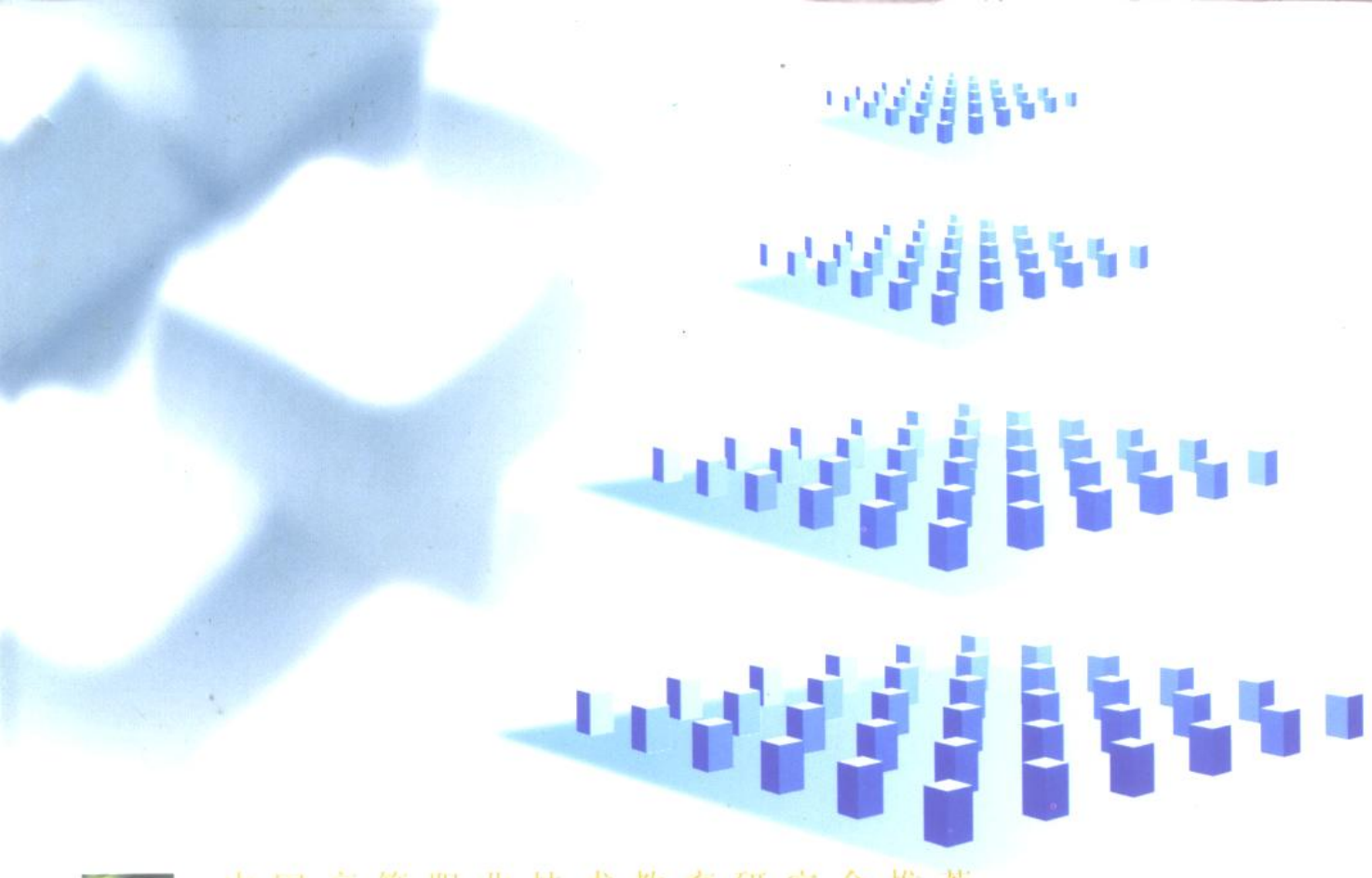

中国高等职业技术教育研究会推荐

高职系列教材

数据结构

朱振元 朱 承 编著

面向21世纪高级应用型人才

西安电子科技大学出版社
http:// www.xduph. com

□ 中国高等职业技术教育研究会推荐

高职系列教材

数据结构

——面向对象实现方法

朱振元　朱　承　编著

西安电子科技大学出版社
2000

内 容 简 介

本书介绍了计算机学科中常用的数据结构，内容包括线性表、栈队、列、串、数组、广义表、树、图、查找及排序等。

本书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，侧重内容的先进性、实用性及科学性。书中全面引入并应用面向对象的方法具体实现某些结构，每章均引入一个实例，将其抽象、改造和拓宽并归结为一个程序设计问题或一个演示程序，便于学生理解和掌握教学内容。

本书主要为适应高等职业教育而著，同时也可作为大专院校计算机专业必修课的教材及计算机科技人员与爱好者的自学参考书。

图书在版编目(CIP)数据

数据结构/朱振元，朱承编著. —西安：西安电子科技大学出版社，2000.7

高职系列教材

ISBN 7-5606-0878-7

Ⅰ.数… Ⅱ.①朱… ②朱… Ⅲ.数据结构-高等学校：技术学校-教材 Ⅳ.TP311.12

中国版本图书馆 CIP 数据核字(2000)第 32027 号

责任编辑 云立实 戚文艳

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)8227828 邮 编 710071

http://www.xduph.com E-mail：xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 西安长青印刷厂

版 次 2000 年 7 月第 1 版 2000 年 7 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 20.5

字 数 484 千字

印 数 1~4 000 册

定 价 21.00 元

ISBN 7-5606-0878-7/TP·0462

序

在即将跨入21世纪的前夕，中共中央、国务院召开了第三次全国教育工作会议，并颁发了《中共中央、国务院关于深化教育改革全面推进素质教育的决定》，进一步明确了高等职业教育的重要地位，指出“高等职业教育是高等教育的重要组成部分。要大力发展高等职业教育。”在这一方针的指引下，我国高等职业教育取得了空前规模的发展。至1999年，从事高等职业教育的高等职业学校、高等专科学校和独立设置的成人高校已达1345所，占全国高校总数的69.2%；专科层次的在校生占全国高校在校生的55.37%，毕业生占高校毕业生总数的68.5%。这些数字表明，高等职业教育在我国高等教育事业中占有极其重要的地位，在我国社会主义现代化建设事业中发挥着极其重要的作用。随着社会的发展、科技的进步，以及我国高等教育逐步走向大众化，我国的高等职业教育必将进一步发展壮大。

在高等职业教育大发展的同时，也有着许多亟待解决的问题。其中最主要的是按照高等职业教育培养目标的要求，培养一批“双师型”的中青年骨干教师；编写出一批有特色的基础课和专业主干课教材；创建一批教学工作优秀学校。

为解决当前高职教材严重匮乏的问题，西安电子科技大学出版社与中国高等职业技术教育研究会联合策划、组织编写了计算机及应用电子技术两个专业的教材，现已出版。本系列教材，从策划到主编、主审的遴选，从成立专家组反复讨论大纲，研讨职业教材特色到书稿的字斟句酌，每走一步都比较扎实、十分精心。作者在编写中紧密联系实际，尽可能地吸收新理论、新技术、新工艺，并按照案例引入、改造拓宽、课题综合(通过一个大型的课题，综合运用所学内容)的思路，进行编写，努力突出高职教材的特点。本系列教材内容取材新颖、实用；层次清楚，结构合理；文笔流畅，装帧上乘。这套教材比较适合高等职业学校、高等专科学校和成人高校等高等职业教育的需要。

教材建设是高等职业院校基本建设的主要工作之一，是教学内容改革的重要基础。为此，有关高职院校都十分重视教材建设，组织教师积极参加教材编写，为高职教材从无到有，从有到优而辛勤工作。但高职教材的建设还刚刚起步，还需要做艰苦的工作，我们殷切地希望广大从事高等职业教育的教师，在教书育人的同时，组织起来，共同努力，编写出一批高职教材的精品，为推出一批有特色的、高质量的高职教材作出积极的贡献。

中国高等职业技术教育研究会会长 李家瑞

高等职业技术教育“计算机及应用电子技术专业”教材编审专家委员会

主　任：闵光太（中国高等职业技术教育研究会副会长，
金陵职业大学校长，教授）

副主任：俞克新（中国高等职业技术教育研究会秘书长，研究员）
孙建京（北京联合大学教务长，副教授）
余苏宁（深圳职业技术学院计算机应用工程系副主任，副教授）
李荣才（西安电子科技大学出版社总编辑，教授）

计算机组

组　长：余苏宁（兼）

成　员：（按姓氏笔画排列）
丁桂芝（天津职业大学计算机工程系主任，副教授）
朱振元（长沙大学高级工程师）
张　燕（金陵职业大学计算机系讲师）
唐连章（广州大学副教授）
韩伟忠（金陵职业大学计算机系主任，副教授）
樊月华（北京联合大学应用技术学院副教授）
颜　彬（江汉大学副教授）

应用电子技术组

组　长：孙建京（兼）

成　员：（按姓氏笔画排列）
付植桐（天津职业大学副教授）
刘守义（深圳职业技术学院电子通信工程系副主任，高工）
李建民（江汉大学应用物理系副主任，副教授）
高泽涵（广州大学机电工程系副主任，高级实验师）
鲁宇红（金陵职业大学副校长，副教授）
熊幸明（长沙大学工程系主任，副教授）

总策划：梁家新

策　划：马乐惠　徐德源　云立实

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，有关数据结构课程出版了许多很好的教材，这些教材一般都各有特色，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变革，同时也是为了适应我国高等职业教育事业的发展，使数据结构课程的教材能较好地体现出高等职业教育的特点，我们编写了这本教材。

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追求目标，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的特色：

1. 本教材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. 本教材面向高职教学的层面，通过“问题驱动、改造拓宽”的分析方法使教材的内容深入浅出。每一章都以实例引入，对实例进行抽象、改造与拓宽，并尽可能地将实例归结为一个具体的程序设计问题或一个相应的演示程序，然后就基本上环绕该程序的实现来展开讨论，以便于学生对所学内容的理解与掌握。

3. 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。

4. 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，提高学生的思维能力。

本教材在引入了面向对象的概念的同时，还结合了一种相应的开发工具，目的是使各种数据类型的实现方法落到实处。如果光是使用伪码进行抽象的算法描述，则其上机实习的效果可能不会太好。因此我们选择了一种便于实现的开发工具，具体来讲，我们选择 Object Pascal 与 Delphi 分别作为描述语言及实现工具。由于 Pascal 语言可读性好、生命力

强，作为一种较好的教学工具被广泛使用，由 Pascal 语言中的记录类型过渡到面向对象语言中的类定义比较自然，而 Delphi 则是面向对象开发工具中的优秀代表，有广泛的使用前景。

考虑到 Object Pascal 语言与 Delphi 开发工具并不一定是数据结构的先行课程，因此在本书的附录中对其中所涉及到的有关内容进行了概要的介绍。但对于学习本书的读者来说，至少应具备 Pascal 语言程序设计的知识。

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

朱振元

2000 年 4 月

目　录

第 1 章

课 程 概 论

数据是信息的载体，随着信息化社会的发展，由计算机进行处理的数据，其数量类型随之增多，数据结构也随之复杂化。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对象的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

计算机使用初期，其主要应用领域是科学计算。当我们使用计算机来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的线性方程组，该方程组可以使用迭代算法来求解。

然而，随着计算机应用领域的不断扩大，出现了很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**］ 人事信息检索问题。当我们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，我们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。若利用计算机来实现人事信息检索，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与存储与此相关的三张表。一张是按员工编号排列的员工信息表，另两张分别是按技术职称与行政分组顺序排列的索引表。员工信息表中存放着编号、姓名、职称、职务及爱好等信息，在职称索引表中存放着职称与员工编号的对应信息，在组别索引表中存放着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，图 1.1 的这几张表便是数学模型，计算机的主要操作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的数据结构。

［**例 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，我们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		系统分析员	总工	羽毛球、乒乓球
990002		高级程序员	一组组员	
990003		程序员	一组组员	
990004		程序员	一组组员	
990005		高级程序员	一组组长	网球、乒乓球
990006		高级程序员	二组组长	网球
990007		程序员	二组组员	
990008		程序员	二组组员	
990009		程序员	二组组员	
990010		程序员	二组组员	

(a)

系统分析员	1
高级程序员	2，5，6
程序员	3，4，7，8，9，10

(b)

第一组	5，2，3，4
第二组	6，7，8，9，10

(c)

图 1.1　程序中的数据结构

(a) 员工信息表；(b) 职称索引表；(c) 组别索引表

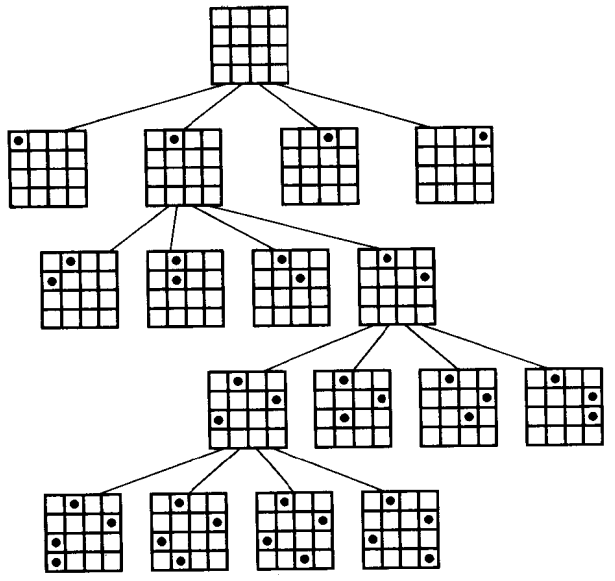

图 1.2　四皇后问题中隐含的状态树

［例 3］ 交通咨询问题。在交通咨询系统中，我们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)，图中的边可以用弧线来表示，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。

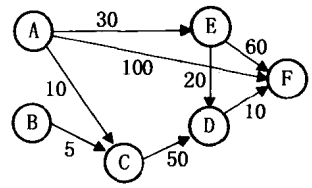

图 1.3 一个表示交通网的图

综上三个例子可见，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、数据库系统等系统程序和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人们的思维能力，通过程序设计的技能训练来促进人们的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。目的也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来解决实际问题的能力。

1.2 数据结构的基本概念

在系统地学习数据结构知识之前，我们先对一些概念和术语赋予确切的含义。

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。如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所接受的数据。

数据元素(**Data Element**)是数据的基本单位。在不同的条件下，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

立含义的最小的单位。

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为结构。根据数据元素间关系的不同特性，通常有下列四类基本的结构：

(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) 图状结构。该结构的数据元素之间存在着多对多的关系，图状结构也称网状结构。

图 1.4 表示上述四类基本结构的关系图。

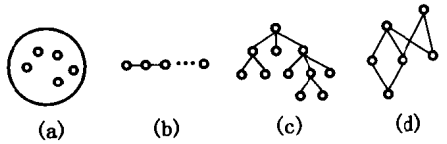

图 1.4 四类基本结构的关系图

(a) 集合；(b) 线性；(c) 树形；(d) 图状

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(又称映象)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，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方法和链式存储方法外，有时为了查找的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

从上面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要素。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上关系的有限集。

例如，在上面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的记录，而元素间关系的集合则可以从不同的视点去建立。我们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；可以按员工的爱好来建立元素间的网状关系，从而形成图状的数据结构，如图1.5所示。

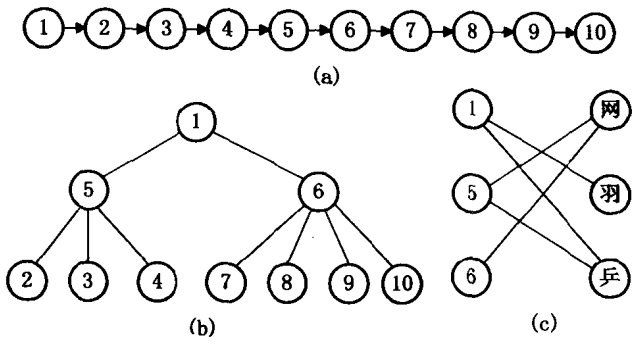

图 1.5 员工信息表中的数据结构

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。原子类型的值是不可分解的，如 Pascal 语言中的整型、实型等标准类型。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所组成的。

1.3.2 抽象数据类型

抽象数据类型(Abstruct 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，各种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，定义一组数据和施与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，而在模块的外部使用的只是抽象的数据及抽象的操作。这也就是面向对象的程序设计方法。

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，因此抽象数据类型一般可以由元素、关系及操作三种要素来进行定义。在本书中，对每一种数据类型都给出它的 ADT 定义。例如，栈的 ADT 定义为

元素：属于同一个数据元素类。

关系：数据元素间呈线性关系。