HZ BOOKS
华章教育

高等院校精品课程系列教材·省级

数据结构与算法

C语言版

第2版

徐凤生◎编著

精品课程

Data Structures and Algorithms in C

Second Edition

机械工业出版社
China Machine Press

高等院校精品课程系列教材·省级

数据结构与算法

C语言版

第2版

徐凤生 编著

Data Structures and Algorithms in C

Second Edition

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

数据结构与算法：C 语言版 / 徐凤生编著 .—2 版 . —北京：机械工业出版社，2014.9
（高等院校精品课程系列教材）

ISBN 978-7-111-47940-6

I. 数… II. 徐… III. ①数据结构 – 高等学校 – 教材 ②算法分析 – 高等学校 – 教材
③ C 语言 – 程序设计 – 高等学校 – 教材 IV. ① TP311.12 ② TP312

中国版本图书馆 CIP 数据核字（2014）第 211059 号

本书涵盖数据结构的基本概念，定义了线性表、栈、队列、串、数组、广义表、树和二叉树、图、查找、排序等各种结构的抽象数据类型，给出了相应操作的实现算法，并在最后一章给出了几个课程设计的实例。另一方面，本书采用 C 语言描述算法，并给出了各种算法的效率分析，以及这些结构在计算机科学及其他领域的应用。此外，每章后均配有典型例题、上机实验和习题。本书中的所有算法都在 VC++ 环境下调试通过。

本书在内容安排上突出由浅入深、循序渐进、通俗易懂的特点，算法分析透彻、讲解清晰、便于学生自学。为了激发学生的学习兴趣，培养学生解决实际问题的能力，书中融入了一些典型的应用实例，如命题公式真值表的求解算法、出栈序列的求解算法等。

本书可作为高等院校计算机及相关专业本科生的“数据结构”课程教材，也可供相关科技人员学习参考。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：李　燕
印　　刷：北京诚信伟业印刷有限公司　　版　　次：2014 年 10 月第 2 版第 1 次印刷
开　　本：185mm × 260mm　1/16　　印　　张：18
书　　号：ISBN 978-7-111-47940-6　　定　　价：35.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

前　言

“数据结构”是计算机程序设计的重要理论技术基础，是计算机学科的核心课程，也是计算机专业考研的必考课程，同时已成为其他理工科专业的热门课程。学好该课程，不仅对学习后续算法设计、数值分析、操作系统、编译原理等课程有很大帮助，而且在实际中有广泛的用途。

数据结构主要研究数据的各种组织形式以及建立在这些结构之上的各种运算的实现。它不仅为用计算机语言进行程序设计提供了方法性的理论指导，还在一个更高的层次上总结了程序设计的常用方法和常用技巧。

“数据结构”课程的特点是概念多、算法灵活和抽象性强。针对这种情况，我们在参考各种数据结构教材的基础上，结合作者多年的教学经验，编写了这本适用于普通高等院校计算机及相关专业本科生的数据结构教材。本书的编写突出了课程学科能力的培养，体现了“理论和应用”兼顾的教学改革理念。

本书分为11章：第1章绪论，介绍数据、数据结构、抽象数据类型等基本概念，特别是算法分析的方法；第2章线性表，介绍线性表的两种存储结构（顺序表和链表）的逻辑结构与基本运算的实现过程；第3章栈与队列，介绍两种特殊的线性结构的概念与应用；第4章串，介绍串的概念与模式匹配算法；第5章数组与广义表，介绍数组和稀疏矩阵的概念及相关运算的实现，以及广义表的存储结构及相关运算的实现；第6章树与二叉树，介绍树与二叉树的概念和各种运算的实现过程，其中特别突出二叉树的各种递归和非递归算法；第7章图，介绍图的基本概念和各种运算的实现过程；第8章查找，介绍各种常用查找算法的实现过程；第9章排序，介绍各种常用排序算法的实现过程；第10章文件，介绍常用的文件结构；第11章课程设计举例，给出了几个重要数据结构的具体实例。

数据结构是一门应用性非常强的课程，必须在掌握了各种数据结构的基础上，尽可能多地上机练习。为此，本书每章后面都配有相应的上机实验题。

全书采用C语言作为数据结构和算法的描述语言，所有算法均在VC++环境下调试通过。

本书具有以下特色：

1）语言通俗易懂，阐述简洁明了。

2）重点突出算法设计思路，注重培养学生的编程思想和解决实际问题的能力。

3）为激发学生学习该课程的兴趣，增强学生的创新意识，书中融入了一些利用所学知识解决实际问题的例子，如真值表的求解算法、出栈序列的求解算法等。

4）算法丰富，讲解透彻，便于学生自学。

5）通过课程设计的综合训练，使学生在学习理论知识的同时，进一步提高解决实际问题的能力，进一步强化综合应用训练，熟练掌握利用计算机解决问题的一般步骤。

6）通过典型算法设计的分析，使学生所学的知识更加系统化和条理化，更易于对所学知识融会贯通和举一反三。

7）为方便教师教学，本书配有电子教案和习题答案，可登录华章网站（www. hzbook. com）下载或发送邮件至 xfs@dzu. edu. cn 与作者联系。

在编写中我们参阅了许多数据结构教材和相关资料，在此向其作者一并表示感谢。本教材的出版得到了德州学院教材出版基金的资助。最后，还要特别感谢机械工业出版社华章公司的大力支持，使得本书得以顺利出版。

限于作者水平，书中不当和疏漏之处在所难免，敬请读者不吝指正。

作　者

教学建议

教学内容	学习要点及教学要求	课时安排
第1章 绪　论	熟悉各名词、术语的含义，掌握基本概念 理解算法的五个要素的确切含义 掌握计算语句频度和估算算法时间复杂度的方法	4
第2章 线性表	了解线性表的逻辑结构特性及其两类不同的存储结构 熟练掌握线性表的顺序存储和链式存储结构的描述方法，以及线性表的各种基本操作的实现算法 熟悉静态链表的存储结构和基本操作的实现算法，能够从时间和空间复杂度的角度综合比较线性表的两种存储结构的不同特点及其适用场合	12
第3章 栈与队列	掌握栈和队列类型的特点，并能在相应的应用问题中正确选用它们 熟练掌握栈类型的两种实现方法，应特别注意栈满和栈空的条件以及它们的描述方法 熟练掌握循环队列和链队列的基本操作实现算法，特别注意队满和队空的描述方法 理解递归算法执行过程中栈的状态变化过程	8
第4章 串	熟悉串的六种基本操作的定义，并能利用这些基本操作来实现串的其他操作 熟练掌握在串的定长数组存储和堆分配存储结构上实现串的各种操作的方法 了解串的链式存储结构 理解串匹配的KMP算法，熟悉 *next* 函数的定义，学会手工计算给定模式串的 *next* 函数值和改进的 *next* 函数值	6 （选讲）
第5章 数组与广义表	掌握数组的定义、顺序表示和实现 掌握矩阵的压缩存储 了解广义表的定义及其存储结构	8 （广义表部分选讲）
第6章 树与二叉树	熟练掌握二叉树的结构特性，了解相应的证明方法 熟悉二叉树的各种存储结构的特点及适用范围 掌握二叉树的各种遍历的递归算法和主要的非递归算法，灵活运用遍历算法实现二叉树的其他操作 理解二叉树线索化的实质，熟练掌握二叉树的线索化过程以及在中序线索化二叉树上找到给定结点的前驱和后继的方法 熟悉树的各种存储结构及其特点，掌握树和森林与二叉树的转换方法 学会编写实现树的各种操作的算法 了解哈夫曼树的特性，掌握建立哈夫曼树和哈夫曼编码的方法	12

（续）

教学内容	学习要点及教学要求	课时安排
第7章 图	熟悉图的各种存储结构及其构造算法 熟练掌握图的两种遍历算法，理解图的遍历算法与树的遍历算法之间的异同 能够应用图的遍历算法求解各种简单路径问题 掌握最小生成树的两种生成算法 掌握拓扑排序、最短路径、关键路径算法的思想	12
第8章 查　找	掌握顺序表和有序表的查找方法及其平均查找长度的计算方法 熟练掌握二叉排序树、二叉平衡树的构造和查找方法 理解B－树、B＋树和键树的特点以及它们的查找过程 熟练掌握哈希表的构造方法和处理冲突的方法 掌握按定义计算各种查找方法在等概率情况下查找成功时的平均查找长度	12
第9章 排　序	了解排序的定义和各种排序方法的特点 熟悉各种方法的排序过程及其依据的原则，掌握各种排序方法的时间复杂度的分析方法 理解排序方法“稳定”或“不稳定”的含义，弄清楚在什么情况下要求应用的排序方法必须是稳定的 了解外部排序的基本过程及其时间分析	12
第10章 文　件	了解文件组织的多种形式及其特点 掌握文件的插入、删除、检索等主要操作在不同组织形式上的实现	4 （选讲）
第11章 课程设计举例	熟悉各种数据结构的应用	12 （课下）

说明：

1）本教材主要是为计算机专业的本科“数据结构”课程而编写的。建议授课学时数为72～90（包含实验课、习题课、讨论课等必要的教学环节），不同学校可根据各自的教学要求、教学计划学时及学生实际情况对教材内容进行取舍。

2）非计算机专业的师生在使用本教材时可适当降低要求或减少教学内容。

3）本书每章后面配有典型例题、上机实验和习题，各学校可根据实际需要进行遴选。

4）建议授课与上机学时数比例为2:1。

5）第11章由学生课下完成。

目　录

第1章 绪 论

随着计算机产业的飞速发展，计算机的应用领域越来越广泛，已不再局限于科学计算，而是更多地用于控制、管理及数据处理等非数值计算的处理工作。所有的系统软件和应用软件都要用到各种类型的数据结构，在计算机中如何组织数据、处理数据就成了人们进行程序设计的关键所在。因此，我们必须研究数据的特性和数据间的相互关系及其对应的存储表示，并设计出相应的算法，更好地实现程序。数据结构是计算机专业的核心课程，该课程的学习可以帮助人们更好地进行程序设计，解决实际问题。

1.1 数据结构的研究对象

自然界中的许多问题是可以用数学工具进行描述的。例如，造桥中的受力问题可描述为线性方程，人口增长的情况可描述为微分方程。但更多的非数值计算问题无法用数学方程加以描述。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，才能有效地解决问题。请看以下列举的具体问题。

例1 学生信息检索系统。当我们需要查找某个学生的有关信息或某个专业的情况时，只要建立了相应的数据结构，按照某种算法编写了程序，就可以实现计算机自动检索。由此，可以在学生信息检索系统中建立一个按学号顺序排列的学生信息表和按专业排列的索引表，如表1-1所示。由这个表构成的文件便是学生信息检索的数学模型，计算机的操作就是按照某种要求对学生信息文件进行查询。

表1-1 学生信息查询系统中的数据结构

a）学生信息表

学 号	姓 名	性 别	专 业	年 级
200701002001	韦志君	男	信息管理与信息系统	2007级
200701001088	唐 立	男	计算机科学与技术	2007级
200701002002	宋奇锋	男	信息管理与信息系统	2007级
200701001089	熊 伟	男	计算机科学与技术	2007级
200701001091	许文锋	男	计算机科学与技术	2007级
200701002003	张小龙	男	信息管理与信息系统	2007级
200701002004	陈亚雯	女	信息管理与信息系统	2007级
200701001084	彭金萍	女	计算机科学与技术	2007级

b）专业索引表

信息管理与信息系统	1，3，6，7
计算机科学与技术	2，4，5，8

诸如此类的还有电话自动查号系统、考试查分系统、仓库库存管理系统等。在这类文档管理的数学模型中，计算机处理的对象之间存在一种简单的线性关系，因此这类数学模型可称为线性数据结构。

例 2 八枚硬币问题。假设有八枚硬币，分别表示为 a、b、c、d、e、f、g、h，其中有且仅有一枚硬币是假币，并且假币的重量与真币的重量不同，可能比真币轻，也可能比真币重。现要求以天平为工具，用最少的比较次数挑选出假币，并同时确定这枚假币的重量比其他真币轻还是重。解决这个问题的最自然的想法就是把硬币分成两组，依次进行判断。其判断的全过程如图 1-1 所示。这里用到的是树形数据结构。

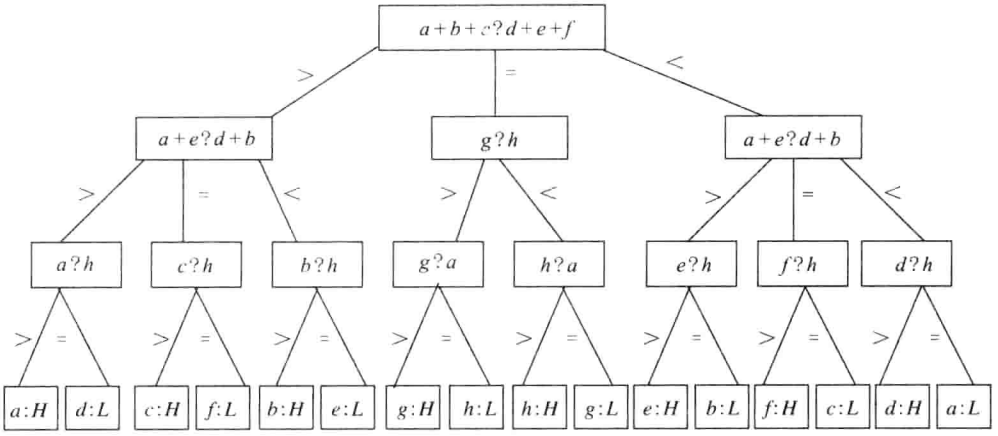


图 1-1 八枚硬币问题的数据结构

例 3 教学计划编排问题。一个教学计划包含许多课程，在这些课程之间，有些课程要按规定的先后次序进行，有些则没有次序要求。也就是说，有些课程之间有先修和后继的关系，有些课程可以任意安排次序。假设计算机专业的课程设置如表 1-2 所示，则这些课程之间的次序关系可以用一个称为有向图的数据结构来表示，如图 1-2 所示。有向图的每个结点表示一门课程，若从结点 C_i 到 C_j 之间存在有向边 $<C_i, C_j>$，则表示课程 C_i 必须先于课程 C_j 开设；若两门课程之间没有有向边，则表示这两门课程之间没有开设的先后次序。

表 1-2 计算机专业的课程设置

课程编号	课程名称	先修课程
C_1	计算机导论	无
C_2	高等数学	无
C_3	C 语言程序设计	C_1
C_4	离散数学	C_2, C_3
C_5	数据结构	C_1, C_3, C_4
C_6	汇编语言	C_1, C_3
C_7	接口技术	C_6
C_8	操作系统	C_5
C_9	数据库原理	C_5, C_8
C_{10}	编译原理	C_3

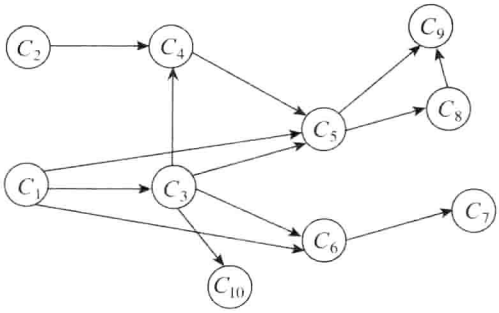


图 1-2　教学计划编排问题的数据结构

由以上例子可知，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、图之类的数据结构。使用计算机解决这些实际问题，大致需要经过下列三个步骤：

1）从具体问题中抽象出一个适当的数学模型。

2）设计一个解此数学模型的算法。

3）编出程序、进行测试、调整直至得到最终答案。

数据结构这门课程就是要解决这三个步骤中的第一步和第二步中所提到的问题。建立数学模型的实质是分析问题，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。设计解数学模型的算法就是给出处理问题的策略。

概括地说，**数据结构**是一门讨论“描述现实世界实体的数学模型（非数值计算）及其上的操作在计算机中如何表示和实现”的学科。实质上，好的程序设计就是好的算法加上好的数据结构。

数据结构在计算机科学中是一门综合性的专业基础课。数据结构的研究不仅涉及计算机硬件（特别是编码理论、存储装置和存取方法等）的研究范围，而且和计算机软件的研究有着密切的关系，无论是编译程序还是操作系统，都涉及数据元素在存储器中的分配问题。在研究信息检索时也必须考虑如何组织数据，以便更为方便地查找和存取数据元素。因此，数据结构是介于数学、计算机硬件和计算机软件三者之间的一门核心课程。在计算机科学中，数据结构不仅是一般程序设计（特别是非数值计算的程序设计）的基础，而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的重要基础。

数据结构课程集中讨论软件开发过程中的设计阶段，同时涉及编码和分析阶段的若干基本问题。此外，为了构造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与选择。因此，数据结构的内容包括 3 个层次的 5 个要素，如表 1-3 所示。

表 1-3　数据结构课程的内容体系

要素 / 层次	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	

1.2 数据结构的发展概况

数据结构作为一门独立的课程是从 1968 年开始的。在此之前，其内容曾出现在不同的其他课程中，如表处理语言等。1968 年在美国一些大学计算机系的教学计划中，虽然把数据结构规定为一门课程，但对该课程的范围没有作出明确规定。当时，数据结构几乎和图论，特别是和表、树的理论互为同义语。随后，数据结构这个概念被扩充到包括网络、集合代数论、格、关系等方面，从而变成了现在称之为离散数学的内容。然而，由于数据必须在计算机中进行处理，因此，不仅需要考虑数据本身的数学性质，而且必须考虑数据的存储结构，这就进一步扩大了数据结构的内容。近年来，随着数据库系统的不断发展，数据结构课程中又增加了文件管理的内容。

1968 年美国唐·欧·克努特教授所著的《计算机程序设计艺术：第 1 卷　基本算法》是第一本较系统地阐述数据的逻辑结构和存储结构及其操作的著作。从 20 世纪 60 年代末到 20 世纪 70 年代初，出现了大型程序，软件相对独立，结构程序设计成为程序设计的主要内容，人们越来越重视数据结构，认为程序设计的实质是对确定的问题选择一种好的数据结构，并设计一个好的算法。从 20 世纪 70 年代中期到 20 世纪 80 年代初，各种版本的数据结构著作相继出现。

目前在我国，数据结构已成为计算机专业的核心课程之一，而且是其他非计算机专业的主修课程之一。

1.3 基本概念与术语

计算机科学是研究信息表示和处理的科学，信息在计算机内是用数据表示的。直观地说，**数据**是用于描述客观事物的数值、字符以及一切可以输入到计算机中并由计算机程序加以处理的符号的集合，是计算机操作的对象的总称。

数据元素是数据的基本单位，它是数据中的一个“个体”，如整数“5”、字符“N”等。有时，一个数据元素可由若干**数据项**组成，例如，描述一个学生的信息为一个数据元素，学生信息中的每一项（如姓名、学号等）为一个数据项。数据项是数据的不可分割的最小单位。

数据对象是具有相同性质的数据元素的集合，是数据的一个子集。数据元素是数据对象的实例。例如，整数数据对象是集合 $\{0, \pm1, \pm2, \pm3, \cdots\}$。

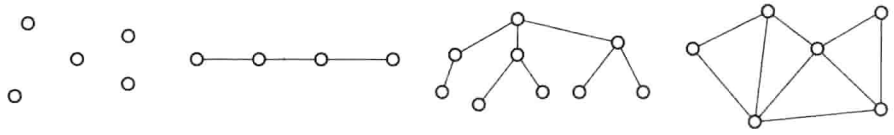
数据结构是指相互之间存在一种或多种关系的特性相同的数据元素的集合。根据数据元素之间关系的不同，数据通常有以下四类基本的结构。

1）集合结构：在集合结构中，数据元素之间的关系是“属于同一个集合”。集合是元素关系极为松散的一种结构。

2）线性结构：该结构中的数据元素之间存在着一对一的关系。

3）树形结构：该结构中的数据元素之间存在着一对多的关系。

4）图形结构：该结构中的数据元素之间存在着多对多的关系。由于集合结构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。图 1-3 为上述四类基本结构的关系图。



a）集合结构　b）线性结构　c）树形结构　d）图形结构

图 1-3　四类基本结构关系图

由数据结构的概念可知，数据结构有三个要素：一是数据元素的集合，二是数据元素之间关系的集合，三是定义在其上的操作。在形式上，数据结构可定义为一个二元组：

$$\text{Data\_Structures} = (D, S)$$

其中：D 是数据元素的有限集，S 是 D 上关系的有限集。

数据结构包括数据的逻辑结构和物理结构。数据的**逻辑结构**是对数据元素之间的逻辑关系的描述，它可以用一个数据元素的集合和定义在此集合上的若干关系来表示，与数据的存储无关；数据的**物理结构**是逻辑结构在计算机中的表示和实现，故又称“存储结构”。

数据结构在计算机中的表示，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的一位，叫做位（bit）。在计算机中，可以用由若干位组合起来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 8 位二进制数表示一个字符等）。

数据元素之间的关系在计算机中有四种不同的表示方法，下面分别介绍。

（1）顺序存储方法

该方法把逻辑上相邻的元素存储在物理位置上相邻的存储单元里，结点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，该结构通常是借助于计算机程序设计语言（例如 C、C++）的数组来描述的。

顺序存储方法的主要优点是节省存储空间，因为分配给数据的存储单元完全用于存放数据（不考虑 C、C++语言中数组需指定最大存储空间大小的情况），结点之间的逻辑关系不占用额外的存储空间。采用这种方法，可实现对数据的随机存储。但顺序存储方法的主要缺点是不便于修改，对数据进行插入、删除操作时，可能要移动一系列数据。

（2）链式存储方法

该方法不要求逻辑上相邻的元素在物理位置上也相邻，元素之间的逻辑关系是由附加的指针字段表示的。由此得到的存储表示称为链式存储结构，该结构通常是借助于计算机程序设计语言（例如 C、C++）的指针类型来描述的。

链式存储方法的主要优点是便于修改，在进行插入、删除操作时，仅需要修改指向相应数据元素的指针，而不必移动数据元素。但与顺序存储方法相比，其主要缺点是存储空间的利用率较低，因为分配给数据的存储单元有一部分用来存储数据元素之间的逻辑关系了。另外，由于逻辑上相邻的数据元素在存储空间上不一定相邻，所以不能对其进行随机存取。

（3）索引存储方法

该方法通常在存储数据元素的同时，还建立附加的索引表。索引表中的每一项称为索引项，索引项的一般形式是：（关键字，地址），关键字唯一标识一个数据元素，地址作为指向该数据元素的指针。这种带有索引表的存储结构可大大提高数据查找的速度。

线性结构中采用索引存储方法后，可对数据元素进行随机存取。在进行插入、删除操作时，只需移动存储在索引表中对应数据元素的存储地址，而不必移动数据元素本身，所以仍能保持较高的数据修改运算效率。索引存储方法的缺点是增加了索引表，降低了存储空间的利用率。

（4）哈希（或散列）存储方法

该方法的基本思想是根据数据元素的关键字，通过哈希函数直接计算出一个值，并将这个值作为该数据元素的存储地址。

哈希存储方法的优点是查找速度快，只要给出待查找数据元素的关键字，就可以立即计算出该数据元素的存储地址。但与前三种存储方法不同的是，哈希存储方法只存储数据元素，不存储数据元素之间的逻辑关系。哈希存储方法一般只适合要求对数据进行快速查找和插入的场合。

上述四种存储方法既可以单独使用，也可以组合起来使用。

1.4 数据类型与抽象数据类型

1.4.1 数据类型

数据类型是和数据结构密切相关的一个概念，它最早出现在高级程序语言中，用以刻画（程序）操作对象的特性。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。类型显式或隐含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。因此，**数据类型**是一个值的集合和定义在此集合上的一组操作的总称。例如，C 语言中的整型变量，其值为某个区间上的整数（依赖于机器），定义在其上的操作为加、减、乘、除和取模等算术运算。

按“值”的不同特性，高级程序语言中的数据类型分为**原子类型**和**结构类型**两类。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。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，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，而数据类型则可看成是由一种数据结构和定义在其上的一组操作组成的。

1.4.2 抽象数据类型

抽象数据类型（Abstract Data Type，ADT）是指一个数学模型以及定义在此数学模型上的一组操作。例如，“整数”是一个抽象数据类型，其数学特性和具体的计算机或语言无关。“抽象”的意义在于强调数据类型的数学特性。抽象数据类型和数据类型实质上是一个概念，只是抽象数据类型的范围更广，除了已有的数据类型外，抽象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。

ADT 的定义取决于它的一组逻辑特性，而与其在计算机内的表示和实现无关。因此，不论 ADT 的内部结构如何变化，只要其数学特性不变，都不影响其外部的使用，从而为实现软件的部件化和可重用性提供了保证，进而提高了软件生产率。