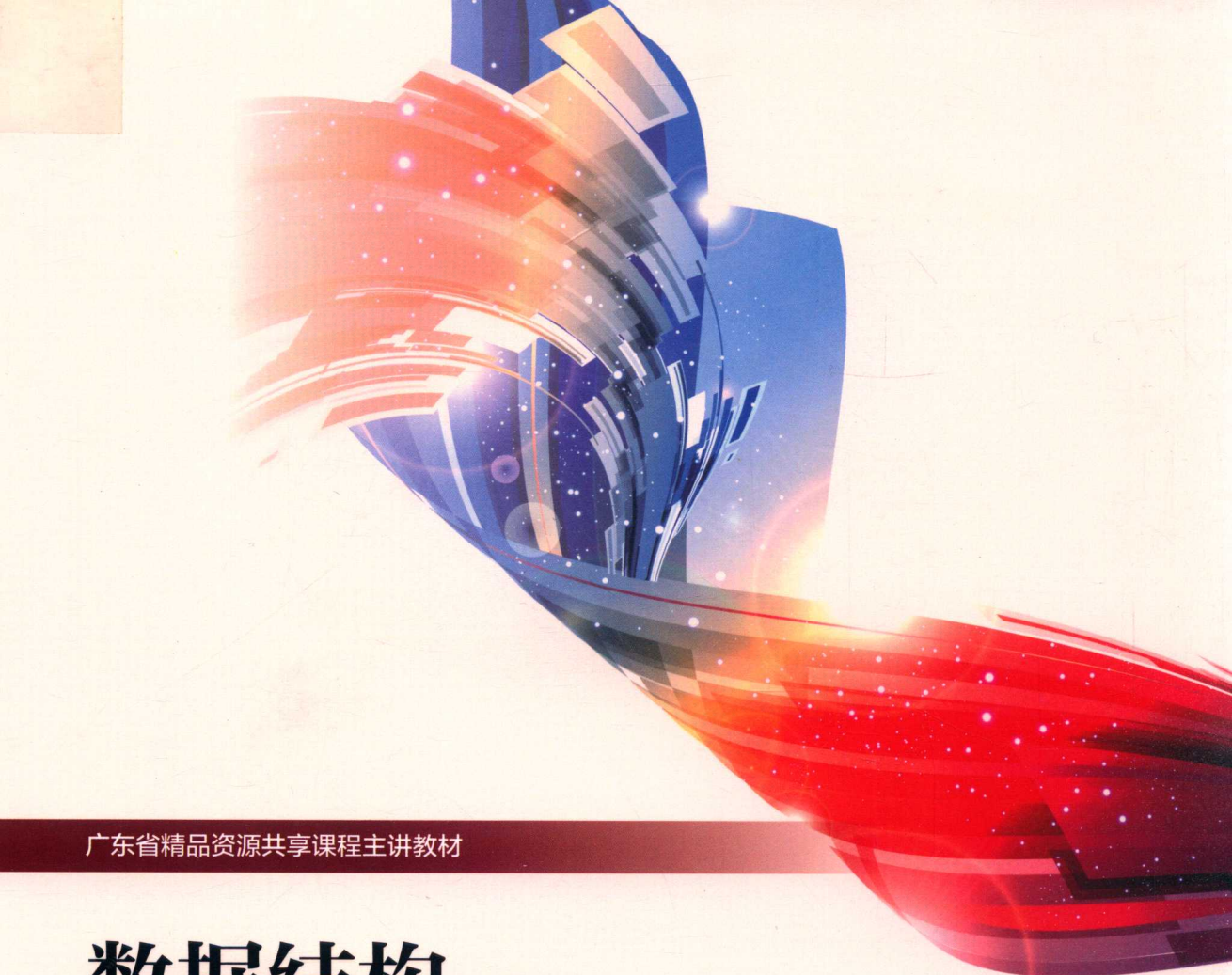




广东省精品资源共享课程主讲教材

数据结构

Data Structures

主 编 吴伟民
副主编 李小妹
编 者 刘添添 黄剑锋 苏 庆
林志毅 李 杨

高等教育出版社

广东省精品资源共享课程主讲教材

数据结构

主 编 吴伟民
副主编 李小妹
编 者 刘添添 黄剑锋 苏 庆
林志毅 李 杨

高等教育出版社·北京

内容提要

本书主要讲述数据结构、算法及算法分析的理论。全书共分8章，在绪论部分介绍数据结构、算法的相关概念和算法分析方法等，其后各章分别讨论栈、队列、线性表、哈希表、二叉树、树、森林和图等数据结构的定义、表示和实现。将查找和排序融入相应的数据结构的讨论中，并在二叉树前介绍递归内容。在多数章节中加入应用实例，介绍运用数据结构和算法进行程序设计和解决实际问题的方法，以增强读者对基本知识的理解与掌握，有利于分析问题能力和程序设计能力的提高。全书采用C语言作为数据结构和算法的描述语言。

本书可作为高等学校计算机类和信息类相关专业的本科或专科教材，也可作为相关教师、研究生和工程技术人员的参考书。

图书在版编目(CIP)数据

数据结构 / 吴伟民主编; 刘添添等编. -- 北京: 高等教育出版社, 2017. 7
ISBN 978-7-04-047946-1

I. ①数… II. ①吴… ②刘… III. ①数据结构-高等学校-教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字(2017)第 150807 号

数据结构

Shuju Jiegou

策划编辑 倪文慧	责任编辑 倪文慧	封面设计 张志	版式设计 杜微言
插图绘制 杜晓丹	责任校对 胡美萍	责任印制 刘思涵	

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
印 刷 山东鸿君杰文化发展有限公司

开 本 787mm×1092mm 1/16
印 张 15
字 数 300 千字
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.hepmall.com.cn>
<http://www.hepmall.com>
<http://www.hepmall.cn>
版 次 2017 年 7 月第 1 版
印 次 2017 年 7 月第 1 次印刷
定 价 30.00 元

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 47946-00



吴伟民

国务院政府特殊津贴专家，广东省计算机学会常务理事，广东工业大学计算机学院教授。与清华大学严蔚敏教授合著数据结构系列教材，曾先后荣获国家级普通高校优秀教材特等奖和国家级科技进步三等奖。主要研究领域：计算机系统软件与系统结构，计算机系统逆向和介入工程技术及工具，数据结构与算法，可视计算及程序可视化运行、调试与测评，程序设计语言和编译系统，虚拟机和虚拟化技术，智能系统与电器，嵌入式系统等。其他主要获奖：电子工业部科技成果二等奖、霍英东青年教师三等奖、曾宪梓高师教师三等奖、广东省教学成果二等奖、广东省计算机学会特等奖等。

数字课程资源使用说明

与本书配套的数字课程资源发布在高等教育出版社易课程网站,请登录网站后开始课程学习。

一、注册/登录

访问 <http://abook.hep.com.cn/1858120>,单击“注册”按钮,在注册页面输入用户名、密码及常用的邮箱进行注册。已注册的用户直接输入用户名和密码登录即可进入“我的课程”页面。

二、课程绑定

单击“我的课程”页面右上方“绑定课程”链接,正确输入教材封底防伪标签上的 20 位密码,单击“确定”按钮完成课程绑定。

三、访问课程

在“正在学习”列表中选择已绑定的课程,单击“进入课程”按钮即可浏览或下载与本书配套的课程资源。刚绑定的课程请在“申请学习”列表中选择相应课程并单击“进入课程”按钮。

四、与本书配套的易课程数字课程资源包括电子教案、知识点讲解视频等素材,以便读者学习使用。

账号自登录之日起一年内有效,过期作废。

如有账号问题,请发邮件至 abook@hep.com.cn。

“数据结构”是一门研究用计算机进行信息数据表示和处理的课程。一方面,信息的表示和组织直接关系到处理信息的程序的效率;另一方面,信息的处理方法往往是根据信息的组织关系来设计的。课程致力于训练计算机软件开发人员运用抽象思维表示和处理数据,进而以计算机程序的形式运行并获得结果。

基于多年的数据结构教学经验,编者编写了这部教材,其具有以下特色。

1. 内容体系重构优化

遵循循序渐进和由易到难的教学原则,重构、优化了教材内容体系。

对于线性数据结构,一改以往先介绍线性表再介绍栈和队列的做法,先介绍接口简单的栈和队列,再拓展到更一般的线性表。

紧接线性结构之后,设置“排序基础”一章,介绍排序概念和基本算法,涉及递归的排序算法则分散于后续相关章节。类似地,对于查找结构与算法,单独设置“哈希表”一章,二叉排序树和平衡二叉树以及 B 树则安排在二叉树和树的章节中。经典的排序和查找算法都是与具体的数据结构相结合的,本质上是相应数据结构的具体应用,如此编排体现了数据结构与算法的紧密结合。

将教学难点“递归”单独设章,介绍递归函数的调用原理,通过折半查找、快速排序和归并排序 3 个经典算法的讨论,学习递归与分治的算法设计思想。作为从线性结构到非线性结构的过渡,广义表为后续二叉树和树的学习建立了递归数据结构的基本概念。

2. 接口定义精简实用

现有教材大多对接口的定义往往是过分追求操作集的完备,而忽略了实用性,不够合理。基于对实际应用的分析和归纳,本书对各种数据结构的接口定义进行了全面的精简与优化,总体降低了教学难度,减少了学时。

比如,现有教材大多在顺序表和单链表的接口中都含有插入和删除第 i 个元素的操作,其中顺序表需要移动大量数据元素,单链表需要找到第 $i-1$ 个位置的元素。这些操作效率不高且实际应用很少,本书已将其去掉。又如,一般二叉树的接口操作都相当多,本书将其缩减到仅 7 个。

3. 算法代码可上机运行

本书采用扩展了引用形参的 C 语言描述数据结构的存储结构和接口的实现,所有代码无须转换就可以上机编译、运行和实际应用。

4. 教学资源丰富多彩

课程组为教材开发了数据结构学习网站。精心制作了 PPT, 还录制了重要知识点的微视频, 可在网站在线观看, 也可以扫描教材中的二维码在手机等移动终端上观摩。视频对数据结构课程的知识网络进行了清晰梳理, 补充了部分扩展知识, 支持学生拓展学习宽度和深度。读者还可以在网站上注册, 在线做题并得到实时测评。数据结构的课件与相关资料可在网站下载。

本书共 8 章。第 1 章为绪论, 主要介绍数据结构的基本概念和算法复杂度评价方法。第 2 章介绍线性数据结构, 介绍 3 种典型的线性数据结构: 栈、队列和线性表。第 3 章介绍线性数据结构的排序基础, 包括排序的定义以及基本的排序算法。第 4 章介绍哈希表, 这种结构在需要实现快速查找的应用场合具有相当高的使用价值。第 5 章介绍递归结构与算法的基础及广义表, 递归思维在后续章节的学习中相当重要, 而递归数据结构广义表是线性结构与非线性结构之间的过渡。第 6 章和第 7 章介绍的二叉树、树和森林是层次数据结构, 这两章主要介绍二叉树和树的概念、性质、遍历与应用, 与堆排序相关的完全二叉树——堆, 以及与查找相关的二叉查找树、二叉平衡树和 B 树。第 8 章介绍网状数据结构——图。

本书建议授课学时为 32~64 学时, 并安排上机实习 16~32 学时。

在“互联网+”时代, 教学也是教师与学生、学生与学生之间通过网络互联互动的过程。本书作为讲义已经试用了 3 年, 每年均结合实际情况进行修订。3 年来, 课程组承担省级慕课教学试验课题, 授课 32 学时, 实验室研讨 16 学时, 除要求学生在线完成各章节的基础知识测试之外, 还要在 Anyview C 程序可视化调试与实时测评平台上在线完成 40~80 道算法设计题。本书也可以在翻转课堂的新型教学模式下使用。

浙江大学陈越教授在百忙之中对本书进行了审阅, 并提出了宝贵的建议和意见, 特此表示衷心的感谢!

本书尚有不足和需完善之处, 欢迎广大读者与我们交流, 主编吴伟民教授的 Email 地址为 wuwm@gdut.edu.cn。

广东工业大学计算机学院数据结构教材编写组

2017 年 7 月

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》,其行为人将承担相应的民事责任和行政责任;构成犯罪的,将被依法追究刑事责任。为了维护市场秩序,保护读者的合法权益,避免读者误用盗版书造成不良后果,我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为,希望及时举报,本社将奖励举报有功人员。

反盗版举报电话 (010)58581999 58582371 58582488

反盗版举报传真 (010)82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号 高等教育出版社法律事务与版权管理部

邮政编码 100120

防伪查询说明

用户购书后刮开封底防伪涂层,利用手机微信等软件扫描二维码,会跳转至防伪查询网页,获得所购图书详细信息。也可将防伪二维码下的20位密码按从左到右、从上到下的顺序发送短信至106695881280,免费查询所购图书真伪。

反盗版短信举报

编辑短信“JB,图书名称,出版社,购买地点”发送至10669588128

防伪客服电话

(010)58582300

目录

第1章 绪论	1	2.4.1 线性表的顺序表示与实现	41
1.1 数据抽象与数据结构	1	2.4.2 一元稀疏多项式	45
1.1.1 抽象与结构	1	2.4.3 稀疏矩阵	48
1.1.2 抽象与封装	2	2.5 链栈与链队列	53
1.1.3 程序设计中的抽象	3	2.5.1 链栈	53
1.1.4 数据结构	5	2.5.2 链队列	55
1.2 抽象数据类型与应用程序接口	9	2.6 线性表的链式表示和实现	57
1.2.1 抽象数据类型	9	2.6.1 单链表	57
1.2.2 接口和实现	9	2.6.2 双向链表	64
1.2.3 良好的接口设计规则	10	2.6.3 循环链表	66
1.3 算法和算法分析	12	2.7 线性表两种存储结构的比较	69
1.3.1 算法和算法描述	12	习题2	70
1.3.2 算法分析基础	13	第3章 排序基础	73
1.4 数据结构与算法的描述与实现	18	3.1 排序的概念与分类	73
1.4.1 一维数组	18	3.1.1 排序的概念	73
1.4.2 指针与结构体	21	3.1.2 排序的分类	74
习题1	27	3.2 直接插入排序	75
第2章 线性数据结构	29	3.3 希尔排序	77
2.1 典型线性数据结构	29	3.4 基数排序	80
2.1.1 线性结构的逻辑描述	29	3.4.1 多关键字排序	80
2.1.2 线性结构的存储表示	31	3.4.2 基数排序	81
2.2 顺序栈	32	习题3	86
2.2.1 栈的顺序表示和实现	32	第4章 哈希表	88
2.2.2 应用举例	34	4.1 哈希表的概念	88
2.3 循环队列	37	4.2 哈希函数的构造方法	89
2.3.1 队列的顺序表示	37	4.2.1 直接定址法	89
2.3.2 循环队列的实现	38	4.2.2 除留余数法	90
2.3.3 应用举例	41	4.2.3 数字分析法	91
2.4 顺序表	41		

4.2.4 折叠法	91	6.2.2 链式存储结构	130
4.2.5 平方取中法	92	6.3 遍历二叉树	132
4.3 处理冲突的方法	93	6.3.1 二叉树的递归遍历	133
4.3.1 链地址法	93	6.3.2 二叉树的非递归遍历	134
4.3.2 开放定址法	93	6.3.3 遍历的应用	138
4.4 哈希表的实现	95	6.4 堆	140
4.4.1 链地址哈希表的实现	95	6.4.1 堆的定义	140
4.4.2 开放定址哈希表的实现	97	6.4.2 基本操作的实现	141
4.5 哈希表的查找性能	100	6.4.3 堆排序	145
习题 4	102	6.5 二叉查找树	149
第 5 章 递归	104	6.5.1 二叉查找树的定义	149
5.1 递归基础	104	6.5.2 二叉查找树的查找	150
5.1.1 汉诺塔问题	104	6.5.3 二叉查找树的插入	151
5.1.2 递归函数执行过程	105	6.5.4 二叉查找树的删除	152
5.2 递归与分治	108	6.5.5 二叉查找树的查找性能	155
5.2.1 分治法	108	6.6 平衡二叉树	155
5.2.2 折半查找	109	6.6.1 平衡二叉树的定义	156
5.2.3 归并排序	111	6.6.2 平衡二叉树的失衡及 调整	157
5.2.4 快速排序	113	6.6.3 平衡二叉树的插入	161
5.3 递归与迭代	116	习题 6	163
5.3.1 迭代三要素	116	第 7 章 树和森林	166
5.3.2 迭代与递归的联系与 区别	119	7.1 树的定义	166
5.4 广义表	119	7.2 树的存储结构	167
5.4.1 广义表的定义	120	7.2.1 双亲表示法	167
5.4.2 广义表的存储结构	121	7.2.2 双亲孩子表示法	168
5.4.3 广义表常用操作的实现	123	7.2.3 孩子兄弟表示法	168
习题 5	124	7.3 树和森林的遍历	172
第 6 章 二叉树	126	7.4 并查集	174
6.1 二叉树的概念和性质	126	7.5 B 树	179
6.1.1 二叉树的定义和术语	126	7.5.1 B 树的定义	179
6.1.2 二叉树的性质	127	7.5.2 B 树的查找	180
6.2 二叉树的存储结构	129	7.5.3 B 树的插入	182
6.2.1 顺序存储结构	129	7.5.4 B 树的删除	184

7.5.5 B^+ 树	186	8.3.2 广度优先遍历	204
习题 7	187	8.3.3 遍历的应用	206
第 8 章 图	189	8.4 最小生成树	207
8.1 图的基本概念	189	8.4.1 普里姆算法	207
8.1.1 图的定义	189	8.4.2 克鲁斯卡尔算法	211
8.1.2 图的术语	190	8.5 最短路径	213
8.2 图的存储结构	192	8.6 拓扑排序	216
8.2.1 邻接数组	192	8.7 关键路径	218
8.2.2 邻接表	196	习题 8	224
8.3 图的遍历	202	参考文献	227
8.3.1 深度优先遍历	202		

第1章 绪论

信息的表示和处理是计算机科学技术的基础。计算机加工处理的对象已由单纯的数值扩展到字符、表格、图形、图像、动画、声音和影像等各种结构复杂的多媒体数据。为了有效地设计和实现满足各种应用的计算机软件,必须正确地分析、描述、存储和处理数据对象及其相互关系。研究数据结构和算法是计算机科学的核心问题之一。



本章课件

1.1 数据抽象与数据结构

计算机是一种通用工具。为了用计算机解决特定问题,需要开发相应的软件。在问题分析、模型设计和程序实现等软件工程环节中,最常运用的思想方法有抽象、封装、分类、关联、组装和构造等,其中抽象尤为重要。

1.1.1 抽象与结构

抽象(Abtract)是对一类事物或一个系统的简化描述。它舍弃个别的、非本质的属性,抽取共同的、本质的属性,是形成概念的必要手段。

抽象涉及对一个或一组对象的本质属性的发现和命名。例如,善良、老实和漂亮是人的特性,因此可以作为描述具体某个人的属性值。但是,描述特定场合中的一个人,并不需要罗列全部属性。比如,大学图书馆通常只关注读者的号码、姓名、专业、身份、所借图书等相关属性,不必登记诸如体重或性格等其他无关属性。借书证或校园卡就是图书馆对读者的抽象。人们的各种证件是其面向特定需求的抽象,如身份证、驾驶证、学生证、毕业证、校园卡、银行卡等。

例 1-1 想在图书馆借书,可以先在书目检索柜查阅书目卡片。每张卡片是对一种藏书的面向检索的抽象。卡片上的书目信息有书名、作者名、分类号、出版单位和出版时间等,并列该种图书全部复本的馆藏登录号。书目卡片集合就是对书库的面向检索的抽象。当然,几乎所有图书馆都已配置了更便利的网上查询系统了,但那也是基于对图书馆业务的抽象的。

人的记忆分为短期记忆和长期记忆两个阶段。在短期记忆中,记住事物的细节的能力是很有限的,这对人的智力是极大的限制。通过反复多次对事物细节进行机械地识记而达到长期记忆是一种“死记硬背”,往往“事倍功半”。实际上,人对事物的理解是认识和记忆



视频 1-1:
抽象与数据
结构-1

的基础。人的抽象能力极大地弥补了其记忆上的缺陷。对事物抽象的结果往往表现为发现对象的结构(Structure)。结构是各个组成部分的搭配和排列,可以形式地表示为各成分之间的关系。

例 1-2 用几秒钟就能够记住以下整数集合:

$\{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30\}$

但是,如果要用同样的时间记住并背诵

$\{18, 2, 20, 16, 24, 4, 30, 26, 6, 22, 14, 12, 28, 10, 8\}$

就难多了!前一个集合的元素枚举是有序的,容易发现和记忆它的一个有序结构:2和30之间的有序排列的偶数集。后一个集合不是这样的有序结构,因此不易记住。前者的结构是一个较低级的抽象,可以利用这个抽象建立高级抽象,比如“头15个正偶数之和”。如果记住:前 n 个正整数之和是 $n(n+1)/2$ (这是一个抽象),并且推出:头 n 个正偶数之和是其两倍(用分配律,这也是一个抽象)或者 $n(n+1)$ 。那么,这个和就是 $15 \times 16 = 240$ 。现在,如果发现:第二个数集正是第一个数集的一个排列(抽象),而且一组数之和与其任意排列之和相同(抽象)的事实(复合抽象),就可以发现第二个数集之和也是240(具体事实)。

通常,事物的细节越繁杂,越有必要分析和利用它的结构。这不仅便于记忆和认识事物,可能还更有利于提高处理事物的效率。

例 1-3 图书馆的书目检索卡片通常有多种排列:有的按书名编排,有的按作者编排,还有的按分类号编排,等等。读者不必逐张卡片地顺序查阅,可以利用卡片集合的有序结构,迅速缩小卡片范围,实现高效率的检索。

抽象和结构是人们克服复杂性的重要认知形式。在认识事物时,对结构的分解和合成十分重要。例如,一辆汽车能够描述为“由人操纵的、燃油动力的和陆上行驶的运输设备”。方向盘由人操纵,引擎通常由燃油获得动力,加上车轮和车厢后构成了运输的功能。但是每一部分都可以放到别的机器上,没有哪部分能够单独称为汽车。构成汽车的是所有部件的正确组装(Composition),或者说协同作用(Synergy)。

1.1.2 抽象与封装

对事物的抽象不仅可以表现为结构形式,而且面对特定的需求,可以进一步划分为“外部结构”和“内部结构”。比如各种电器设备,呈现给普通用户的只是外部结构:机箱以及与应用相关的输入输出接口(Interface,也称为界面,如机箱或遥控器的开关、按键、屏幕或插座),厂商提供的使用说明书只涉及如何通过操作接口使用设备的内容。这种隐藏了与使用操作无直接关系的内部结构的做法称为封装(Encapsulation)。通常,对封装良好的设备的一切操纵和信息的输入输出都通过操作接口进行。尽管不同品牌电器的内部配置不同,但它们的操作接口基本一致。不必知道机器功能的实现细节,用户就可以用相似的方法操纵各种品牌的同类机器。封装具有以下几个特点。

- (1) 简单。尽可能易学和易用。
- (2) 完整。封装成一个整体,便于管理和移动。
- (3) 一致。同类设备的内部结构或者功能实现细节可以不同,但可以具有相同或相似的外观和操作接口。
- (4) 保密。不公开内部结构和实现细节。
- (5) 安全。避免对设备内部的不合适操作和破坏。
- (6) 易维护。对设备内部进行专业维修后,设备功能不变,外部结构不变,使用操作也不变。

1.1.3 程序设计中的抽象

现实世界的各种事物都可以称为对象(Object)。例如,一株细菌,一个基因,一本书,一台计算机,一位学生,一座图书馆,一所学校,一架飞机,甚至宇宙中的一个星球,大大小小和形形色色的物体和系统都是对象。同样,各种事件,比如一次借书,一个会议,一场球赛,一台晚会,也是对象。复杂对象由简单对象组成。现实世界的对象可以通过抽象,用某种符号形式表示为计算机程序处理的数据对象(Data Object)。所有数据对象统称为数据(Data)。在特定问题中,基本的数据对象称为数据元素(Data Element)。本书将数据对象也简称为对象。

人们运用抽象、结构和协同等思想方法来构建计算机软件系统。如果不借助结构,就难以直接理解、表示和处理程序的复杂细节。与C、C++、Java和Pascal等高级语言相比,对汇编语言的读写要难得多,因为它要求程序员关注太多的细节描述。高级语言比低级语言体现了更高级的抽象。在程序设计中,使用最多的抽象是功能抽象和数据抽象。此外,面向对象程序设计还采用了更多形式的抽象,如封装、继承和多态等。

1. 功能抽象

用高级语言编写一个函数,实际上是创建执行特定任务的一组指令。任务的细节可能相当复杂,但程序员能够用一个单词或短语给函数命名,用文字简洁地描述函数的功能。函数名是对函数代码复杂细节的抽象。例如,一个计算账目结余的函数可取名为computeBalance。该函数可用账号或名字作为输入参数,并返回账目的结余。

功能抽象是为一段代码命名,以便能用其名字来调用它。功能抽象也称为过程抽象,变量参数化是它的一个主要方面。可以将函数中的常量和变量分离,并对变量初始化。在许多情况下,函数操作的数据由变量组成。参数化后,函数可以得到良好的封装,变得更加通用。

允许在一个函数中调用另一个函数,就有了从低级抽象建立高级抽象的基础。例如,computeBalance函数能够出现在一个工资程序中,工资程序表示了比computeBalance更高级的抽象。

函数的嵌套调用体现了多层抽象。在层次设计中,人们把整个程序设计项目看作一个主函数(main)。可以把功能复杂的函数分解为若干较小的子函数。如果子函数的功能仍然过于复杂,则可以进一步分解。在较高的层次上,设计者只需按功能需要调用低层函数,不必关注其实现细节。

2. 数据抽象

数据抽象(Data Abstraction)可表示为一个三元组:

$$(S, D, P)$$

其中, S 为数据对象的结构, D 为数据对象的取值范围(也称为值域), P 为数据基本操作集。本质上, 数据抽象是把具有相同取值范围 D 并可以进行相同操作 P 的一批数据抽象为一个数据类型, 并限定只能用这些操作来引用和修改数据的值。程序设计语言的类型定义就是这一原则的运用。

数据抽象最本质的是封装, 把数据类型的使用与实现分离, 使得程序设计者能够实现以下操作。

(1) 把大的系统分解为多个小的部分, 每部分具有对所处理的数据的操作接口。

(2) 这些操作接口是其相应部分的功能抽象, 是外部可见的, 而它的具体实现则是对外部隐藏的。

(3) 对外部隐藏数据, 外部只能通过操作接口访问数据。

数据类型(Data Type) T 是一个值域 D 以及 D 上的一组基本操作 P 的集合:

$$T = \{D, P\}$$

例如, C 语言的基本数据类型有整型 **int**, 浮点型 **float**, 字符型 **char**, 等等。整型 **int** 的值域为整数区间 $[-2^{15}, 2^{15}-1]$, 操作集为 $\{+, -, *, /, \%, <<, >>, ==, !=, <, <=, >, >=, \&, \&\&, |, \|, !\}$ 。使用基本数据类型即可编写程序, 但在更高层次数据抽象之上编写程序会更加方便。定义和使用数据类型的过程称为数据抽象。

变量、常量或表达式都具有确定的数据类型, 用以刻画(程序)操作对象的特性。类型明示或隐含地规定了变量或表达式在程序执行期间的可能取值范围, 以及允许对这些值进行的操作。因此, 数据类型是值的集合和在其上定义的一组操作的总称。例如, 高级语言中的整型变量, 其值集为某个区间上的整数(区间大小依赖于不同的编译器和机器), 定义在其上的操作为加、减、乘、除和取模等算术运算。

按“值”的不同特性, 高级语言的数据类型可分为两类: 一类是非结构的原子类型(Atomic Data Type), 该类型的值不可分解, 如 C 语言的基本类型(整型、浮点型和字符型)。另一类是结构类型(Structured Data Type), 该类型的值可以分解, 由若干成分按某种结构组成, 其成分可以是非结构的, 也可以是结构的。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。

实际上, 数据类型的概念并非局限于高级语言中, 每个处理器(包括计算机硬件系统、操作系统、高级语言、数据库等)都提供了一组原子类型或结构类型。例如, 一个计算机硬件系统通常含有“位”“字节”“字”等原子类型, 它们的操作通过一套为计算机设计的指令直接由电路系统完成, 而高级语言提供的数据类型, 其操作需通过编译器或解释器转化成低层的汇编语言或机器语言的数据类型来实现。

为何引入“数据类型”呢?从硬件的角度看,是作为解释计算机内存中信息含义的一种手段;而对使用数据类型的用户来说,则是实现信息的隐蔽,把用户不必了解的细节都封装在类型中。例如,用户在使用“整数”类型时,既不需要了解“整数”在计算机内部是如何表示的,也不需要知道其操作是如何实现的。如“两整数求和”,程序设计者注重的仅仅是其“数学上求和”的抽象特性,而不是其在硬件的“位”操作如何进行。

1.1.4 数据结构

从数据类型的观点看,原子类型变量的值不可分解。在一般情况下,程序设计语言提供的基本数据类型可以满足需求。但有时也需要定义新的原子数据类型,例如 30 位的十进制长整数。结构类型的值可以分解,并可以划分为以下两种类型。

固定聚合类型(Fixed-aggregate Data Type)。该类型变量的值由确定数目的成分按某种结构组成。例如,复数由两个实数依确定的<实部,虚部>次序构成。

可变聚合类型(Variable-aggregate Data Type)。和固定聚合类型相比,构成可变聚合类型“值”的成分的数目不确定。例如,可定义一个“有序整数序列”的抽象数据类型,其中序列的长度是可变的。

一个结构类型的数据对象的结构 S 由其元素集 C 和元素之间的关系集 R 组成:

$$S = (C, R)$$

S 称为数据逻辑结构(Data Logical Structure),简称数据结构(Data Structure)。冠以“逻辑”是因为该结构只是对数据对象的形式描述,与其在计算机内部的物理表示无关。由元素集 C 构建一个数据结构时,通常是根据操作集 P 的需求,定义 C 的元素之间的关系集 R 。这些关系有些是自然存在的,有些则是为了满足某些操作的需要而人为添加的。例如,图书馆为了在读者集进行快速查找,可以在读者之间定义一个查找次序(关系)。

例 1-4 图书馆的书目检索系统问题。如例 1-1 和例 1-3 所述,具有不同编排结构的卡片箱方便了检索。箱内的卡片按某种次序线性排列,是一种线性结构。在书目自动检索系统中,可以如图 1.1 所示,建立一个按登录号顺序排列的书目总表和 3 个分别按书名、作者名和分类号顺序排列的索引表。由这 4 个表构成的文件便是一种书目自动检索模型,计算机的主要操作是按照某个特定要求(如给定书名)对书目文件进行查询。通常,书目自动检索系统还提供按关键字、出版时间等其他多种实用的书目检索功能。类似的系统还有电话自动查号、仓库账目管理等。如果对象之间定义的是线性关系,这类模型可称为线性数据结构。实际的书目自动检索系统不是人工检索的卡片箱的简单实现。为了提高检索效率,通常采用非线性数据结构。



视频 1-2:
抽象与数据
结构-2

登录号	书名	作者	分类号	...
⋮	⋮	⋮	⋮	⋮
A482	C语言程序设计	刘韶涛	S01	...
A483	C++程序设计实用教程	金世双	L01	...
A484	C语言程序设计	吴登峰	S01	...
A485	Java简明教程	皮德常	L02	...
⋮	⋮	⋮	⋮	⋮

书名	登录号	作者	登录号	分类号	登录号
⋮	⋮	⋮	⋮	⋮	⋮
C语言程序设计	A482, A484...	金世双	A483, ...	L	A483, A485...
C++程序设计实用教程	A483, ...	刘韶涛	A482, ...	S	A482, A484...
Java简明教程	A485, ...	皮德常	A485, ...		
⋮	⋮	⋮	⋮	⋮	⋮

图 1.1 图书目录文件和索引表

例 1-5 动物分类。图 1.2 为动物的部分分类示例,其中层次中最高的是动物,位于顶部。动物分成无脊椎动物和脊椎动物两类,各自又分为多个子类。类似的例子还有族谱、组织结构图等。这种层次化分类可采用“树”的数据结构表示。

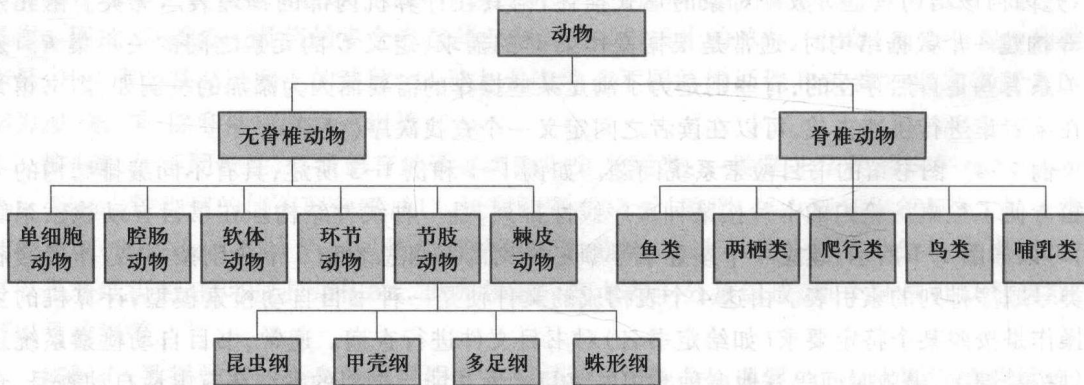


图 1.2 动物的部分分类

例 1-6 城际公路导航系统。在 n 个城市之间建立公路导航系统,输入所在城市和目的城市,导航系统给出两个城市间路程最短的线路。其中 n 个城市可用图中的顶点表示,