

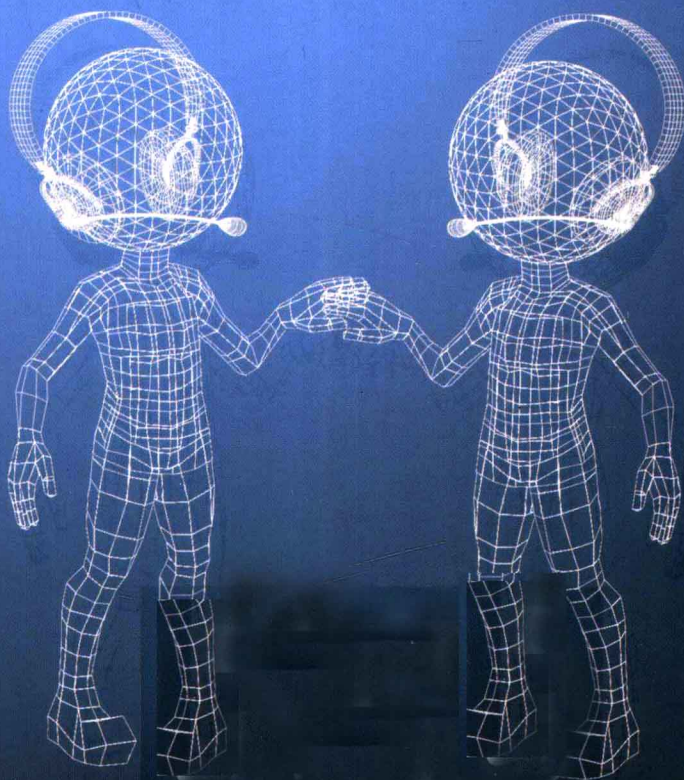


普通高等教育“十一五”国家级规划教材

数 据 结 构

(第三版)

陈 雁 主编



高等教育出版社
HIGHER EDUCATION PRESS

普通高等教育“十一五”国家级规划教材

数 据 结 构

Shuju Jiegou

(第三版)

陈 雁 主编



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书是普通高等教育“十一五”国家级规划教材。

本书主要包括绪论、线性表和数组、栈和队列、树、图、排序、查找、数据结构程序设计等,最后,以 Java 语言为例,介绍了面向对象程序设计的数据结构。书中各章后附有自测练习题。

本书适用于高等职业学校、高等专科学校、成人高等学校、本科院校举办的二级职业技术学院,也可供示范性软件职业技术学院、继续教育学院、民办高等学校、技能型紧缺人才培养使用,还可供本科院校、计算机专业人员和爱好者参考。

图书在版编目(CIP)数据

数据结构/陈雁主编. —3 版. —北京:高等教育出版社, 2010. 12

ISBN 978-7-04-030695-8

I. ①数… II. ①陈… III. ①数据结构-高等学校:技术学校-教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字 (2010) 第 215559 号

策划编辑 冯 英 责任编辑 俞丽莎 封面设计 杨立新
版式设计 马敬茹 责任校对 姜国萍 责任印制 尤 静

出版发行 高等教育出版社

社 址 北京市西城区德外大街 4 号

邮政编码 100120

经 销 蓝色畅想图书发行有限公司

印 刷 人民教育出版社印刷厂

开 本 787×1092 1/16

印 张 15.5

字 数 380 000

购书热线 010-58581118

咨询电话 400-810-0598

网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>

畅想教育 <http://www.widedu.com>

版 次 2001 年 9 月第 1 版
2010 年 12 月第 3 版

印 次 2010 年 12 月第 1 次印刷

定 价 26.60 元(含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 30695-00

第三版前言

本书是普通高等教育“十一五”国家级规划教材。

“数据结构”课程不仅为后续软件类课程提供知识基础,而且在算法设计、程序设计方面提供一定的技能训练,在计算机专业课程中的地位非常重要。本书同时重视知识的掌握和实际动手能力的训练。书中通过一系列任务,引导读者循序渐进地阅读、理解、输入、编辑、编译、调试、运行各种典型的算法和程序,以提高熟练运用知识和技能解决实际问题的能力。

本书保持了前一版本的优点,即针对高职高专的特点,本着理论够用、注重实用的原则,对不常用的数据结构和算法进行了删减,对常用的数据结构进行详细介绍。每章中不仅有大量例题,而且有专门章节进行数据结构程序设计指导,介绍算法书写、上机实习规范、程序调试技巧、实习报告整理等,还专门配有光盘,以多媒体形式形象地演示主要算法和应用实例软件。全书概念表述清楚、简洁,内容由浅入深,强调实践,有利于教学和自学。

为了更好地适应高职高专教学改革的要求,近年来,我们在听取广大读者和专家意见的基础上,针对高技能型人才培养的需求,结合“工学结合”培养模式的课程改革,对全书的内容组织进行了较大幅度的调整和修订,增加了实践性的项目和任务,注重学生的实践动手能力的培养。新版教材的主要特点有以下几点:

① 区别于传统的数据结构教材,使用任务驱动法编写。每章中均提出了多个具体任务,以激发读者的学习兴趣。每个任务都包括任务的提出和描述、任务分析、任务完成示例、要求独立完成任务和过程评价等环节。在教学过程中,以任务为主线,通过对示例任务的详细剖析,使读者掌握任务所要求的技能和知识。通过每章后配套的实训任务,读者可以自行循序渐进地完成任务,以便于巩固和内化已学知识和技能。

② 系统地介绍了各种数据结构的特点、存储结构和有关算法,并采用 C 语言描述算法。主要内容包括线性表、栈、队列、树、图、查找、排序等,每项内容均以任务为主线展开,各示例任务难度逐渐增加,代码规范、可读性强,更便于读者自学。

③ 第 9 章以 Java 语言为实现语言,以“教材管理系统”项目为例,通过完成 14 个任务,对如何利用成熟的数据结构算法实际问题进行了描述。通过简明扼要的任务分析和精简有效的代码实现,体现抽象之美,引导学生更好地学以致用,将数据结构灵活应用于各相关领域。

本教材由陈雁主编,其中第 2~5 章由顾小晶编写,第 6、7 章由陈雁编写,第 1、8、9 章由曾海编写。全书由陈雁统稿。本书附带的光盘由顾成喜完成。

在本书的编写过程中,苏州职业大学李金祥教授在百忙中抽出时间担任主审并提出了许多宝贵的修改建议,高等教育出版社给予了热情帮助和指导,还得到了作者所在单位苏州职业大学领导和同行的支持。在此一并表示衷心的感谢。

由于编者水平有限,书中错误和不妥之处在所难免,敬请广大读者批评指正。

作者联系方式:cy@jssvc.edu.cn。

编者

2010 年 11 月于苏州

目 录

第1章 绪论	1	2.3.2 单链表的基本操作	26
1.1 概述	1	2.3.3 任务 2.3:一元多项式的单链表 表示	30
1.1.1 数据结构的重要性和应用场合	1	2.3.4 任务 2.4:一元多项式相加	32
1.1.2 数据结构的学习方法	3	2.3.5 循环链表和双向链表	35
1.2 数据结构的基本知识	3	2.4 数组	38
1.2.1 数据结构的定义	3	2.4.1 数组的基本概念	38
1.2.2 元素的逻辑关系和物理关系	4	2.4.2 数组的顺序存储结构	38
1.2.3 数据结构应用实例	4	2.4.3 特殊矩阵的压缩存储	39
1.2.4 抽象数据类型	7	2.4.4 任务 2.5:稀疏矩阵的转置	39
1.3 算法和算法描述	9	2.5 实训任务	42
1.3.1 算法的概念	9	2.5.1 实训任务 2.1:超长正整数相加 ...	42
1.3.2 算法的表达	9	2.5.2 实训任务 2.2:求约瑟夫环的出列 顺序	42
1.4 算法的评价	11	自测练习题	43
1.4.1 时间复杂度和空间复杂度的 概念	11	算法设计题	44
1.4.2 时间复杂度的估算规则	12	第3章 栈和队列	46
1.4.3 时间复杂度估算任务	13	3.1 栈	46
1.4.4 常见的不同数量级时间复杂度	15	3.1.1 栈的定义及其运算	46
自测练习题	15	3.1.2 任务 3.1:表达式处理——抽象 描述	46
第2章 线性表和数组	17	3.1.3 栈的顺序存储结构的实现	48
2.1 线性表的逻辑结构	17	3.1.4 栈的链式存储结构的实现	49
2.1.1 线性表的定义	17	3.1.5 任务 3.2:表达式处理——C 语言 实现	51
2.1.2 线性表的基本操作	17	3.1.6 任务 3.3:汉诺塔问题的递归 实现	53
2.1.3 任务 2.1:合并线性表——抽象 描述	18	3.2 队列	55
2.2 线性表的顺序存储结构的实现 ...	18	3.2.1 队列的定义及运算	55
2.2.1 顺序存储结构	18	3.2.2 任务 3.4:消息的加密和解密—— 改进的凯撒密码	56
2.2.2 基本操作的实现	19	3.2.3 队列的顺序存储结构的实现	57
2.2.3 动态分配的顺序存储结构介绍	21	3.2.4 任务 3.5:消息的加密和解密——	
2.2.4 任务 2.2:合并线性表——C 语言 实现	22		
2.3 线性表的链式存储结构的实现 ...	25		
2.3.1 单链表	25		

II 目 录

C 语言实现	59	4.6.1 赫夫曼树的基本概念	101
3.2.5 队列的链式存储结构的实现	61	4.6.2 任务 4.5:构造赫夫曼树	101
*3.3 拓展任务	63	4.6.3 任务 4.6:赫夫曼编码	104
3.3.1 任务 3.6:算术表达式的求值	63	4.7 实训任务	106
3.3.2 任务 3.7:模拟停车场管理	68	4.7.1 实训任务 4.1:二叉树遍历及 应用	106
3.4 实训任务	73	4.7.2 实训任务 4.2:设计并输出赫夫曼 编码	107
3.4.1 实训任务 3.1:数制转换问题	73	自测练习题	107
3.4.2 实训任务 3.2:手机短信息功能的 模拟	73	算法设计题	108
自测练习题	74	第 5 章 图	109
算法设计题	75	5.1 图的基本概念	109
第 4 章 树和二叉树	76	5.1.1 图的定义	109
4.1 树	76	5.1.2 图的基本术语	110
4.1.1 树的定义	76	5.2 图的存储结构	111
4.1.2 树的基本术语	77	5.2.1 邻接矩阵表示法	111
4.2 二叉树	78	5.2.2 任务 5.1:建立无向图的邻接 矩阵	112
4.2.1 二叉树的定义	78	5.2.3 邻接表	115
4.2.2 二叉树的重要性质	78	5.2.4 任务 5.2:建立有向图的邻 接表	116
4.2.3 二叉树的存储结构	79	5.3 图的遍历	118
4.2.4 任务 4.1:建立二叉树的二叉 链表	81	5.3.1 连通图的深度优先搜索遍历	119
4.3 遍历二叉树	82	5.3.2 连通图的广度优先搜索遍历	120
4.3.1 先根遍历二叉树	83	5.3.3 任务 5.3:求无向图的连通 分量	121
4.3.2 任务 4.2:求二叉树的叶子结 点数	85	5.4 图的最小生成树	123
4.3.3 中根遍历二叉树	88	5.4.1 生成树的概念	123
4.3.4 任务 4.3:打印二叉树	89	5.4.2 网络的最小生成树	124
4.3.5 后根遍历二叉树	91	5.4.3 任务 5.4:局域网布线造价 问题	128
4.3.6 任务 4.4:销毁二叉树	92	5.5 最短路径	129
4.4 线索二叉树	94	5.5.1 从某源点到其余顶点之间的最短 路径	130
4.4.1 线索二叉树的基本概念	94	5.5.2 任务 5.5:输出最短路径	132
4.4.2 中根线索二叉树	95	5.5.3 求有向网中每一对顶点间的最短 路径	134
4.5 树和森林	97	5.6 有向无环图及其应用	136
4.5.1 树的存储结构	97	5.6.1 拓扑排序	136
4.5.2 树与二叉树之间的转换	97		
4.5.3 森林与二叉树的转换	98		
4.5.4 树和森林的遍历	100		
4.6 赫夫曼树及其应用	101		

*5.6.2 关键路径	139	7.3.3 平衡二叉树与动态平衡技术	185
5.7 实训任务	145	7.3.4 B-树用于外部查找	187
5.7.1 实训任务 5.1:游览线路设计	145	7.4 哈希表及其查找	192
5.7.2 实训任务 5.2:交通咨询	146	7.4.1 哈希函数与哈希表	192
自测练习题	147	7.4.2 构造哈希函数的常用方法	193
算法设计题	148	7.4.3 任务 7.2:设计哈希函数	195
第 6 章 排序	149	7.4.4 解决冲突的主要方法	196
6.1 排序的基本概念	149	7.4.5 哈希查找效率的分析	200
6.2 插入排序	150	7.4.6 任务 7.3:设计哈希表	201
6.2.1 直接插入排序	150	7.5 实训任务	203
6.2.2 折半插入排序	151	7.5.1 实训任务 7.1:二叉排序树的 应用	203
6.2.3 希尔排序	152	7.5.2 实训任务 7.2:哈希表的应用	204
6.3 交换排序	153	自测练习题	204
6.3.1 冒泡排序	153	第 8 章 数据结构程序设计及其他	205
6.3.2 任务 6.1:字符串排序	154	8.1 数据结构程序设计	205
6.3.3 快速排序	156	8.1.1 算法书写规范	205
6.4 选择排序	157	8.1.2 实习步骤规范	207
6.4.1 简单选择排序	157	8.2 实习报告范例	208
6.4.2 堆排序	159	8.3 其他数据结构和加密算法	211
6.4.3 任务 6.2:实现堆排序	162	自测练习题	214
6.5 归并排序	164	第 9 章 Java 中已实现数据结构的 使用	215
6.6 基数排序	166	9.1 Java 中数据结构概述	215
6.7 内部排序总结	169	9.1.1 Java 中数据结构相关类体系	215
6.8 多路归并用于外排序的简介	170	9.1.2 迭代器	217
6.9 实训任务	172	9.2 链表的实现	219
6.9.1 实训任务 6.1:成绩表排序	172	9.2.1 任务 9.1:建立一个链表 LinkedList	219
6.9.2 实训任务 6.2:操作系统进程 优先级排序	173	9.2.2 任务 9.2:遍历整个链表	219
自测练习题	173	9.2.3 任务 9.3:对链表的常用操作的 Java 实现	220
第 7 章 查找	174	9.2.4 程序改进	221
7.1 查找的基本概念	174	9.3 可变长数组的实现	221
7.2 静态查找表	175	9.3.1 任务 9.4:教材管理软件的数据 存储实现	221
7.2.1 顺序查找	175	9.3.2 任务 9.5:全部教材信息的 显示	222
7.2.2 折半查找	177		
7.2.3 索引顺序的查找	179		
7.3 动态查找表	180		
7.3.1 二叉排序查找树	180		
7.3.2 任务 7.1:建立二叉排序树	183		

IV 目 录

9.3.3 任务 9.6:教材订购方法的参数 传递	223	9.6.2 任务 9.11:将教材信息按教材名 排序	232
9.4 任务 9.7:将教材链表逆序 输出	224	9.6.3 任务 9.12:将教材信息按教材价 格排序	234
9.5 哈希表的实现	225	9.6.4 任务 9.13:将教材信息对按关键 字排序	234
9.5.1 任务 9.8:根据书号查询教 材名	225	9.6.5 任务 9.14:显示满足条件的部分 教材	236
9.5.2 任务 9.9:程序初始值存储和 装入	228	自测练习题	237
9.6 自动排序的实现	231	参考文献	238
9.6.1 任务 9.10:教材名自动排序	231		

1.1 概述

1.1.1 数据结构的重要性和应用场合

“数据结构”课程是计算机相关专业学生必修的重要专业基础课,同时也是一门具有一定学习难度的课程。数据结构主要研究数据的各种逻辑结构和存储结构以及对数据的各种操作。简言之,数据结构有数据的逻辑结构、数据的物理存储结构和对数据的操作(或算法)等三个方面的内容。通常,算法的设计取决于数据的逻辑结构,算法的实现取决于数据的物理存储结构,如图 1-1-1 所示。

第一门结构化高级语言的发明者,瑞士科学家沃斯教授(N. Wirth)说过,“算法+数据结构=程序”。作为软件开发类课程的最重要先修课程之一,数据结构的重要作用体现在多个方面。

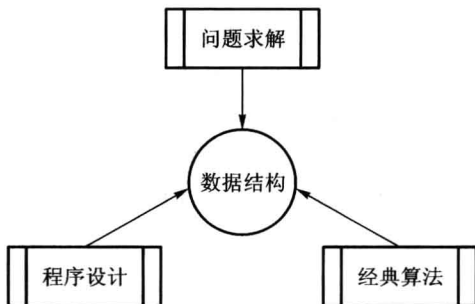


图 1-1-1 数据结构的组成

1. 学好数据结构是参加各类计算机考试的必备要求

在不同层次、不同水平的各类计算机考试中,数据结构部分通常是必考的。

(1) 计算机软件考试

根据人事部、信息产业部文件(国人部发[2003]39号),对计算机专业技术人员职称评定和水平认定均由计算机技术与软件专业技术资格(水平)考试(简称计算机软件考试)认定。其中,在2008年软件设计师考试试题中,纯数据结构的知识点部分就占9分,与数据结构相关的程序设计更是占了考试的一半份量。

(2) 计算机相关专业“专升本”考试

大部分本科院校要求报考计算机相关专业“专升本”的学生先修“数据结构”课程。例如,2008年江苏某招收“专转本”学生的高校在招生章程中特别注明:

学习成绩优良,基础课与专业课的学习成绩达到良好以上,而且必须已修“C/C++程序设计”、“数据结构”两门专业基础课。

(3) 硕士研究生和博士研究生入学考试

大多数院校招收计算机专业硕士的入学考试均需要考数据结构。以2008年中国科学技术大学计算机专业硕士的入学考试科目为例,科目823软件技术基础、科目861计算机技术基础、科目863计算机软件基础和科目864计算机系统结构均指定必考数据结构。

2. 学好数据结构是培养编程能力的必由之路

计算机相关专业学生最重要的专业核心能力之一是编程能力。良好的编程能力意味着学生需要有严密的逻辑思维能力和良好的程序设计习惯。通常,数据结构课程还同时承担了程序设计基础能力培养和算法能力培养等任务。在实际工作中,程序设计语言只是工具,数据结构和算法则是灵魂。编程者使用不同的语言,以不同的形式实现数据结构和算法。

以2009年某企业校园招聘中的“搜索研发工程师”岗位技术要求为例,部分要求是:

- ① 熟悉 C/C++ 语言编程。
- ② 熟悉 Perl / Python / Linux Shell / Java / Java EE 编程技术者优先。

该岗位要求掌握多种计算机语言,但几乎没有一个学校能把上面列举的所有语言列入教学计划。因此,学好多种计算机程序设计语言的诀窍之一就是学好“数据结构”课程,将语言作为工具,让良好数据结构基础提供清晰的思路和方法。

3. 学好数据结构能为计算机各门专业课程学习提供良好的基础

通常,“数据结构”课程开设较早,一般在入学的第二学期开设。开设早的原因是,离开了“数据结构”课程,许多后续课程的学习将遇到非常大的障碍。除了程序设计类课程外,下面以几门核心课程为例进行说明。

(1) 计算机网络

计算机网络中大量使用数据结构的知识。作为网络互连的最重要设备的路由器和交换机,其核心软件中使用的数据结构分别是图和树。其中,路由器的作用之一是在网络的源和目的之间找出最短路径,这正是数据结构图论中的基本算法。为了避免环路的出现,智能交换机使用了树理论部分的最小生成树算法。路由器和交换机中还大量使用了堆栈和队列。甚至在网络界的权威厂商思科(Cisco)公司的网络工程师认证考试中都涉及以上数据结构和算法。

(2) 操作系统

目前流行的操作系统,如 Windows、Linux 均是大型应用软件,其中数据结构的影响更是无处不在。一般操作系统的目录结构使用的是一种特殊的树,CPU 进程管理中用到的数据结构是优先级队列,设备管理中使用了双向循环链表,作业管理中用到了多优先级轮转队列。在进行手机软件研发时,重要的工作就是了解手机操作系统中的数据结构并在此基础上编程。显然,如果没有学好数据结构,那么这方面工作根本无法开展。

(3) 数据库相关课程(包括数据库概论、SQL-Server、Oracle 等)

数据库技术在计算机业界已经是一个独立的分支,数据库程序员和数据库管理员都是独立的岗位。对数据库程序员而言,进行数据库操作时需要把数据读入内存中,进行排序、查找、删除和插入。例如,某程序登录需要验证用户名和密码。用户名密码存在数据库中,程序运行时数据被读入内存以加速检索。在内存中,一组用户名密码是以何种数据结构存储?如何进行快速排序和检索?如何进行多用户情况下的增加、删除、修改和查找?这些均需要数据结构的知识支撑。对于数据库管理员而言,提高数据库性能的重要方法之一是建立索引,而索引的数据结构就是经过改良的 B 树。

1.1.2 数据结构的学习方法

数据结构是计算机科学这幢大楼的重要基石之一,学好“数据结构”课程也是锻炼逻辑思维和编程能力的重要途径。在具体介绍课程内容前,有一个重要问题要解答,如何学好数据结构?这里给出几个建议,如图 1-1-2 所示。

(1) 保持耐心,多阅读程序,多做实验

数据结构是一门实践性非常强的课程,没有实践是学不好的。许多人羡慕某些“高手”阅读程序速度很快,写程序速度也迅速。实际上,除了极少数天才之外,“高手”们都是通过大量阅读和编程逐渐锻炼出来的。由于数据结构是在练习“内功”,因此永远不要指望能在几天内速成。也许刚翻开数据结构方面的书感觉非常复杂,但是只要认认真真阅读程序,认真完成课后习题和实训,随着时间的积累,思路和编程能力都会明显改善。切忌不读程序,不写程序。要有思想准备,光靠上课认真听讲还是远远不够的。

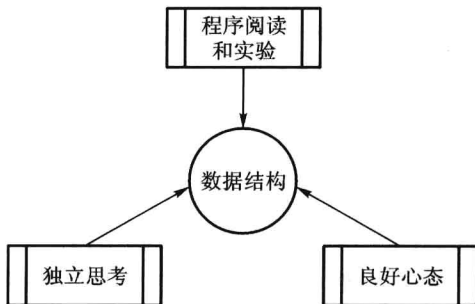


图 1-1-2 学好数据结构的建议

(2) 保持独立思考

在软件开发中需要团队精神,有困难可以互助。但是,数据结构的学习需要领悟。有些人习惯问别人算法的关键思路,然后参考别人写好的程序完成作业。这虽然不算抄袭,但对自己的独立思考能力的培养相当不利。没有独立思考的过程,要自己动手写程序,是完全不可能的事情,这完全不是危言耸听。

(3) 保持良好心态,认真练好基本功

数据结构的上机练习可以使用 Turbo C 2.0 进行。有人提出,现在已经是 21 世纪了,微软阵营使用 C#和 Visual Studio 2010 等进行应用程序开发,Java 阵营往往使用 Java 语言加 Eclipse 等集成开发环境进行应用程序开发,而数据结构还在使用古老的 C 语言上课。实际上,数据结构是锻炼思维能力和编程能力的课程,更注重内涵,因此使用何种语言并不重要。C 语言有着优良的通用性,向其他语言转型非常方便。而且,C 语言小巧玲珑,运行迅速,完全能够满足“数据结构”课程的教学需要。

1.2 数据结构的基本知识

1.2.1 数据结构的定义

什么是数据结构?数据结构是计算机科学中广泛使用的一个术语,它反映一个数据的内部构成,即一个数据由哪些成分数据构成,以什么方式构成,呈什么结构。

严格地说,数据结构是相互之间存在一种或多种特定关系的数据元素的集合。数据结构是一个二元组,记为:

$$\text{data_structure} = (D, S)。$$

其中, D 为数据元素的集合, S 是 D 上关系的集合。

数据元素之间有多种关系,其中常见的关系有以下几类,如图1-2-1所示。

① 集合关系。各数据元素同属于一个集合。例如,一个班级里的同学同属于这个班级,所有同学同属于一个集合。

② 线性结构。线性结构中的数据元素间存在顺序关系。例如,在车站排队买票的人员之间有先后次序,他们形成一个线性结构。

③ 树状结构。树状结构中的数据元素间的关系是一对多的关系。例如,政府机构的结构中,部下设许多厅,厅下设许多局,各部门从上至下形成树状结构。

④ 图(网)状结构。图(网)状结构中的数据元素间的关系是多对多的关系。例如,因特网中每个网络结点都可以和任意一个其他结点连接,各结点之间形成网状结构。

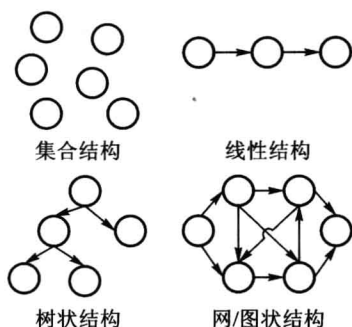


图 1-2-1 常见的关系

因此,数据结构是信息的一种组织方式,其目的是提高算法的效率。它通常与一组算法的集合相对应,通过这组算法集合可以对数据结构中的数据进行某种操作。

1.2.2 元素的逻辑关系和物理关系

数据元素在计算机中的存储方式和数据元素间的逻辑关系并没有必然相同的关系。

例如,储户到银行办理业务时,银行都有一台称为“叫号机”的设备。储户进入银行后首先取得一个顺序号。银行办理业务时,按此顺序号依次办理。这称为元素间的逻辑关系。储户在银行里免去了排队之苦,他们可以在银行大厅的任意位置就座,顺序并不重要。这是元素间的物理关系。如图1-2-2所示。

在本例中,数据元素的逻辑关系和物理关系是不一样的。数据元素之间存在的关系(逻辑关系)称为数据的逻辑结构。数据结构在计算机中的表示(映像)称为数据的物理结构。一种逻辑结构可映像成不同的存储结构。

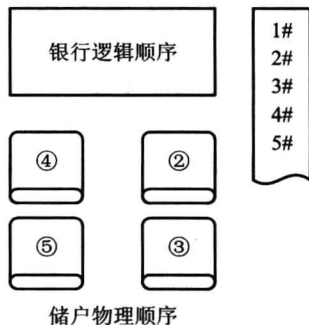


图 1-2-2 元素间的物理关系

1.2.3 数据结构应用实例

在程序设计课程的学习中,用计算机解题的基本方法是首先分析问题,确定数据模型;然后设计相应的算法;接着编写程序;最后反复调试程序直至得到正确的结果。有些问题的数据模型可以用具体的代数方程、矩阵等来表示。

但是,大多数日常业务无法用简单的数学模型所表示,如移动公司的业务办理系统、医院的医疗信息系统、民航和铁路部门的售票系统等。这类业务称为非数值业务,也称为管理信息系统业务,其开发流程一般遵守软件工程方法的开发流程,如图1-2-3所示。

开发软件的重要步骤之一是总体设计。在此阶段必须确定数据的存储方式,之后所有的程序设计都依赖于这一存储方式。

任务 1.1:某小型培训机构教材管理的数据结构分析

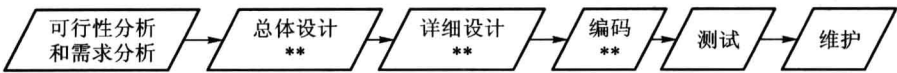


图 1-2-3 软件工程方法的开发流程

任务分析：

某小型培训机构有一个维护教材进货信息的程序。应该有一个数据结构负责存储目前已有的教材信息。这个数据结构应该有多行，每行数据由一条书号、当前库存、单价、进货优先级的数据记录描述。这里书号、当前库存数、单价、进货优先级称为数据项。

数据结构：

书号	当前库存数	单价	进货优先级
----	-------	----	-------

只定义数据结构是没有意义的，关键是要同时定义对数据的操作。从教材管理的业务来看，主要操作可以包括修改某些数据项以及改变进货优先级。

对数据结构的操作：

- ① 修改当前库存数。
- ② 修改教材单价。
- ③ 当库存低于一定的值时，提高进货优先级。库存高于一定值时，降低进货优先级。

从 C 语言角度看，教材进货维护程序描述可以分为两部分。

- ①定义数据结构。每本书的订单可以用 C 语言中的结构体来表示。

```
struct BookInfo{
    int bookNumber;
    int stock;
    float price;
    int priority;
}
```

- ② 定义对此数据结构的操作。

```
/* 修改当前库存数 */
Void changeStock( struct BookInfo oneBook ,int stk ) {
    oneBook. stock = stk;
}
/* 修改教材单价 */
Void changePrice( struct BookInfo oneBook ,int price ) {
    oneBook. price = price;
}
/* 改变教材进货优先级 */
Void changePriority( struct BookInfo oneBook ,int priority ) {
    oneBook. priority = priority;
}
```

最终,某小型培训机构的教材管理程序的描述如图 1-2-4 所示。

将数据结构定义和数据操作结合,就能完成一个完整的程序,正如本章开始写到的 N. Wirth 教授的一句话“算法+数据结构=程序”。

任务 1.2:图形处理程序数据结构分析

图形处理是计算机应用的一大领域,复杂的图形处理都是由基本图形处理组合后完成的。这里以一个圆为例说明如何编写程序对圆进行操作。

任务分析:

在计算机内部表示一个圆,从显示来看需要两个数据,一个是圆心,一个是半径。在计算机中,圆的圆心由 (x,y) 一对坐标构成。对圆的操作包括求面积和求周长的操作。

数据结构:

圆心	半径
----	----

对数据结构的操作:

- ① 求面积。
- ② 求周长。

使用 C 语言实现时,同样可以用结构体表示一个圆。

① 定义数据结构。

```
struct Circle
{
    int x;
    int y;
    int radius;
}
```

② 定义对此数据结构的操作。

要求面积可以定义一个函数:

```
/* 求圆的面积 */
void float area( struct Circle cir)
{
    return 3.14159 * cir.radius * cir.radius;
}

/* 求圆的周长 */
void float circumference( struct Circle cir)
{
    return 2 * 3.14159 * cir.radius;
}
```

数据定义	BookManangerment
	#bookNumber #stock #price #priority
数据操作	+changeStock() +changePrice() +changePriority()

图 1-2-4 教材管理程序的描述

最终,图形处理程序可以描述如图 1-2-5 所示。

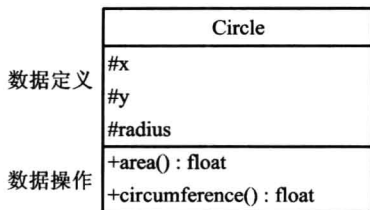


图 1-2-5 图形处理程序

任务 1.3:教学计划编排数据结构分析

任务 1.1 和任务 1.2 难度不高,容易实现,数据结构也非常简单,是最简单的集合。实际问题要比它们复杂许多,看下面的任务。

任务描述:

教学计划编排问题。一个教学计划包含许多课程,在教学计划包含的许多课程之间,有些必须按规定的先后次序进行,有些则没有次序要求,即有些课程之间有先修和后续的关系,有些课程可以任意安排次序,如图 1-2-6 所示。

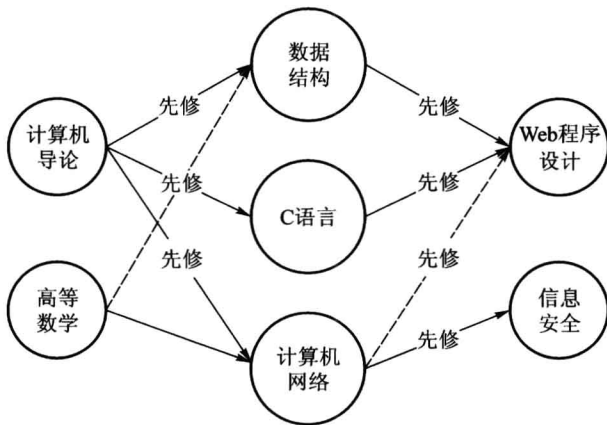


图 1-2-6 教学计划

任务分析:

这种各个课程之间的次序关系可用一种称作图的数据结构来表示。图的数据结构不是 C 语言中的结构体能简单表示的。图中任意一个结点可以有多个邻接结点。

对数据结构的操作包括遍历整个图,查询某门课程的先修课程是什么,查询某门课程的后修课程是什么。所有操作都将在第 5 章中详细叙述。

1.2.4 抽象数据类型

在 1.2.3 节中,对任务的分析包括了对数据结构的定义和在此数据结构上操作的定义。这

正是抽象数据类型要表达的核心内容。

抽象数据类型 (Abstract Data Type, ADT) 是指基于一类逻辑关系的数据类型以及定义在这个类型上的一组操作。在软件设计中,抽象数据类型通常包含定义、表示和实现三部分。

抽象数据类型的定义可以看做是描述问题的模型,它独立于具体实现。从任务 1.1 和任务 1.2 中可以看出,对数据结构和操作的定义是完全独立于 C 语言实现的。对一个问题的描述,可以用 C 语言实现,也可以用 Java 语言或者其他语言实现。

ADT 的主要作用之一就是问题抽象。

在面向对象的分析和设计中,数据和操作是密不可分的,它们形成一个“类”(Class)。再看一遍图 1-2-7,它实际上是统一建模语言 (Unified Model Language, UML) 中的类图,就是 ADT 的一种表现形式。

其中,数据定义称为属性,而数据操作部分称为方法。类图可以被所有面向对象 (Object Oriented Programming, OOP) 的语言实现为具体的代码,而后产生属于某个“类”的具体变量称之为“对象”(Object),最后通过方法调用可以运行程序。

本书使用传统的 C 语言作为数据结构的描述工具。数据和对数据的操作是分离的,因此类图不能直接被转换成可执行代码。此时可以使用另一种文字描述 ADT 的方式来表示问题。

例如,一个圆的 ADT 可以表示如下。

```
ADT Circle {  
    数据元素:圆心坐标(x,y),半径 radius  
    数据逻辑结构:无  
    基本操作:  
        area():求圆的面积  
        circumference():求圆的周长  
}
```

特别注意,ADT 的描述大多数为文字以及不能执行的代码,称为“伪码”。其主要目的是利于理解,不要求严格的语法,在不同的情况下可以灵活表示,只要保证数据元素、数据逻辑结构和基本操作均有即可。

在面向对象的语言中,ADT 可以直接具体转换为类的定义。圆的类在 Java 中可以表示为:

```
class Circle {  
    int x,y;  
    int radius;  
    public float area() { return 3.14159 * radius * radius; }  
    public float circumference() { return 2 * 3.14159 * radius; }  
}
```

和 C 语言中的伪码 ADT 不同,Java 中用类表示的 ADT 可以直接被 Java 编译系统接受的可执行代码。本书主要使用伪码类 ADT 描述问题。

抽象数据类型设计是软件设计的基础。在数据结构课程中讨论的各种基本数据结构,如线

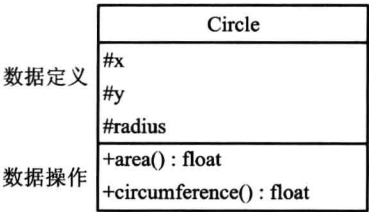


图 1-2-7 面向对象的分析和设计中的数据和操作

性表、栈、队列等,都可以作为设计更复杂软件的抽象数据类型。当应用程序中所要用到的各种基本数据结构都已有设计好的可重复使用的抽象数据类型时,以后的程序设计和程序维护将大大简化。

1.3 算法和算法描述

“算法+数据结构=程序”,这是本章中第三次出现的公式。对问题描述的一大重点就是如何对数据结构进行操作;如何使操作效率更高,速度更快。这些问题依靠算法来解决。

1.3.1 算法的概念

算法被定义为一个运算序列。这个运算序列中的所有运算定义在一类特定的数据模型上,并以解决一类特定问题为目标。

简单而言,算法就是对特定的数据结构进行一系列操作,其目的是解决一个具体问题。

这个运算序列应该具备下列4个特征:

- ① 有限性,即序列的项数有限,且每一运算项都可在有限的时间内完成。
- ② 确定性,即序列的每一项运算都有明确的定义,无二义性。
- ③ 可以没有输入运算项,但一定要有输出运算项。
- ④ 可行性,即对于任意给定的合法的输入都能得到相应的正确的输出。

算法描述了问题,最终要由计算机程序来实现。这里关心的是算法的描述和程序表达。

1.3.2 算法的表达

1. 算法的程序表达

算法的程序表达,归根到底是算法要素的程序表达,因为一旦算法的每一项要素都用程序清楚地表达,整个算法的程序表达也就不成问题。

作为运算序列的算法有以下3个要素。

(1) 作为运算序列中各种运算的运算对象和运算结果的数据

由于算法变化万千,其中的运算所作用的目标数据和得到的结果数据种类繁多。其中最简单、最基本的有布尔值数据、字符数据、整数和实型数据等,稍复杂的有向量、矩阵、记录等数据,更复杂的有集合、树和图等。C语言中支持大部分数据类型。

(2) 运算序列中的各种运算

同样由于算法变化万千,运算的种类也多种多样。其中最基本、最初等的有赋值运算、算术运算、逻辑运算和关系运算等,稍复杂的有算术表达式和逻辑表达式等,更复杂的有函数值计算、向量运算、矩阵运算、集合运算以及表、栈、队列、树和图上的运算等。

(3) 运算序列中的控制转移

在串行控制序列中,控制转移只有顺序、分支、循环、递归和无条件转移等几种。它们相对简单,C语言均支持。

从算法的3个特点可以看出,C语言作为一种高级语言,有能力用程序表达算法。但是,在设计算法时,一些过于具体的细节没有必要过早考虑,细节问题可以放到具体程序设计时去解