



21st CENTURY

实用规划教材

21世纪全国高职高专
计算机系列实用规划教材

数据结构

(C#语言描述)

主 编 陈 广
副主编 林 沅 邹园园 雷运理

内容特点:

- 配套10小时视频教程,以动画方式对算法及数据结构进行讲解,使学习变得轻松自如
- 深入C#类库进行剖析,提升编写高性能应用程序的能力
- 减少数学公式,全部使用可运行代码,降低学习难度
- 配套完整的实训指导及综合实训



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国高职高专计算机系列实用规划教材

数据结构 (C#语言描述)

主 编 陈 广

副主编 林 沅 邹园园 雷运理



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书使用 C#语言及面向对象的方法讲解了数据结构的基础知识,并针对数据结构中的难点及关键点制作了配套的视频教程,使用动画加讲解的方式对数据结构及算法进行了详细的介绍。

全书共分 10 章,第 1~6 章主要介绍线性表、栈、队列、串、树、图这些基本的数据结构;第 7、8 章介绍查找和排序算法及哈希表;第 2~9 章每章的结尾部分均配备了实训指导,以加深读者对各个章节理论知识的理解;第 10 章是综合实训部分,通过实例演示数据结构及算法在程序中的应用。配书光盘内容为与本书配套使用的视频教程。

本书体系新颖,层次清晰,特别注重可读性和实用性,并结合数据结构知识深入 C#类库进行解析。全书通俗易懂、由浅入深,不但能使读者了解数据结构知识,而且对 C#语言也会有更进一步的认识。

本书既可以作为高等职业院校计算机及相关专业的教材,也适合作为自学教材以及 C#程序开发人员的参考书。

图书在版编目(CIP)数据

数据结构(C#语言描述)/陈广主编. —北京:北京大学出版社,2009.3

(21 世纪全国高职高专计算机系列实用规划教材)

ISBN 978-7-301-14475-6

I. 数… II. 陈… III. ①数据结构—高等学校:技术学校—教材②C 语言—程序设计—高等学校:技术学校—教材 IV. TP311.12 TP312

中国版本图书馆 CIP 数据核字(2009)第 005911 号

书 名: 数据结构(C#语言描述)

著作责任者: 陈 广 主编

责任编辑: 王显超

标准书号: ISBN 978-7-301-14475-6/TP·0982

出版者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.com>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电子邮箱: pup_6@163.com

印 刷 者: 北京飞达印刷有限责任公司

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 18.5 印张 420 千字

2009 年 3 月第 1 版 2009 年 3 月第 1 次印刷

定 价: 38.00 元(含 1 张 CD)

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有 侵权必究

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

21 世纪全国高职高专计算机系列实用规划教材

专家编审委员会

主 任 刘瑞挺

副主任 (按拼音顺序排名)

陈玉国 崔锁镇 高文志 韩希义

黄晓敏 魏 峥 谢一风 张文学

委 员 (按拼音顺序排名)

安志远 丁亚明 杜兆将 高爱国 高春玲 郭鲜凤

韩最蛟 郝金镇 黄贻彬 季昌武 姜 力 李晓桓

连卫民 刘德军 刘德仁 刘辉珞 栾昌海 罗 毅

慕东周 彭 勇 齐彦力 沈凤池 陶 洪 王春红

闻红军 武凤翔 武俊生 徐 红 徐洪祥 徐受容

许文宪 严仲兴 杨 武 易永红 于巧娥 袁体芳

张 昕 赵 敬 赵润林 周朋红 訾 波 周 奇

信息技术的职业化教育

(代丛书序)

刘瑞挺/文

北京大学出版社第六事业部组编了一套《21 世纪全国高职高专计算机系列实用规划教材》。为此,制订了详细的编写目的、丛书特色、内容要求和风格规范。在内容上强调面向职业、项目驱动、注重实例、培养能力;在风格上力求文字精练、图表丰富、脉络清晰、版式明快。

一、组编过程

2004 年 10 月,第六事业部开始策划这套丛书,分派编辑深入各地职业院校,了解教学第一线的情况,物色经验丰富的作者。2005 年 1 月 15 日在济南召开了“北大出版社高职高专计算机规划教材研讨会”。来自 13 个省、41 所院校的 70 多位教师汇聚一堂,共同商讨未来高职高专计算机教材建设的思路和方法,并对规划教材进行了讨论与分工。2005 年 6 月 13 日在苏州又召开了“高职高专计算机教材大纲和初稿审定会”。编审委员会委员和 45 个选题的主、参编,共 52 位教师参加了会议。审稿会分为公共基础课、计算机软件技术专业、计算机网络技术专业、计算机应用技术专业 4 个小组对稿件逐一进行审核。力争编写出一套高质量的、符合职业教育特点的精品教材。

二、知识结构

职业生涯的成功与人们的知识结构有关。以著名侦探福尔摩斯为例,作家柯南道尔在“血字的研究”中,对其知识结构描述如下:

- ◆ 文学知识——无;
- ◆ 哲学知识——无;
- ◆ 政治学知识——浅薄;
- ◆ 植物学知识——不全面。对于药物制剂和鸦片却知之甚详。对毒剂有一般了解,而对于实用园艺却一无所知;
- ◆ 化学知识——精深;
- ◆ 地质学知识——偏于应用,但也有限。他一眼就能分辨出不同的土质。根据裤子上泥点的颜色和坚实程度就能说明是在伦敦什么地方溅上的;
- ◆ 解剖学知识——准确,却不系统;
- ◆ 惊险小说知识——很渊博。似乎对近一个世纪发生的一切恐怖事件都深知底细;
- ◆ 法律知识——熟悉英国法律,并能充分实用;
- ◆ 其他——提琴拉得很好,精于拳术、剑术。

事实上,我国唐朝名臣狄仁杰,大宋提刑官宋慈,都有类似的知识结构。审视我们自己,每人的知识结构都是按自己的职业而建构的。因此,我们必须面向职场需要来设计教材。

三、职业门类

我国的职业门类分为 18 个大类：农林牧渔、交通运输、生化与制药、地矿与测绘、材料与能源、土建水利、制造、电气信息、环保与安全、轻纺与食品、财经、医药卫生、旅游、公共事业、文化教育、艺术设计传媒、公安、法律。

每个职业大类又分为二级类，例如电气信息大类又分为 5 个二级类：计算机、电子信息、通信、智能控制、电气技术。因此，18 个大类共有 75 个二级类。

在二级类的下面，又有不同的专业。75 个二级类共有 590 种专业。俗话说：“三百六十行，行行出状元”，现代职业仍在不断涌现。

四、IT 能力领域

通常信息技术分为 11 个能力领域：规划的能力、分析与设计 IT 解决方案的能力、构建 IT 方案的能力、测试 IT 方案的能力、实施 IT 方案的能力、支持 IT 方案的能力、应用 IT 方案的能力、团队合作能力、文档编写能力、项目管理能力以及其他能力。

每个能力领域下面又包含若干个能力单元，11 个能力领域共有 328 个能力单元。例如，应用 IT 方案能力领域就包括 12 个能力单元。它们是操作计算机硬件的能力、操作计算机软件包的能力、维护设备与耗材的能力、使用计算机软件包设计机构文档的能力、集成商务计算机软件包的能力、操作文字处理软件的能力、操作电子表格应用软件的能力、操作数据库应用软件的能力、连接到互联网的能力、制作多媒体网页的能力、应用基本的计算机技术处理数据的能力、使用特定的企业系统以满足用户需求的能力。

显然，不同的职业对 IT 能力有不同的要求。

五、规划梦想

于是我们建立了一个职业门类与信息技术的平面图，以职业门类为横坐标、以信息技术为纵坐标。每个点都是一个函数，即 IT(Professional)，而不是 IT+Professional 单纯的相加。针对不同的职业，编写它所需要的信息技术教材，这是我们永恒的主题。

这样组合起来，就会有 $IT((328) \times (Pro(590)))$ ，这将是一个非常庞大的数字。组织这么多的特色教材，真的只能是一个梦想，而且过犹不及。能做到 $IT((11) \times (Pro(75)))$ 也就很不容易了。

因此，我们既要在宏观上把握职业门类的大而全，也要在微观上选择信息技术的少而精。

六、精选内容

在计算机科学中，有一个统计规律，称为 90/10 局部性原理(Locality Rule)：即程序执行的 90% 代码，只用了 10% 的指令。这就是说，频繁使用的指令只有 10%，它们足以完成 90% 的日常任务。

事实上，我们经常使用的语言文字也只有总量的 10%，却可以完成 90% 的交流任务。同理，我们只要掌握了信息技术中 10% 频繁使用的内容，就能处理 90% 的职业化任务。

有人把它改为 80/20 局部性原理，似乎适应的范围更广些。这个规律为编写符合职业教育需要的精品教材指明了方向：坚持少而精，反对多而杂。

七、职业本领

以计算机为核心、贴近职场需要的信息技术已经成为大多数人就业的关键本领。职业教育的目标之一就是培养学生过硬的 IT 从业本领，而且这个本领必须上升到职业化的高度。

职场需要的信息技术不仅是会使用键盘、录入汉字，而且还要提高效率、改善质量、降低成本。例如，两位学生都会用 Office 软件，但他们的工作效率、完成质量、消耗成本可能有天壤之别。领导喜欢谁？这是不言而喻的。因此，除了道德品质、工作态度外，必须通过严格的行业规范和个人行为规范，进行职业化训练才能养成正确的职业习惯。

我们肩负着艰巨的历史使命。我国人口众多，劳动力供大于求的矛盾将长期存在。发展和改革职业教育，是我国全面建设小康社会进程中一项艰巨而光荣的任务，关系到千家万户人民群众的切身利益。职业教育和高技能人才在社会主义现代化建设中有特殊的作用。我们一定要兢兢业业、不辱使命，把这套高职高专教材编写好，为我国职业教育的发展贡献一份力量。

刘瑞挺教授 曾任中国计算机学会教育培训委员会副主任、教育部理科计算机科学教学指导委员会委员、全国计算机等级考试委员会委员。目前担任的社会职务有：全国高等院校计算机基础教育研究会副会长、全国计算机应用技术证书考试委员会副主任、北京市计算机教育培训中心副理事长。

前言

数据结构是计算机科学中最重要的课程之一，它对程序设计思想的建立和提升有着重要意义，既可以为后续计算机课程学习奠定扎实的基础，又能提高读者分析和解决问题的能力，并且能够显著地减少读者在学习新技术时学习曲线的坡度。

1. 本书特点

(1) 减少数学公式的使用。

数学对于相当一部分高职生来说并不是那么精通，减少数学公式的使用可以有效降低数据结构的学习门槛。使用其他方式也可以把数据结构描述得很清楚。

(2) 抛弃伪代码。

本书的所有代码均为可运行代码，它可以让读者看到实实在在的结果，也可以通过断点调试、单步运行、修改参数等方式查看数据变化，更深刻地理解数据结构。

(3) 配套视频教程。

笔者在多年的教学实践中发现，只要方法得当，算法的讲解比代码的讲解更容易让人接受。使用动画的方式并配以适当讲解所产生的效果是文字无法替代的。为此笔者专门针对各种数据结构和算法制作了视频教程，以有效降低数据结构的难度。

(4) 深入 C# 类库。

本书所介绍的很多数据结构在 C# 集合类中都有实现，如果对这些集合类没有深刻的理解，是很难写好程序的。本书以数据结构为切入点，深入到 C# 类库中剖析部分常用集合类的实现原理，以帮助读者提高 C# 语言的编程能力。

2. 本书适用对象

阅读本书需要有较好的 C# 语言基础，但即使通过其他语言学习数据结构，观看本书的配套视频也会有所收获。

本书非常适合作为高职高专计算机及相关专业的教材，同时也适合作为自学教材以及 C# 程序开发人员的参考书。

3. 本书约定

本书所配套的视频只有配合书本使用方能发挥最大的作用。在阅读本书时如果发现 ，则应先按图标后面的编号观看视频，然后再阅读后面的段落。

如果不知道如何运行本书的示例代码，可观看光盘根目录下的【如何运行本书代码.swf】视频文件。

4. 学时分配

建议课程安排 102 课时，其中，理论教学为 36 课时，实验教学为 36 课时，综合实训为 30~60 学时。

由于编者水平所限,加之时间仓促,书中疏漏之处在所难免,如果读者有任何疑问或意见,可通过电子邮件(cgbluesky@126.com)和编者联系或把意见留言在笔者的博客(<http://cgbluesky.blog.163.com>)上。本书勘误会及时在博客上发布。

编 者

2008 年 10 月

目 录

第 1 章 绪论.....	1	3.1.3 栈的应用.....	54
1.1 什么是数据结构.....	1	3.1.4 双向栈.....	55
1.1.1 数据结构的产生与发展.....	2	3.2 队列.....	56
1.1.2 数据和数据结构.....	3	3.2.1 队列的概念及操作.....	56
1.1.3 数据的逻辑结构.....	3	3.2.2 循环队列.....	57
1.1.4 数据的存储结构.....	4	3.2.3 System.Collections.Queue.....	58
1.1.5 数据类型.....	6	3.3 本章小结.....	61
1.2 算法与算法分析.....	6	3.4 实训指导：虚拟循环队列.....	62
1.2.1 算法.....	6	3.5 习题.....	67
1.2.2 算法设计的目标.....	6	第 4 章 串.....	69
1.2.3 算法的时间复杂度.....	7	4.1 串的基本概念.....	69
1.2.4 算法的空间复杂度.....	10	4.2 String.....	69
1.3 本章小结.....	10	4.3 System.Text.StringBuilder.....	70
1.4 习题.....	11	4.4 串的模式匹配.....	72
第 2 章 线性表.....	13	4.4.1 Brute-Force 算法.....	72
2.1 线性表的定义.....	13	4.4.2 KMP 算法.....	74
2.2 线性表的顺序存储结构——顺序表.....	13	4.5 本章小结.....	83
2.2.1 顺序表的特点.....	13	4.6 实训指导：求最长公共子串.....	83
2.2.2 数组.....	14	4.7 习题.....	85
2.2.3 System.Collections.ArrayList.....	16	第 5 章 树.....	88
2.2.4 类型安全.....	22	5.1 树的基本概念.....	89
2.3 线性表的链式存储结构——链表.....	23	5.1.1 树的定义.....	89
2.3.1 单向链表.....	24	5.1.2 树的表示.....	89
2.3.2 循环链表.....	29	5.1.3 树的基本术语.....	90
2.3.3 双向链表.....	34	5.2 二叉树.....	91
2.4 本章小结.....	38	5.2.1 二叉树的基本概念.....	91
2.5 实训指导：虚拟线性表.....	39	5.2.2 二叉树的存储结构.....	92
2.6 习题.....	48	5.3 二叉树的遍历.....	94
第 3 章 栈和队列.....	51	5.3.1 二叉树的深度优先遍历.....	94
3.1 栈.....	51	5.3.2 二叉树的广度优先遍历.....	98
3.1.1 栈的概念及操作.....	51	5.4 线索二叉树.....	100
3.1.2 System.Collections.Stack.....	52	5.4.1 线索二叉树的定义.....	100

5.4.2 中序线索二叉树.....	101	7.2 顺序查找.....	179
5.5 树和森林.....	105	7.3 二分查找.....	180
5.5.1 树的存储结构.....	105	7.3.1 二分查找的基本原理.....	180
5.5.2 森林、树、二叉树的相互 转换.....	107	7.3.2 二分查找的算法实现.....	181
5.6 可绘制二叉树的设计.....	109	7.3.3 Array.BinarySearch 方法.....	183
5.6.1 二叉树结点的位置关系.....	110	7.3.4 剖析 System.Collections. SortedList.....	183
5.6.2 接口设计.....	110	7.4 分块查找.....	188
5.6.3 二叉树绘制类的设计.....	112	7.5 二叉查找树.....	188
5.6.4 实现可绘制二叉树.....	115	7.5.1 二叉查找树的定义.....	189
5.7 二叉树画树算法.....	117	7.5.2 二叉查找树的查找.....	189
5.7.1 满二叉树画法.....	117	7.5.3 二叉查找树的插入.....	190
5.7.2 界内画法.....	120	7.5.4 二叉查找树的删除.....	191
5.7.3 最小面积画法.....	121	7.5.5 二叉查找树的代码实现.....	192
5.8 本章小结.....	126	7.6 本章小结.....	196
5.9 实训指导: 虚拟二叉树.....	126	7.7 实训指导: Array.BinarySearch 的使用.....	196
5.10 习题.....	137	7.8 习题.....	199
第 6 章 图.....	139	第 8 章 哈希表.....	201
6.1 基本概念和术语.....	139	8.1 概念引入.....	201
6.2 图的存储结构.....	142	8.2 构造哈希函数的方法.....	204
6.2.1 邻接矩阵表示法.....	142	8.2.1 直接定址法.....	204
6.2.2 邻接表表示法.....	143	8.2.2 数字分析法.....	204
6.3 图的遍历.....	147	8.2.3 除留余数法.....	205
6.3.1 深度优先搜索遍历.....	148	8.3 哈希冲突解决方法.....	205
6.3.2 广度优先搜索遍历.....	150	8.3.1 闭散列法(开放地址法).....	206
6.3.3 非连通图的遍历.....	151	8.3.2 开散列法(链地址法).....	208
6.4 生成树和最小生成树.....	152	8.4 剖析 System.Collections.Hashtable	209
6.4.1 生成树.....	152	8.4.1 Hashtable 的实现原理.....	209
6.4.2 最小生成树.....	153	8.4.2 Hashtable 的代码实现.....	211
6.4.3 普里姆算法.....	153	8.5 剖析 Dictionary<TKey, TValue>	220
6.4.4 克鲁斯卡尔算法.....	157	8.5.1 Dictionary<TKey, TValue>类 实现原理.....	220
6.5 最短路径.....	161	8.5.2 Dictionary<TKey, TValue>的 代码实现.....	224
6.5.1 单源点最短路径.....	161	8.6 本章小结.....	229
6.5.2 所有顶点之间的最短路径.....	165	8.7 实训指导: 虚拟哈希表.....	229
6.6 本章小结.....	168	8.8 习题.....	235
6.7 实训指导: 迷宫最短路径问题.....	168		
6.8 习题.....	176		
第 7 章 查找.....	179		
7.1 查找的基本概念.....	179		

第9章 排序.....237	10.2.1 从初始状态到达目标状态 是否有解..... 258
9.1 排序的基本概念.....237	10.2.2 使用什么方法求解八数码 问题的最优解..... 259
9.2 插入排序.....237	10.2.3 如何避免重复访问一个 状态..... 260
9.2.1 直接插入排序.....237	10.2.4 怎样记录查找路径..... 260
9.2.2 希尔排序.....239	10.2.5 使用什么数据结构表示 棋盘状态..... 261
9.3 交换排序.....241	10.3 设计目标..... 262
9.3.1 冒泡排序.....241	10.4 界面设计..... 263
9.3.2 快速排序.....242	10.5 代码编写..... 263
9.4 选择排序.....244	10.5.1 MoveDirection.cs..... 263
9.4.1 直接选择排序.....244	10.5.2 AIResult.cs..... 263
9.4.2 堆排序.....245	10.5.3 HashHelpers.cs..... 264
9.5 归并排序.....248	10.5.4 SimpleDictionary.cs..... 265
9.5.1 二路归并排序.....248	10.5.5 NumSwitch.cs..... 268
9.5.2 二路归并排序的实现.....249	10.5.6 IEightNumAI.cs..... 270
9.6 本章小结.....250	10.5.7 BFS_AI.cs..... 270
9.7 实训指导：使用 IComparer<T>接口 进行排序.....250	10.5.8 MainForm.cs..... 272
9.8 习题.....255	10.6 调试运行..... 279
第10章 综合实训——八数码问题.....258	10.7 思考与改进..... 279
10.1 什么是八数码问题.....258	参考文献..... 280
10.2 八数码问题的解析.....258	

第1章 绪 论

在面向对象技术中，数据的组织方式对于一个软件的优劣、效率和质量具有举足轻重的作用，实质上程序设计就是对确定的问题构造一种好的结构，并设计一种好的算法，也就是人们常说的“程序设计=数据结构+算法”。因此要编写出一个“好”软件，就必须分析所需处理的对象的特性以及各种对象之间存在的关系。这些问题就是“数据结构”这门学科所要研究的主要问题。

1.1 什么是数据结构

随着计算机的日益普及，人类已经进入以信息化为标志并飞速发展的时代。而数据作为信息的载体，随着计算机技术的发展，需要由计算机处理的数据量随之增大，数据类型也随之增多，而不仅仅局限于单纯的数值数据，更多的是用于表示文字、图像、声音等的非数值数据，在数据类型增多的同时，数据结构也更加复杂。因此，大多数的计算机程序不但要完成运算，还要进行数据的存储、检索和处理。首先通过以下 3 个例子来简单地认识一下数据结构。

【例 1-1】 新生入学时，需要注册每个学生的基本信息，包括学生的班级、姓名、性别、籍贯、出生年月和民族等，并为每个学生分配一个学号，见表 1-1。通过这个表可以看出每个学生的学号是唯一的，并且它们按照一定的顺序进行排列，而这就是一种简单的数学模型，通常称为线性表。

表 1-1 学生信息表

学号	班级	姓名	性别	籍贯	出生年月	民族
07080901	软件 001	陈明	男	广西	1987.7	汉
07080902	软件 001	李德庆	男	广西	1988.8	壮
07080903	软件 001	张基德	男	山西	1987.3	侗
07080904	软件 001	邱仲全	男	海南	1987.5	汉
07080905	软件 001	徐明明	女	广东	1986.12	汉
07080906	软件 001	齐飞鸿	男	贵州	1986.11	苗

【例 1-2】 Windows 操作系统中的文件系统如图 1.1 所示。这是一个层次结构：在结构图中，顶点结点代表整个文件系统，用根目录“我的电脑”表示；它的下一层结点代表各个盘符，如 C:/、D:/、F:/等；再下一层结点代表各盘符的文件目录，如 WINDOWS、Program Files；以此类推，直到底层为可执行文件，如/Program Files/ Microsoft Office/WIN WORD.EXE。此时会发现这样的结构就像一棵倒长的“树”，也称为树结构。



图 1.1 文件系统结构图

【例 1-3】 公路运输问题，假设要从广东运送货物到黑龙江，从路程和经济效益角度考虑该如何运送最节省成本？这时需要建立部分城市的公路交通模型进行分析，如图 1.2 所示。通常这种交通、道路问题的数学模型是一种称为“图”的数据结构。图中每一个顶点表示一个省市，而两个省市之间的连线表示一条通路，没有连线则不能通行。广东到黑龙江的路线可以是广东→上海→辽宁→黑龙江，或者是广东→河南→北京→黑龙江等。

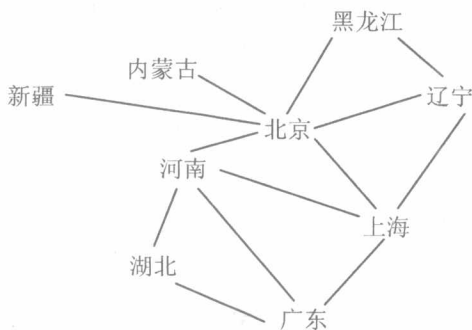


图 1.2 部分城市公路运输网络图

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，而是类似于表、树、图之类的结构。因此，简单地说，数据结构是一门研究在非数值计算的程序设计中计算操作对象以及它们之间的关系和操作的学科。

1.1.1 数据结构的产生与发展

数据结构与程序设计的发展密切相关。迄今为止，程序设计经历了无结构、结构化程序设计，以及现在的面向对象程序设计 3 个阶段。

无结构阶段：从 20 世纪 40 年代至 20 世纪 60 年代，程序设计主要针对科学计算，所涉及的数据对象简单，程序多以算法为中心，程序的设计语言是机器语言或汇编语言。

结构化程序设计阶段：从 20 世纪 60 年代末至 20 世纪 80 年代，这一阶段出现了比较复杂的程序，软件也相对独立，此时人们已经意识到了规范化程序设计的重要性，提出了程序结构模块化。与此同时，计算机开始广泛应用于非数值数据处理领域，操作系统、数

数据库等系统软件的设计也进入方法化时期,人们开始注意到数据表示与操作的结构化,程序中常用的一些数据结构如表、栈、队列、树、图等开始被单独研究。1968年,D.E.Knuth(中文名:高德纳)的著作《计算机编程艺术》的第一卷《基本算法》第一次系统地阐述了数据的逻辑结构和存储结构与其基本操作的设置与实现,对数据结构的发展起到了巨大的推动作用。

面向对象技术兴起于 20 世纪 80 年代初,在面向对象技术中,数据是程序的主体,对象是划分与构造软件系统的基本单位,这与之前的以功能为中心、软件开发采用功能分解的方法有本质的区别。面向对象技术实质上是数据结构概念的自然扩展与延续,数据结构也在随着面向对象技术的发展而发展。

1.1.2 数据和数据结构

1. 数据

数据(Data)是客观事物的符号表示,在计算机科学中是指能输入计算机并能由计算机程序进行处理的符号的总称。例如,数值计算中方程求解程序的处理对象是整数和实数,文字处理程序的处理对象是字符串、而多媒体程序的处理对象是声音、图片和影像等。虽然计算机不能直接处理声音和图像,但它们经过数字化或编码等技术处理后,可以变成能被计算机程序处理的对象,因此,数值、文档、声音、图像等都可以统称为数据。在计算机领域,人们把能够被计算机加工的对象,或者说能够被计算机输入、存储、处理和输出的一切信息都叫做数据。

2. 数据元素

数据元素(Data Element)是数据的基本单位,在程序中通常是作为一个整体来进行考虑和处理的。一个数据元素通常由若干个数据项(Data Item)组成,而数据项是数据不可分割的最小单位。例如,在一个人事档案管理系统中,一个人的信息就是一个数据元素,它由编号、姓名、性别、籍贯、出生年月等数据项组成,这些数据项是不可分割的最小单位。

3. 数据对象

数据对象(Data Object)也称数据记录(Data Record),是性质相同的数据元素的集合,是数据的一个子集。例如,整数数据对象的集合 $I = \{0, \pm 1, \pm 2, \dots\}$, 字母字符数据对象的集合 $C = \{'A', 'B', \dots, 'Z'\}$ 。

4. 数据结构

数据结构(Data Structure)是相互之间存在一种或多种特定关系的数据元素的集合。数据结构这个概念至今还没有一个统一的定义,而本书的定义是说数据结构是由数据对象及其相互之间的关系两大部分组成的,即有关系的数据对象集合才是数据结构。

1.1.3 数据的逻辑结构

数据元素之间的关系称为结构。根据数据元素之间关系的不同特性,通常有下列 4 种基本结构形式。

1. 集合

结构中的数据元素之间除了“属于同一个集合”之外，别无其他关系，如图 1.3 所示。

2. 线性结构

结构中的元素存在一对一的关系，除第一个和最后一个元素外，其他每个数据元素有且仅有一个直接前驱和一个直接后继，如图 1.4 所示。

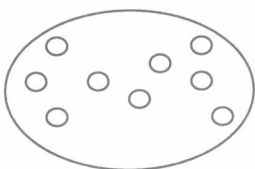


图 1.3 数据的集合关系



图 1.4 数据的线性关系

3. 树结构

数据元素之间存在一对多的关系。一个数据元素可以与一个或多个数据元素存在关系，其结构形式犹如倒立生长的树，如图 1.5 所示。

4. 图结构或网状结构

数据元素之间存在多个多对多的关系。在这种关系中，数据之间的关系不受任何限制，如图 1.6 所示。

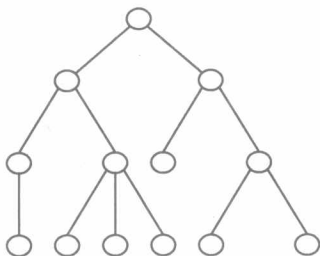


图 1.5 数据的树关系

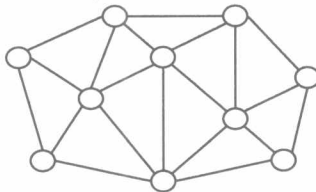


图 1.6 数据的图关系

1.1.4 数据的存储结构

数据结构在计算机中的存储方式称为数据的物理结构，又称存储结构。它包含数据元素及数据元素之间关系的表示，它不同于逻辑结构，是依赖于计算机语言的、具体的。通常在计算机内数据元素用一组连续的二进制位串来表示，位串称为结点。结点之间的关系，即数据元素之间的关系在计算机内有 4 种基本的存储表示方法。

1. 顺序存储结构

顺序存储结构(Sequence Storage Structure)是将逻辑上相邻的结点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表示，这样只需要存储结点的值，不必存储结点之间的关系，这种存储方式称为顺序存储结构。它主要应用于

线性的数据结构,非线性的数据结构也可以通过某种线性化过程后进行顺序存储。如图 1.7 所示,用顺序存储结构来存储复数。

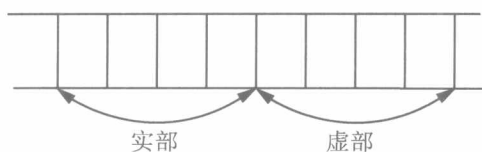


图 1.7 复数的顺序存储

顺序存储结构的主要特点如下。

- (1) 结点中只有自身信息域,没有连接信息域。因此,存储密度大,存储空间利用率高。
- (2) 可以通过计算直接确定任意一个结点的存储地址。
- (3) 插入和删除都将改变结点的位置。

2. 链式存储结构

链式存储结构(Linked Storage Structure)不要求逻辑上相邻的结点在物理位置上也相邻,以在结点中附设指针域存储与该结点相邻的结点的地址来实现结点间的逻辑关系。这样将所有地址域串联在一起的存储方式称为链式存储结构,即链式存储结构不仅存储结点的值,而且还存储结点之间的关系。例如,用链式存储结构表示复数的存储,如图 1.8 所示。

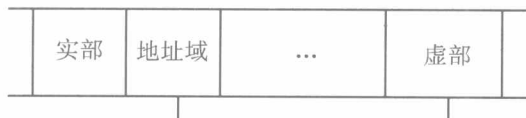


图 1.8 复数的链式存储

链式存储结构的主要特点如下。

- (1) 结点中除自身信息外,还有连接信息的地址域,因此,比顺序存储结构的存储密度小,存储空间利用率较低。
- (2) 逻辑上相邻的结点在物理上不必邻接,可用于线性表、树、图等多种逻辑结构的存储表示。
- (3) 删除和插入操作灵活方便,不必移动结点,只要改变结点中地址域的值即可。

3. 索引存储方法

索引存储是在存储结点信息的同时,建立一个附加的索引表,然后利用索引表中索引号的值来确定结点的存储单元地址。其优点是检索速度快,但是由于增加了附加的索引表,会占用较多的存储空间。

4. 散列存储方法

散列存储方法的基本思想是根据结点的值来确定结点的存储地址。把结点的值作为自变量,通过一个称为散列函数的计算机规则确定出该结点的存储单元地址。