学校教材·计算机科学与技术

朱振元
朱　承　编著

数据结构

（面向对象语言描述）

清华大学出版社

高等学校教材·计算机科学与技术

数据结构
（面向对象语言描述）

朱振元　朱　承　编著

清华大学出版社
北京

内容简介

本书采用面向对象的观点讨论数据结构技术，以类定义为线索对各数据类型中所定义的操作进行说明。内容包括：线性表、栈、队列、串、二维数组、广义表、树、图、查找和排序等。

书中使用面向对象的开发工具对各章所定义的类的功能进行演示，可使读者加深对课程内容的理解，并促进软件开发能力的提高。

本书可作为大专院校计算机专业必修课的教科书，也可作为计算机科技人员及电脑爱好者的自学参考书。

图书在版编目(CIP)数据

数据结构(面向对象语言描述)/朱振元，朱承编著. —北京：清华大学出版社,2004.2
(高等学校教材·计算机科学与技术)
ISBN 7-302-07960-9

Ⅰ.数… Ⅱ.①朱… ②朱… Ⅲ.①数据结构—高等学校—教材 ②面向对象语言—程序设计—高等学校—教材 Ⅳ.①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字(2004)第 002559 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社总机：010-62770175　　**客户服务**：010-62776969
责任编辑：闫红梅
封面设计：张　音
印 装 者：清华大学印刷厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：18.5　**字数**：456 千字
版　　次：2004 年 2 月第 1 版　2004 年 2 月第 1 次印刷
书　　号：ISBN 7-302-07960-9/TP·5782
印　　数：1～5000
定　　价：26.00 元

前　言

“数据结构”课程是我国计算机教学中较早形成和完善的一门专业基础课程，也是计算机课程体系中的核心课程之一。在该课程中所介绍的各类数据的逻辑结构、存储方式及相关的算法既是程序设计，特别是非数值性程序设计的基础，又是设计和实现系统软件及大型应用软件的重要基础。

多年来，有关数据结构课程的教材已出版了许多，这些教材一般都各有特色并在各高校的计算机教学中发挥了重要的作用。随着计算机技术的飞速发展，程序设计方法及软件开发技术也出现了重大的发展及变革，这为我们对各专业课程的教材更新提出了新的要求。为了适应程序设计方法的这种变革，笔者编写了这本教材。

本教材以培养与提高学生的基本专业素质及综合应用能力为目标，侧重于教材的先进性、适应性、实用性及可读性，在教材中体现了以下特色：

1. 在本教材中引入了面向对象的概念及实现方法。以面向对象的实现方法逐步取代传统的面向过程的实现方法，这是程序设计方法发展过程的必然趋势，也是数据结构教材版本更新过程的必然趋势。在数据结构课程中所介绍的每一种抽象的数据类型都可以直接定义成一种对象，因而，使用这种实现方法不仅适应了程序设计方法及开发技术的发展与变革，而且对数据结构课程也非常适宜。这体现了教材内容的先进性。

2. 本教材非常重视实验环节。在每一章最后都结合该章内容设置一个相应的教学演示程序。这个教学演示程序可作为学生的综合实习程序。通过编制与上机调试程序，使学生加深对课程内容的理解，并逐步积累编制与调试程序的经验，从而促进学生应用能力的培养与提高。

3. 在本教材中对每一种数据类型都有比较规范的表述过程，对每一种算法都有比较规范统一的说明步骤，对算法的含义、参数与功能、工作变量说明、处理过程都进行明确的说明，并通过图示、文字注释、实例的执行过程等多种方式来帮助学生理解算法，提高学生的计算机思维能力。

本教材在引了面向对象的概念的同时，还结合相应的开发工具C++ Builder来实现书中所介绍的各种算法及演示程序，目的是使各种数据类型的实现方法落到实处。书中所出现的程序代码都是经过严格调试，确定为正确无误后才复制到书中来，从而使代码的出错率降低为零，并提高了程序的有效性。

本书在《数据结构——面向对象实现方法》（西安电子科技大学出版社出版）的基础上编写而成，在保持原书中的演示程序及叙述风格不变的前提下，在描述语言、目录结构及程序的适应性等方面，做了许多更合理的改进。

由于本教材在实现方法及编写策略等方面都做了一些新的尝试，加之作者水平有限、时间仓促，书中难免存在缺点与疏漏，敬请读者及同行们予以批评指正。

朱振元

2003年9月

目　录

第1章 课程概论

当今人类社会已进入了一个以信息化为标志的飞速发展时代。数据是信息的载体，随着信息化社会的发展，由计算机处理的数据数量随之增大，数据类型随之增多，数据结构随之更加复杂。由于数据的组织方式直接关系到程序结构的优劣、程序处理的效率，这就给程序设计带来了一些新的问题。为了编出一个结构好、效率高的处理程序就必须分析待处理的对象特性以及各处理对象之间存在的关系。这就是“数据结构”这门学科形成和发展的背景。

1.1 课程的初步认识

在计算机的使用初期，它的主要应用领域是科学计算。当使用计算机解决一个具体问题时，一般需要经过下列几个步骤：首先要从该具体问题抽象出一个适当的数学模型，然后设计或选择一个解此数学模型的算法，最后编出程序进行测试、调试，直至得到最终的解答。例如求解梁架结构中应力的数学模型为线性方程组，可以使用迭代算法来求解线性方程组。

然而，随着计算机应用领域的不断扩大，出现了许多无法用数值及数学方程加以描述的具体问题。这是一类非数值计算问题，下面所列举的就是属于这一类的具体问题。

例 1-1 人事信息检索问题。当要查找某公司员工的信息时，一般是给出该员工的编号或姓名，通过信息目录卡片和信息卡片来查得该员工的有关信息。但也有可能要查找某一种类别的员工信息，例如要查找该公司的员工中具有“高级程序员”技术职称的人员信息，或者是属于某一行政分组的人员信息等。若利用计算机来检索人事信息，则计算机所处理的对象便是这些信息目录卡片及员工信息卡片中的信息。为此，在计算机中必须建立和存储与此相关的3张表，一张是按员工编号排列的员工信息表，其余两张分别是按技术职称与按行政分组顺序排列的索引表。在员工信息表中存放编号、姓名、职称、职务及爱好等信息，在职称索引表中存放职称与员工编号的对应信息，在组别索引表中存放行政分组与员工编号的对应信息，如图1.1所示。

在人事信息检索问题中，上述这几张表便是数学模型，计算机的主要操作便是按指定的要求对这些表进行查找。在这一类属于文档管理的数学模型中，计算机的处理对象之间通常存在着一种最简单的线性关系，相应的数据结构可称为线性数据结构。

例 1-2 八皇后问题。在八皇后问题中，处理过程不是根据某种确定的计算法则，而是利用试探和回溯的探索技术求解。为了求得合法布局，在计算机中要存储布局的当前状态。从最初的布局状态开始，一步步地进行试探，每试探一步形成一个新的状态，整个试探过程形成了一棵隐含的状态树，如图1.2所示(为了描述方便，将八皇后问题简化为四皇后问题)。回溯法求解过程实质上就是一个遍历这棵状态树的过程。在这个问题中所出

现的“树”也是一种数据结构，它可以应用在许多非数值计算的问题中。

编　号	姓　名	职　称	职　务	爱　好
990001	丁一	系统分析员	总工	羽毛球、乒乓球
990002	丁二	高级程序员	一组组员	
990003	马一	程序员	一组组员	
990004	马二	程序员	一组组员	
990005	张一	高级程序员	一组组长	网球、乒乓球
990006	张二	高级程序员	二组组员	网球
990007	李一	程序员	二组组员	
990008	李二	程序员	二组组员	
990009	王一	程序员	二组组员	
990010	王二	程序员	二组组员	

(a) 员工信息表

系统分析员	1
高级程序员	2,5,6
程序员	3,4,7,8,9,10

(b) 职称索引表

第一组	5,2,3,4
第二组	6,7,8,9,10

(c) 组别索引表

图 1.1　人事信息检索系统中的数据结构

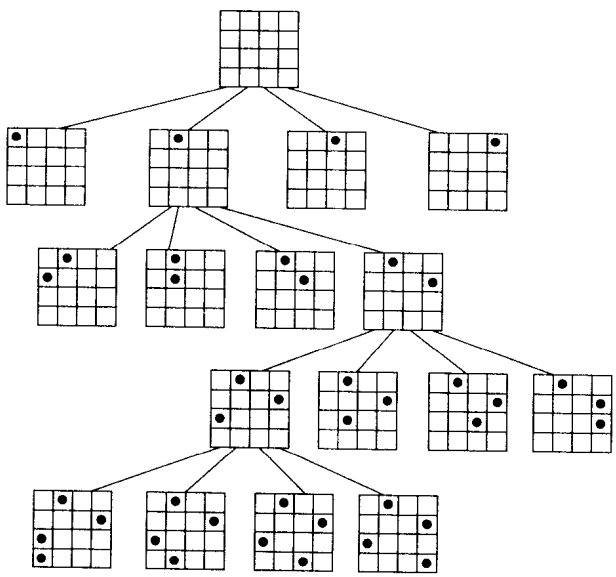

图 1.2　四皇后问题中隐含的状态树

在交通咨询系统中，也可以采用一种图的结构来表示实际的交通网络。图中的顶点表示城市，边表示城市间的交通联系，对边所赋予的权值可以表示两城市间的距离，或途中所需的时间或交通费用等。考虑到交通图的有向性(如航运、逆水和顺水时的船速就不一

样)，图中的边可以用弧来表示，如图 1.3 所示。这个咨询系统可以回答旅客提出的各种问题，例如从某地到某地应如何走才最节省费用，也就是要求从某地到某地的最短路径。在这个问题中所使用的图也是一种数据结构，它被用于解决这一类实际问题。

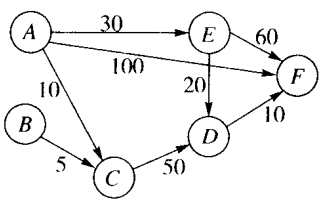

图 1.3　一个表示交通网的例图

综上 3 个例子可见，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。因此，可以说数据结构课程主要是研究非数值计算的程序设计问题中所出现的计算机操作对象以及它们之间的关系和操作的学科。

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、数据库系统以及其他系统程序的大型应用程序的重要基础。

在计算机专业中，数据结构是一门综合性的专业基础课程，它不仅是计算机专业教学计划中的核心课程之一，而且是非计算机专业的主要选修课程。

学习数据结构课程的目的是为了了解计算机处理对象的特性，将现实世界中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，通过算法训练提高计算机思维的能力，通过程序设计的技能训练来促进综合应用能力和专业素质的提高。

本书按面向对象程序设计的基本方法，以数据类型及其相应的实现为主线索，以实例引入、数据存储、基本操作、程序实现、归纳概括为基本环节，由浅入深地展开讲解。目的就是使读者掌握课程中最基本最重要的内容——掌握面向对象的实现方法，培养运用数据结构的知识解决实际问题的能力。

1.2　数据结构的基本概念

在系统学习数据结构知识之前，我们先了解一些概念和术语的确切含义。

1.2.1　基本术语

数据(Data)是信息的载体，它能够被计算机识别、存储和加工处理。它是计算机程序加工的原料。例如，一个利用数值分析方法解代数方程的程序所用的数据是整数和实数，而一个编译程序或文本编辑程序所使用的数据是字符串。随着计算机软、硬件技术的发展，应用领域的扩大，数据的含义也随之拓宽。像多媒体技术中所涉及的视频和音频信号，经采集转换后都能形成被计算机接受的数据。

数据元素(Data Element)是数据的基本单位。在不同条件下，数据元素又可称为元素、结点、顶点、记录。例如，在人事信息检索问题中员工信息表的一个记录，在八皇后问题中状态树的一个状态，在交通咨询系统中交通网的一个顶点等。在数据元素是记录的情形下，一个数据元素又可由若干数据项(也称为字段、域)组成，数据项是具有独立含义的最小单位。

数据元素类(Data Element Class)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)，属于同一数据元素类，数据元素是数据元素类的一个实例。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，是该数据元素类中的两个实例，其数据元素的值分别为 A 和 B。

1.2.2 数据结构的概念

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，数据元素之间都不会是孤立的，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之为结构。根据数据元素间关系的不同特性，通常有下列 4 类基本结构：

- 集合　在集合结构中，数据元素间的关系是“属于同一个集合”，集合是元素关系极为松散的一种结构。
- 线性结构　结构中的数据元素之间存在一个对一个的关系。
- 树形结构　结构中的数据元素之间存在一个对多个的关系。
- 图状结构　结构中的数据元素之间存在多个对多个的关系，图状结构也称网状结构。图 1.4 表示上述 4 类基本结构的关系图。

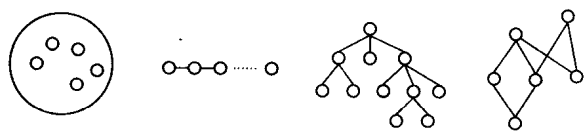

图 1.4　4 类基本结构的关系图

1.2.3 逻辑结构和物理结构

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中抽象出来的数学模型，它与数据的存储无关，而我们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两种。

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，元素间的逻辑关系由存储单元的相邻关系来体现。由此得到的存储表示称为顺序存储结构。顺序存储结构通常是借助于程序语言中的数组来实现的。

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附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。链式存储结构通常是借助于程序语言中的指针类型来实现的。

除了通常采用的顺序存储结构和链式存储结构外，有时为了查找方便还采用索引存储方法和散列存储方法。

1.2.4　数据结构形式定义

从 1.2.2 节所介绍的数据结构的概念中我们可以知道，一个数据结构有两个元素，一个是元素的集合，另一个是关系的集合。因此在形式上，数据结构通常可以采用一个二元组来表示，即：

```
Data Structure = (D,R)
```

其中，*D* 是数据元素的有限集，*R* 是 *D* 关系的有限集。

例如，在 1.1 节中所介绍的人事信息检索问题中，数据元素集合主要是员工信息表中的所有记录，而元素间关系的集合则可以从不同的视点去建立：可以按员工的编号来建立元素间的线性关系从而形成线性的数据结构；可以按员工的行政分组来建立元素间的层次关系从而形成树形的数据结构；可以按员工的爱好建立元素间的网状关系从而形成网状的数据结构，如图 1.5 所示。

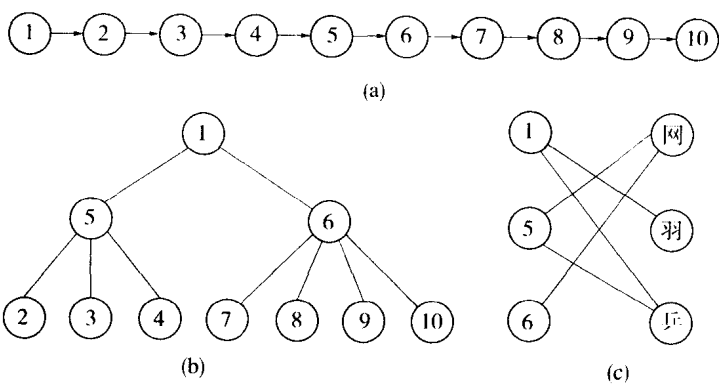

图 1.5　员工信息表中的数据结构

如果我们使用二元组来表示上述的线性结构，则可表示为：

```
Linear-List = (D,R)
```

其中：

```
D = {01,02,03,04,05,06,07,08,09,10} (用编号的后两位表示)
R = { <01,02>, <02,03>,... <09,10> }
```

树形结构则可表示为：

```
Tree = (D,R)
```

其中：

```
D = {01,02,03,04,05,06,07,08,09,10}
R = { <01,05>, <01,06>, <05,02>,... <06,10> }
```

图状结构则可表示为:

```
Graph =(D,R)
```

其中:

```
D = {01,02,03,04,05,06,07,08,09,10, 网,羽,乒}
R = {<01,羽>,<01,乒>,<05,网>,<05,乒>,<06,网>}
```

1.3 数据类型及面向对象概念

1.3.1 数据类型概述

数据类型(Data Type)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,用以刻画程序中操作对象的特性。在用高级语言编写的程序中,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这个值集上的一组操作的总称。例如 PASCAL 语言中的整数类型,其取值的范围为[-maxint, maxint]上的整数(maxint 是依赖特定计算机的最大整数),定义在其上的一组操作为加、减、乘、除及取模等。

在高级程序语言中,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。原子类型的值不可分解,如 PASCAL 语言中的整型、实型等标准类型,而结构类型的值由若干成分按某种结构组成,因此是可分解的,并且它的成分可以是非结构的也可是结构的。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。

1.3.2 抽象数据类型

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。

但在另一方面,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,还包括用户在设计软件系统时自己定义的数据类型。为了提高软件的重用率,在近代程序设计方法学中要求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,并在模块的内部给出这些数据的表示及其操作的细节,