

21世纪高等学校计算机**专业**实用规划教材

数据结构 (Java语言版)

雷军环 吴名星 编著



清华大学出版社

21 世纪高等学校计算机**专业**实用规划教材

数据结构（Java语言版）

雷军环 吴名星 编著

清华大学出版社
北京

内 容 简 介

本书通过引入学习情境,详细、直观地介绍了数据结构及其算法。全书共9章,内容包括数据结构概述、线性表、堆栈、队列、串、二叉树、图的6种主要数据结构的逻辑结构、存储结构、基本操作及排序和查找算法。全书采用Java语言作为算法描述语言。

本书案例形象生动,层次清晰,讲解深入浅出,可作为计算机及相关专业本、专科“数据结构”课程的教材,也适合各类成人教育相关课程使用,还可以供从事计算机软件开发和应用的工程技术人员阅读、参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构:Java语言版/雷军环,吴名星编著.--北京:清华大学出版社,2015

21世纪高等学校计算机专业实用规划教材

ISBN 978-7-302-41842-9

I. ①数… II. ①雷… ②吴… III. ①数据结构—高等学校—教材 ②JAVA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(2015)第247506号

责任编辑:魏江江 赵晓宁

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:14.75

字 数:356千字

版 次:2015年11月第1版

印 次:2015年11月第1次印刷

印 数:1~2000

定 价:29.50元

产品编号:066093-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机专业课程领域,以专业基础课为主、专业课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 反映计算机学科的最新发展,总结近年来计算机专业教学的最新成果。内容先进,充分吸收国外先进成果和理念。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,融合先进的教学思想、方法和手段,体现科学性、先进性和系统性,强调对学生实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。专业基础课和专业课教材配套,同一门课程有针对不同层次、面向不同应用的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机专业实用规划教材

联系人: 魏江江 weijj@tup.tsinghua.edu.cn

前 言

数据结构知识是计算机科学教育的一个基本组成部分,其他许多计算机科学领域都构建在这个基础之上。对于想从事实际的软件设计、实现、测试和维护工作的读者而言,掌握基本数据结构的知识是非常必要的。该领域的知识将对一个人的编程能力产生极深的影响,它介绍在软件开发过程中如何建立一个合理高效的程序。由于“数据结构”是一门实践性较强而理论知识较为抽象的课程,目前很多学生在学完了这门课后,还是不知道如何运用所学的知识解决实际问题的情况,针对这种情况本书进行了精心的设计。本书主要特点如下所示。

1. 基于典型任务

各章都通过典型任务引出问题,通过典型任务设立学习情境。所有典型任务都是经过精心筛选和设计的与生活紧密相连、生动直观、难易适中的实际问题,可以让学生先思考如何利用以往所学的知识去解决该问题,然后再由教师分析教材上是如何运用数据结构的理论来解决同一问题的,让学生深刻体会到所学数据结构在程序中的作用和使用方法,从而真正体会到“程序=数据结构+算法”的真正含义。

2. 基于问题求解过程

本书除第1章外,所有其他章都是按照“问题提出→认识逻辑结构→实现逻辑结构→应用逻辑结构”这样一个完整问题求解过程来组织内容的。也就是说,对于每一个实际的问题,首先明确数据元素及数据元素之间的逻辑关系,即逻辑结构;其次要理解这些数据元素在计算机中的存储结构以及基于这种存储结构对数据元素的基本操作(即算法),并用Java语言将数据结构和算法转换为能够直接运行的程序代码;最后使用已经实现的逻辑结构解决实际问题。

3. Java 语言描述

相比于很多数据结构的教材用C语言描述,本教材的算法采用面向对象编程语言Java进行编写,接口的定义、类的实现都严格按Java语言规范进行编写,这不仅有助于学生学会用面向对象的语言来描述数据结构的算法,更有助于学生理解数据结构理论在实际开发中的具体应用。

本书是对编者2009年出版的《数据结构(C#语言版)》(清华大学出版社)的一次全面升级,组织思路更加清晰,代码更加优化。在本书的编写过程中,清华大学出版社的广大员工为本书的修订和出版做了大量的工作,在此向他们表示感谢。

尽管编者在写作过程中非常认真和努力,但由于编者水平有限,书中难免存在错误和不足之处,恳请广大读者批评指正。如果您对本书有什么意见、问题或想法,欢迎您通知编者,编者将不胜感激。编者 E-mail: 30898045@qq.com。

编 者

2015年9月

目 录

第 1 章 绪论	1
1.1 引言	1
1.1.1 从问题到程序的基本过程	1
1.1.2 什么是数据结构	4
1.2 数据结构基本概念	5
1.2.1 数据	5
1.2.2 数据元素	5
1.2.3 数据项	5
1.3 逻辑结构与存储结构	6
1.3.1 数据的逻辑结构	6
1.3.2 数据的存储结构	7
1.4 认识算法	8
1.4.1 算法的定义	8
1.4.2 算法的特征	8
1.4.3 算法性能分析与度量	9
1.5 抽象数据类型	10
本章小结	10
综合练习	11
第 2 章 线性表	13
2.1 认识线性表	13
2.1.1 线性表的逻辑结构	14
2.1.2 线性表的基本操作	14
2.1.3 线性表的抽象数据类型	15
2.2 线性表的实现	15
2.2.1 用顺序表实现线性表	15
2.2.2 用单链表实现线性表	21
2.2.3 用双向链表实现线性表	30
2.2.4 用循环链表实现线性表	37
2.3 线性表的应用	37

2.3.1	用顺序表实现约瑟夫环	37
2.3.2	用单链表实现约瑟夫环	40
2.3.3	用 Java 类实现约瑟夫环	40
2.3.4	独立实践	41
2.4	度量不同存储结构的算法效率	42
2.4.1	分析顺序表的算法效率	42
2.4.2	分析单链表的算法效率	43
	本章小结	44
	综合练习	44
第 3 章	堆栈	46
3.1	认识堆栈	46
3.1.1	堆栈的逻辑结构	47
3.1.2	堆栈的基本操作	47
3.1.3	堆栈的抽象数据类型	48
3.2	堆栈的实现	48
3.2.1	用顺序栈实现堆栈	48
3.2.2	用链栈实现堆栈	52
3.3	堆栈的应用	55
3.3.1	用顺序栈实现迷宫路径搜索问题的求解	55
3.3.2	用链式栈实现迷宫路径搜索问题的求解	59
3.3.3	用 Java 类库实现迷宫路径搜索问题的求解	59
3.3.4	独立实践	60
	本章小结	61
	综合练习	61
第 4 章	解决队列的编程问题	63
4.1	认识队列	63
4.1.1	队列的逻辑结构	64
4.1.2	队列的基本操作	64
4.1.3	队列的抽象数据类型	64
4.2	队列的实现	65
4.2.1	用顺序队列实现队列	65
4.2.2	用链队列实现队列	69
4.3	队列的应用	73
4.3.1	用顺序队列实现银行排队叫号服务	73
4.3.2	用链队列实现银行排队叫号服务	75
4.3.3	用 Java 类库实现银行排队叫号服务	76
4.3.4	独立实践	76

本章小结	77
综合练习	77
第 5 章 串	79
5.1 认识串	80
5.1.1 串的逻辑结构	80
5.1.2 串的基本操作	81
5.1.3 串的抽象数据类型	81
5.2 Java 的字符串类	82
5.2.1 Java 中的字符串类 String	82
5.2.2 Java 中的字符串类 StringBuilder 和 StringBuffer	88
5.3 串的应用	91
5.3.1 用串解决“以一敌百”游戏的编程	91
5.3.2 独立实践	94
本章小结	94
综合练习	95
第 6 章 二叉树	96
6.1 认识二叉树	97
6.1.1 二叉树的逻辑结构	97
6.1.2 二叉树的基本操作	100
6.1.3 二叉树的抽象数据类型	100
6.2 二叉树的实现	101
6.2.1 二叉树的顺序存储	101
6.2.2 二叉树的链式存储	102
6.3 二叉树的遍历方法及递归实现	106
6.4 二叉树的应用	111
6.4.1 哈夫曼树的基本概念	111
6.4.2 使用哈夫曼树的算法求报文字符编码	113
6.4.3 报文传输编码的实现	115
6.4.4 独立实践	121
本章小结	121
综合练习	122
第 7 章 图	123
7.1 认识图	123
7.1.1 图的逻辑结构	123
7.1.2 图的基本操作	125
7.1.3 图的抽象数据类型	125

7.2 图的实现	126
7.2.1 用邻接矩阵实现图	126
7.2.2 用邻接表实现图	132
7.2.3 图遍历算法的实现	139
7.2.4 图最短路径算法的实现	146
7.3 图的应用	152
7.3.1 用邻接矩阵解决高速公路交通网的编程	152
7.3.2 用邻接表解决高速公路交通网的编程	156
7.3.3 独立实践	159
本章小结	159
综合练习	159
第8章 排序	161
8.1 认识排序	161
8.1.1 排序概念	162
8.1.2 排序的分类	162
8.2 插入排序	163
8.2.1 直接插入排序	163
8.2.2 希尔排序	166
8.3 选择排序	168
8.3.1 直接选择排序	168
8.3.2 堆排序	169
8.4 交换排序	174
8.4.1 冒泡排序	174
8.4.2 快速排序	175
8.5 归并排序	178
8.6 基数排序	181
8.7 排序的应用	184
8.7.1 编程实现第29届奥运会奥运奖牌的排名	184
8.7.2 独立实践	189
本章小结	190
综合练习	191
第9章 查找	192
9.1 认识查找	193
9.2 线性表查找技术	193
9.2.1 顺序查找	193
9.2.2 二分查找	196
9.2.3 分块查找	199

9.3 树表查找技术	202
9.3.1 认识二叉排序树	202
9.3.2 构建二叉排序树	203
9.4 哈希表查找技术	207
9.4.1 认识哈希表	207
9.4.2 构造哈希函数	209
9.4.3 解决哈希冲突	210
9.4.4 哈希表查找的算法实现	212
9.4.5 分析哈希表的性能	214
9.5 查找的应用	215
9.5.1 编程实现查找和管理某公司员工信息	215
9.5.2 独立实践	219
本章小结	220
综合练习	220
参考文献	222

第 1 章

绪 论

1946 年,美国军方为了解决计算大量数据的难题,发明了第一台计算机。如今,计算机的应用不再局限于科学计算,更多地用于控制、管理、数据处理等非数值计算的处理工作。计算机加工处理的对象有数值、字符、表格、图形声音、图像等各种不同类型的数据,如何在计算机中表示和存储数据成为计算机科学研究的主要内容之一。分析待处理的数据特性以及各处理数据之间存在的关系,这就是“数据结构”这门学科形成和发展的背景。

1.1 引 言

1.1.1 从问题到程序的基本过程

在计算机发展的初期,人们使用计算机的目的主要是处理数值计算问题。处理数值计算问题时,通常是先从具体问题中抽象出要运算的数据,然后设计对数据进行计算的算法,最后编写程序对数据进行运算并输出结果,一个程序通常由数据输入、数据处理和数据输出三部分组成。

例如,一个能进行加、减、乘、除运算的计算器,首先确定要操作的数据是两个数字,对数据操作的算法是加、减、乘、除运算,然后编写程序用两个变量接受两个数字,对它们实现运算,并将运算结果输出。下面是用 Java 语言实现的计算器。

```
public class Calculator {  
    public static void main(String[] args){
```

```
        Scanner sc = new Scanner(System.in);  
        int x = sc.nextInt();  
        int y = sc.nextInt();
```

将输入数据存储在变量中

```
        System.out.println("两个数的和是: " + Calculator.add(x, y));  
        System.out.println("两个数的差是: " + Calculator.minus(x, y));  
        System.out.println("两个数的积是: " + Calculator.multiply(x, y));  
        System.out.println("两个数的商是: " + Calculator.divide(x, y));  
        sc.close();
```

调用对数据运算算法的实现方法对数据进行处理,并将运算结果输出

```
}
```

```
public static int add(int x, int y){
    return x + y;
}
public static int minus(int x, int y){
    return x - y;
}
public static int multiply(int x, int y){
    return x * y;
}
public static int divide(int x, int y){
    return x / y;
}
```

将对两个数进行运算的
算法实现定义在方法中

由于数值计算问题所涉及的计算对象是简单的数据类型,如上述计算器中的数据是整型数据,所以程序设计者的主要精力集中于程序设计的技巧上,而无须重视数据的特性及数据间相关的关系。随着计算机应用领域的扩大和软硬件的发展,非数值计算问题显得越来越重要。据统计,当今处理非数值计算性问题占用了 90% 以上的机器时间。

例如,某公司一个知名企业的电话号码信息表,如表 1.1 所示。该公司想开发一个查询知名企业服务电话号码的程序,将表中的数据存入计算机,当使用者输入任意一个企业名称,若该企业已注册其服务电话号码,则迅速找到其电话号码;否则,指出没有该企业的服务电话号码。

表 1.1 知名企业服务电话号码信息表

序 号	企 业 名 称	服 务 电 话
1	索尼(SONY)	800-820-9000
2	惠普(HP)	800-820-2255
3	联想(Lenovo)	800-810-8888
4	海尔(Haier)	4006-999-999
⋮	⋮	⋮

这类问题涉及的处理对象不再是简单的数据类型,数据元素之间的关系无法直接用数学公式加以描述,需要发现更加合适的解决问题的方法。解决非数值问题通常经历两个阶段。

(1) 逻辑分析。该阶段主要完成两个工作:

- ① 分析问题的数据特性及数据之间的关系。电话号码信息表是多个企业电话号码信息的序列集合,每个企业的电话号码信息由企业名称和服务电话两项数据组成。
- ② 分析解决问题的逻辑算法。根据问题的要求,要实现添加和查找功能的算法,该算法根据给定的一个企业名称,从电话号码序列表中查找其对应的电话号码。

(2) 物理实现。该阶段主要完成两个工作:

- ① 确定数据及其关系在计算机中的存储方式。在 Java 中,首先创建一个类用来表示企业电话号码信息,它包含企业名称和服务电话两个属性,然后创建一个数组,数组中的每一项都是该类的实例。其中,类是对操作数据的物理实现;数组是对数据关系的物理实现。
- ② 编程实现物理逻辑算法。在数据及其关系的存储结构上,使用高级语言的语法规

范,实现逻辑分析阶段设计的逻辑算法。

查找企业电话服务电话的 Java 程序代码如下。

```
import java.util.ArrayList;
import java.util.Scanner;
class PhoneInfo {
    String companyName;
    String companyPhone;

    public PhoneInfo(String companyName, String companyPhone) {
        this.companyName = companyName;
        this.companyPhone = companyPhone;
    }
}
public class PhoneService {
```

定义一个动态数组,保存企业服务电话号码信息表

```
    ArrayList<PhoneInfo> phonelst = new ArrayList<PhoneInfo>();
```

//向动态数组中添加企业电话号码信息

```
public void addPhone(PhoneInfo phoneInfo) {
    phonelst.add(phoneInfo);
}
```

//获取指定企业的电话号码信息

```
public PhoneInfo getPhone(String companyName) {
    PhoneInfo phoneInfo = null;
    for (PhoneInfo x : phonelst) {
        if (x.companyName.equals(companyName)) {
            phoneInfo = x;
            break;
        }
    }
    return phoneInfo;
}
```

//主方法,对算法功能进行测试

```
public static void main(String[] args) {
    PhoneService phoneService = new PhoneService();
    char continueflag = 'y';
    char seleflag;
    Scanner sc = new Scanner(System.in);
    String name, phone;
    while (continueflag == 'y') {
        System.out.println("请输入操作选项:");
        System.out.println("1. 添加企业服务电话");
        System.out.println("2. 查询企业服务电话");
        seleflag = sc.nextLine().toCharArray()[0];
        switch (seleflag) {
            //添加企业服务电话信息
            case '1':
                System.out.print("请输入企业名称:");
```

```

        name = sc.nextLine();
        System.out.print("请输入企业电话:");
        phone = sc.nextLine();
        phoneService.addPhone(new PhoneInfo(name, phone));
        break;
//查询企业服务电话信息
case '2':
    System.out.print("请输入查询企业名称:");
    name = sc.nextLine();
    System.out.println(name + "公司的服务电话是:"
        + phoneService.getPhone(name).companyPhone);
    break;
default:
    continueflag = 'n';
    break;
}
}
sc.close();
}
}

```

通过对上面电话号码问题的分析,总结出用计算机求解问题的基本步骤,如图 1.1 所示。

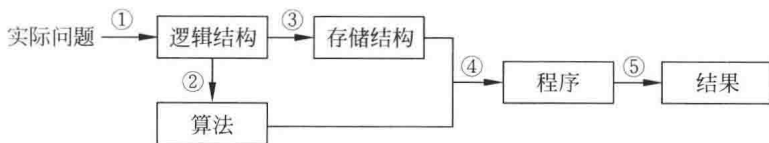


图 1.1 用数据结构求解问题的基本步骤

从图 1.1 可以看出,问题的求解步骤可分解为:

- ① 确定数据及数据之间的关系,对问题所涉及数据的逻辑结构进行设计。
- ② 分析对数据的逻辑结构可能进行的各种操作,对解决问题的算法进行设计。
- ③ 用一种存储结构在计算机内部表示数据的逻辑结构。
- ④ 根据数据的存储结构及设计的算法编写计算机程序。
- ⑤ 运行程序得到问题期望的结果。

1.1.2 什么是数据结构

通过从问题到程序的过程分析,可以知道计算机的程序是对信息进行加工处理。在大多数情况下,这些信息并不是没有组织,信息(数据)之间往往具有重要的结构关系,这就是数据结构的内容。数据的结构,直接影响算法的选择和效率。

综上所述,数据结构是一门研究非数值计算的程序设计问题中计算机的操作对象及其关系和操作等的学科。它包括三个组成成分:数据的逻辑结构、数据的存储结构和数据的运算结构。一个数据结构是由数据元素依据某种逻辑联系组织起来的,对数据元素间逻辑关系的描述称为数据的逻辑结构;数据必须在计算机内存储,数据的存储结构是数据结构

的实现形式,是其在计算机内的表示,一个逻辑数据结构可以有多种存储结构,且各种存储结构影响数据处理的效率;此外,讨论一个数据结构必须同时讨论在该数据结构上执行的运算才有意义。

在许多类型的程序的设计中,数据结构的选择是设计的一个基本要素。许多大型系统的构造经验表明,系统实现的困难程度和系统构造的质量都依赖于是否选择了最优的数据结构。通常,确定了数据结构后,算法就容易得到了;有时,事情也会反过来,根据特定算法来选择数据结构与之适应。不论哪种情况,选择合适的数据结构都是非常重要的。

1.2 数据结构基本概念

在系统地学习数据结构知识之前,先对一些基本概念和术语赋予确切的定义。

1.2.1 数据

数据是对客观事物的符号表示,在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称,它是计算机程序加工的“原料”。正所谓“巧妇难为无米之炊”,强大的程序要“有米下锅”才可以完成强大的功能,否则就是无用的程序,这个“米”就是数据。

例如,一个学生的学习成绩,一个编译程序或文字处理程序的处理字符串,这些都是数据。对计算机科学而言,数据的含义极为广泛,如图像、声音等都可以通过编码而归之于数据的范畴。数据是程序中最基本和最重要的处理对象,数据结构研究的是数据的组织及在计算机中存储和处理方式的学科。

1.2.2 数据元素

数据元素是组成数据的、有一定意义的基本单位,在计算机中通常作为整体处理,也被称为记录。

图 1.2 所示的是学生成绩数据,一个学生的成绩数据用一个数据元素表示。在知名企业服务电话号码信息表中,一个企业的服务电话信息也用一个数据元素表示。

学号	姓名	语文	数学	C语言
6201001	张三	85	54	92
6201002	李四	92	84	64
6201003	王五	87	74	73
6201004				
⋮				

一个数据元素

一个数据项

整个表记录的是学生成绩数据,单个学生的成绩是其中的一个数据元素

图 1.2 数据元素和数据项

1.2.3 数据项

数据项是组成数据元素的基本单位,一个数据元素可以由若干个数据项组成。

在图 1.2 中,一个学生成绩数据元素由学号、姓名、语文、数学、C 语言 5 个数据项组成。

在知名企业服务电话号码信息表中,一个企业服务电话信息是由企业名称和服务电话两个数据项组成。

注意: 数据元素是组成数据的基本单位,数据项是组成数据元素的基本单位,数据项是不可再分割的最小数据单位。在真正解决问题时,数据元素才是真正进行访问和处理的基本单位。例如,在讨论知名企业服务电话号码查询程序时,讨论的是企业服务电话号码表中一行数据即一个数据元素的信息,而不会针对其中单独的数据项企业名称或服务电话去分析,分析的是某企业的电话是多少,会同时涉及企业名称和服务电话两个数据项的信息。

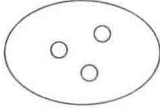
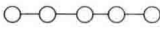
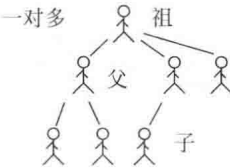
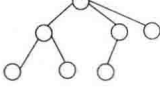
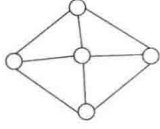
1.3 逻辑结构与存储结构

根据视角不同,把数据的结构分为逻辑结构和存储结构。

1.3.1 数据的逻辑结构

数据的逻辑结构是从逻辑的角度(即数据间的联系和组织方式)来观察数据、分析数据,与数据的存储位置无关。根据数据元素之间关系的不同特性,通常有下列 4 类基本结构,如表 1.2 所示。

表 1.2 数据元素之间关系的类型

关系名称	特 征	示 例	示 意 图
集合	结构中的数据元素之间除了“同属于一个集合”的关系外,别无其他关系,元素间为松散的关系	同属色彩集合 	
线性结构	数据元素间存在严格的一对一关系	如学生信息表中的各元素	
树形结构	数据元素间为严格的一对多关系	一对多 	
图形结构	数据元素间为多对多关系	多对多  公路交通图	

如图 1.2 所示的学生成绩表为学生成绩数据的逻辑结构,这个表中的任何一个数据元素,只有一个直接前趋和一个直接后继(前趋后继就是前相邻后相邻的意思),整个表只有一