容。这些基本内容涉及一些基本概念和术语，在介绍数据结构的相关内容之前，有必要先把这些基本概念和术语的确切含义搞清楚。

1．数据

数据是对客观事物的符号表示。在计算机科学中，数据是指一切能够输入到计算机中，并且能够被计算机程序处理的符号的总称。简言之，数据就是计算机加工处理的“原料”，是信息的载体。

计算机所涉及的数据总体上可以分为两类：一类是数值数据，包括整数、实数和复数等，它们主要用于工程和科学计算以及商务事务处理；另一类是非数值数据，包括字符、字符串以及文字、图形、语言、声音等，它们多用于控制、管理和数据处理等领域。

数据的含义十分广泛，在不同的场合中有不同的含义。例如，在数值计算问题中，计算机处理的数据主要是整数和实数；在文字处理程序中，计算机处理的数据大多是一些符号串；而在一些控制过程问题中，数据又可能是某种信号。

2．数据元素

数据元素是数据中相对独立的基本单位，是数据中的一个个体。在计算机程序中通常把数据元素作为一个整体进行考虑和处理。数据元素既可简单，也可复杂。简单的数据元素只有一个数据项，而复杂的数据元素由若干数据项组成。例如，在例 1-1 的表中，一个学生的信息是一个数据元素，它由姓名、学号、性别、年龄、民族、专业、班级等数据项组成。数据项是数据的不可分割的最小单位。

3．数据对象

数据对象是性质相同的数据元素的集合，是数据的一个子集。例如，自然数的数据对象是集合{1，2，3，…}，该集合中的每一个数据元素都是一个自然数。而由 26 个大写英文字母组成的数据对象则是集合{'A'，'B', 'C'，…，'Z'}。

4．数据结构

在客观世界中，任何事物和活动都不是孤立存在的，在一定程度上都会相互影响，相互联系，甚至相互制约。这表明现实世界中的数据元素之间往往存在着一种或多种关系，这些关系又称为结构。根据数据元素之间关系的不同特性，通常有以下 4 类基本结构：

1）集合，即数据对象中的数据元素之间除了具有“同属于一个集合”的关系外，别无其他关系。

2）线性结构，即数据对象中的数据元素之间存在一对一的先后顺序关系。

3）树结构，即数据对象中的数据元素之间存在一对多的层次关系。

4）图结构，即数据对象中的数据元素之间存在多对多的任意关系。4 类基本结构关系的示意图如图 1-3 所示。

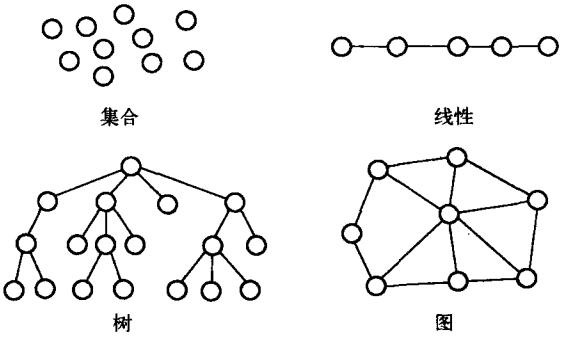


图 1-3　4 类基本结构关系的示意图

人们在处理数据时，往往既要考虑数据元素本身的特性，又要考虑数据元素之间的结构特点。从这个意义上可以将数据结构定义为相互之间存在一种或多种特定关系的数据元素的集合，或者说是具有结构的数据元素的集合。

数据结构可以形式化地描述为：数据结构是一个二元组

$$Data\_Structure=(D,R)$$

其中，D 是数据元素的有限集；R 是 D 上关系的集合。

【例 1-4】 图 1-4 所示的树结构可用以下二元组表示:

$$Tree=(D,R)$$

其中，$D=\{d_1,d_2,d_3,d_4,d_5,d_6,d_7,d_8\}$

$R=\{<d_1,d_2>,<d_1,d_3>,<d_1,d_4>,<d_1,d_5>,<d_3,d_6>,<d_3,d_7>,<d_3,d_8>\}$

【例 1-5】 图 1-5 所示的图结构可用以下二元组表示:

$$Graph=(D,R)$$

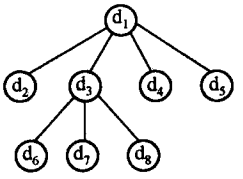


图 1-4　树结构的示例

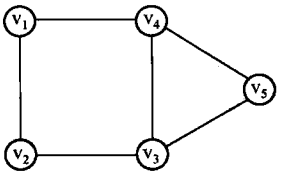


图 1-5　图结构的示例

其中，$D=\{v_1,v_2,v_3,v_4,v_5\}$

$$R=\{(v_1,v_2),(v_1,v_4),(v_2,v_3),(v_3,v_4),(v_3,v_5),(v_4,v_5)\}$$

上述数据结构的定义仅是对操作对象的一种数学描述，即从操作对象抽象出来的数学模型。数据结构定义中的“关系”是指数据元素之间存在的逻辑关系，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对它实现操作，因此还需研究数据结构在存储器中的表示。

数据结构在计算机中的表示（或称存储映像）称为数据的物理结构。物理结构又称为存储结构，它包括数据元素的表示以及关系的表示两个方面。在存储器中通常由若干二进制位组成一个位串来表示一个数据元素，这时的数据元素又简称为元素或结点。若数据元素包含若干个数据项，则每个数据项对应的位串又称为数据域。数据的存储结构通常有以下几种：

（1）顺序存储结构

顺序存储结构即数据元素按其逻辑顺序依次存放在存储器的一段连续存储单元中。例如，逻辑结构（50，36，2，46，44，88），若按图 1-6 方式存放，则采用了顺序存储结构表示。

存储器地址	0300	0302	0304	0306	0308	030A	030C	030E	0310
	···	50	36	2	46	44	88	···	

图 1-6　顺序存储结构

由于顺序存储结构中元素的物理位置顺序与数据元素之间的逻辑顺序一致，因此元素的物理位置顺序直接表示了数据元素之间的逻辑关系。

（2）链式存储结构

链式存储结构即数据元素在计算机的一段存储空间内随意存放在任意位置上。在这种情况下，元素的物理位置顺序与数据元素之间的逻辑关系没有对应关系。为了记录数据元素之间的逻辑关系，链式存储结构的每一个结点除了需要用一个数据域来存放数据元素自身的值以外，还必须至少定义一个地址域（指针域，简称指针）来存放数据元素之间的逻辑关系。例如，逻辑结构（50，36，2，46，44，88），可以按照图 1-7 示意的链式存储结构存放在存储器中。图中，每个结点的指针域存放的是当前元素直接后继在存储器中的地址。由于链式存储结构中的元素随意存放，元素之间的逻辑关系通过指针指示，因此要访问某个元素，必须从结构的第一个元素开始沿指针进行查找。

存储器地址	0330	0304	0308	030C	0310	0314	0318	031C	0320	0324	0328
		36		44		88	50	2	46		
		031C		0314		NULL	0304	0320	030C		

Head=0318

图 1-7　链式存储结构

（3）索引存储结构

索引存储结构即在存储器的一段空间内存储数据元素信息的同时，还为存储的元素建立了索引表。若每个元素在索引表中有一个索引项，则称为稠密索引（见图 1-8）。若每组元素在索引表中有一个索引项，则称为稀疏索引（见图 1-9）。索引项的一般形式是：（关键字，地址）。其中，关键字唯一地标识了一个数据元素，而地址则是该关键字对应数据元素（组）在存储器中的起始地址。

关键字（学号）	元素地址
023	
024	
030	
037	

学号	姓名	性别	年龄	民族	班级	专业
023	王力	男	20	汉	0501	计算机
024	周彬	女	19	汉	0501	计算机
030	赵川	男	20	汉	0501	计算机
037	吴宇	男	21	汉	0502	计算机

图 1-8　稠密索引示意图

关键字（学号）	元素地址
030	
039	

学号	姓名	性别	年龄	民族	班级	专业
023	王力	男	20	汉	0501	计算机
024	周彬	女	19	汉	0501	计算机
030	赵川	男	20	汉	0501	计算机
037	吴宇	男	21	汉	0502	计算机
038	刘奇	男	20	汉	0502	计算机
039	王鹏	男	19	汉	0502	计算机

图 1-9　稀疏索引示意图

（4）散列存储结构

散列存储结构即根据选择的散列（哈希）函数，将数据元素散列到存储器的相应位置上。使用散列方法存储元素，涉及到散列函数的选择和怎样解决地址冲突问题，这些在以后的章节中会详细讨论。

以上介绍的 4 种存储结构，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。对于一个逻辑结构，可以选择不同的存储结构。若逻辑结构相同但存储结构不同，则两种结构的性质差异很大，以致于人们用不同的数据结构名称来区分它们。例如，对于线性结构，若采用顺序存储映像，则称为顺序表；而若采用链式存储映像，则称为链表。

5．数据类型和抽象数据类型

数据类型是与数据结构密切相关的一个概念。在高级编程语言的层次上，可以借助数据类型来描述数据结构。一般高级编程语言中都提供了若干数据类型。例如，在 C++语言中有整型、实型、字符型、数组类型、结构体类型、类类型等数据类型。每一个变量、常量或表达式都属于一个确定的数据类型。数据类型显式或隐含地规定了程序执行期间内变量或表达式的可能取值范围，以及在这些值上允许进行的操作。例如，C++语言中的整型变量，其值为某个值域区间上的整数（值域区间大小依赖于不同机器），能进行的操作有“加”、“减”、“乘”、“除”和“取余”等。因此，数据类型的本质是一个值的集合和定义在这个值集上的

一组操作的总称。

根据“值”的不同特性，高级编程语言中的数据类型可分为两类。

1）非结构的原子类型，其值是不可分解的。C++语言中的整型、实型、字符型、指针类型、空类型等基本类型就属于原子类型。

2）结构类型，其值由若干成分按某种结构组成，因此是可以分解的。C++语言中的数组、结构体、类就属于结构类型。

数据结构在某种意义上可以被看成“一组具有相同结构的值”，而结构类型则可被看成由一种数据结构和定义在该结构上的一组操作组成，因此可以借助数据类型来描述数据结构。事实上，本书在介绍每一种数据结构时，都是通过一个抽象数据类型或者数据类型进行描述的。

数据类型是与计算机密切相关的概念。一个数据类型的具体特征与计算机系统有关。为了更好地描述数据类型的本质特征，可以把数据类型中与计算机无关的数学特性抽象出来，从而得到抽象数据类型（ADT）的概念。

抽象数据类型是指一个数学模型以及定义在该模型上的一组操作，即由数学意义上的一个值域和定义在该值域上的一组操作组成。抽象数据类型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。利用抽象数据类型更容易对一个数据类型的数学特征进行描述。例如，“整型”数据类型尽管随不同计算机系统上实现的方法不同，可以具有不同的特征，但它们具有相同的数学特性。因此，我们可以使用一个“整型”抽象数据类型来对其共有的数学特性进行描述。

本书在介绍每一个数据结构时，都把它描述成一个抽象数据类型。而一个抽象数据类型的软件模块通常包含定义、表示和实现 3 个部分。对抽象数据类型，本书通常按照以下方式进行描述：

```
ADT 抽象数据类型名{
    数据对象：<数据对象的定义>
    数据关系：<数据关系的定义>
    基本操作：<基本操作的定义>
    }ADT 抽象数据类型名
```

其中，数据对象和数据关系的定义用伪码或文字描述，基本操作按以下格式定义：

```
基本操作的函数名（参数表）
    操作说明
```

【例 1-6】 抽象数据类型“复数”可以按以下方式描述：

```
ADT Complex{
    数据对象：D={ c1,c2 | c1,c2 ∈ R(实数集) }
    数据关系：R={ <c1,c2> | c1 是复数的实部，c2 是复数的虚部 }
    基本操作：
        Create( x,y,&z );
          //生成一个以 x 为实部，以 y 为虚部的复数 z
        Add( z1,z2,&sum );
          //求复数 z1 和复数 z2 之和，结果存放在 sum 中
```