



“十二五”普通高等教育本科国家级规划教材

“高等学校本科计算机类专业应用型人才培养研究”项目规划教材

数据结构 ——用 C 语言描述 (第 2 版)

Data Structures in C (Second Edition)

耿国华 张德同 周明全 等编著

高等教育出版社



“十二五”普通高等教育本科国家级规划教材

“高等学校本科计算机类专业应用型人才培养研究”项目规划教材

数据结构

——用 C 语言描述 (第 2 版)

Shuju Jiegou —— Yong C Yuyan Miaoshu

耿国华 张德同 周明全 等编著

高等教育出版社·北京

内容提要

本书根据教育部计算机类专业教学指导委员会制定的“数据结构与算法”课程大纲和《全国硕士研究生入学统一考试计算机专业基础综合考试大纲》的要求而编写,跟踪技术发展需要,教材体系科学,内容简洁实用。

本书共10章,内容包括基本概念、基本结构(线性表、栈与队列、串、数组与广义表、树、图)和基本技术(查找方法与排序方法)三大部分,每章均附有小结与典型题例,并配有大量的例题、习题与实习题,连贯性强。与本书配套的教学资源丰富,按章节知识树的结构构成,包括教学视频、部分扩展内容、课程设计与课程实习指导、多媒体教学课件、算法源码等,适合网络学习使用。

本书既可作为高等学校计算机及相关专业数据结构课程的教材,也可供从事计算机应用开发的工程技术人员参考使用。

图书在版编目(CIP)数据

数据结构:用C语言描述/耿国华等编著. -- 2版

. -- 北京:高等教育出版社,2015.7

ISBN 978-7-04-043305-0

I. ①数… II. ①耿… III. ①数据结构-高等学校-教材 ②C语言-程序设计-高等学校-教材 IV.

①TP311.12②TP312

中国版本图书馆CIP数据核字(2015)第155762号

策划编辑 倪文慧

责任编辑 倪文慧

封面设计 张志

责任印制 赵义民

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 北京东君印刷有限公司
开 本 850mm×1168mm 1/16
印 张 23.5

字 数 520千字
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.landaco.com>
<http://www.landaco.com.cn>
版 次 2011年6月第1版
2015年7月第2版
印 次 2015年7月第1次印刷
定 价 40.00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 43305-00

与本书配套的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、网站登录

1. 访问 <http://abook.hep.com.cn/186946>, 点击“注册”。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”界面。

2. 课程充值: 登录后点击右上角的“充值”图标, 正确输入教材封底标签上的明码和密码, 点击“确定”完成课程充值。

3. 在“我的课程”列表中选择已充值的数字课程, 点击“进入课程”即可开始课程学习。

账号自登录之日起一年内有效, 过期作废。

使用本账号如有任何问题, 请发邮件至: ecourse@pub.hep.cn。



数据结构——用C语言描述 耿国华 主编

用户名 密码 验证码 0625 进入课程

注册

重要通知

因系统升级, 所有用户都需要先注册 (不能用书后的明码密码直接登录)。注册后的用户登录后, 请先点击页面右上方“充值”, 正确输入教材封底标签上的明码和密码完成课程选择。

注册 登录 充值

在“我的课程”列表中选择已充值的数字

数字课程介绍 纸质教材 版权信息 联系方式

书名: 《数据结构——用C语言描述(第2版)》

作者: 耿国华 主编

ISBN: 高等教育出版社

出版日期: 2015年7月

定价:

二、资源使用

与本书配套的易课程数字课程资源按照章、节知识树的形式构成,包括多媒体课件与电子教案、教学视频、算法拓展、示例代码、算法 C 实现模板、课程实习和设计模板范例、试卷及解析、选学内容等,以便读者学习使用。

1. 多媒体课件采用 Flash 动画实现全部算法的执行过程,提供的选件控制方式有助于授课内容重新组合,突出对数据结构课程难点算法本质的理解,可直接用于课堂教学,PPT 电子教案可直接下载使用。

2. 针对教材各章难点及重点提供课程教学视频,可通过在数字课程网站按章节选播或手机扫描教材中二维码的方式进行在线学习。

3. 书中所有算法的函数实现方式(可执行代码、可装配模板及可引用类型头文件等)均可直接下载使用,只需引用相应的类型定义头文件就可直接上机运行。

4. 课程实习和设计模板范例强化了实习和设计的核心内容。

5. 试卷及解析包含多套本科学期样卷及参考答案、近年国内计算机硕士研究生入学统一考试“数据结构”试题与参考答案,便于读者模拟练习和考研参考使用。

6. 选学内容包含 KMP 算法、树的计数等拓展学习内容。

出版说明

信息化社会需要大量的计算机类专业人才。据统计,目前我国计算机类专业布点总数已逾 2 800 个,这些专业点为国家的现代化建设培养了大批计算机类专业人才,其中绝大多数是应用型人才。如何按照社会需求,确定合理的人才培养目标,并在其制导下培养特色突出的应用型人才,是提高教育质量和水平的重要任务。

为了更好地引导高校计算机类各专业点构建有特色的培养方案,例如,能够体现行业特色、区域需求,同时建设体现这些特色的学科基础课和专业课,促进本科计算机类专业应用型人才培养,出版一批体现应用型人才培养特色的新形态教材,教育部高等学校计算机类专业教学指导委员会、全国高等学校计算机教育研究会与高等教育出版社联合组建了“高等学校本科计算机类专业应用型人才培养研究”课题组,基于《计算机类专业教学质量国家标准》,围绕软件工程、网络工程、物联网工程等专业应用型人才培养的研究展开相关工作。

在研究的基础上,课题组汇聚 80 多所高校的教学经验,协同创新,开展了核心课程教学资源建设以及教材建设,这套教材作为课题研究的重要成果之一,具有以下几个显著特点。

- 以课题研制的《高等学校本科计算机类专业应用型人才培养指导意见》为指导,委托有丰富教学实践经验的教师编写,内容覆盖了不同专业的学科基础课、专业核心课及专业方向课。
- 教材内容基于理论适用,突出理论与实践相结合,强调“做中学”,引入丰富的实验案例,摒弃大而全、重理论轻实践的做法,结构新颖、努力突出专业特色。
- 采用纸质教材与数字资源相结合的形式,将教学内容与课程建设充分展示出来,使教师和学生借助网络实现全方位的个性化教学。

相信这套教材的出版能够起到推动各高校计算机类专业建设、提高教学水平和人才培养质量的作用。希望广大教师在教学过程中对教材提出宝贵的意见和建议,使其在使用过程中不断完善。

教育部高等学校计算机类专业教学指导委员会
全国高等学校计算机教育研究会
高等教育出版社
2015年3月

前 言

IEEE-CS/ACM 的 CS 教程已将算法与数据结构课程列为核心课程之首,数据结构愈显出其在信息学科中的重要地位。作为重要的计算机类专业学科基础课,数据结构课程一般开设在大学二年级,应对前两年学习的软件技术进行总结提高,为后续专业课程提供基础。作为计算机科学与技术人才素质培养框架中的中坚课程,它承上启下,贯通始终,是计算机科学与技术人才素质框架中的脊梁,其学习过程体现着创造性思维的信息素质培养过程,对学生的软件开发能力培养至关重要。

本书是“十二五”普通高等教育本科国家级规划教材,根据教育部高等学校计算机类专业教学指导委员会制定的《高等学校计算机科学与技术专业核心课程教学实施方案》和《全国硕士研究生专业综合考试大纲》要求编写而成。本书是在作者编写的《数据结构》(2011年由高等教育出版社出版)教材基础上的更新与提升,紧密跟进学科和技术发展,融入了我们建设国家精品资源共享课的成果。

本书以抽象数据类型为中心,采用面向对象的新观点,将教学内容分为基本概念、基本结构、基本技术三个层次,贯穿了计算机科学中的一些重要的问题求解技术,符合认知规律;使用标准 C 作为算法描述的语言,使之与目前大多数院校讲授的第一语言衔接,便于将读者的注意力集中在算法的理解上。书中给出了大量的 C 函数表示算法,体现数据抽象与过程抽象的结合,突出了算法表示的实质。

本书共 10 章,可分为三部分。其中,第一部分(第 1 章)介绍数据结构的基本概念;第二部分(2~7 章)介绍基本的数据结构,包括线性结构(线性表、栈和队列、串、数组与广义表)与非线性结构(树、图);第三部分(8~10 章)介绍基本技术,包括查找方法与排序方法。书中贯穿了重要的程序设计技术,如参数传递技术、动态处理的指针技术、数组技术(抽象规律处理)和递归技术与队列技术。书中给出的许多精彩典型算法是人们在数据处理中智慧的结晶,我们力求将经典算法的思路表现出来,为学习者继续拓展提供线索。

内容特色:

- 在基本概念部分,为增强对抽象概念的理解,举例说明了数据结构的重要性,增加了算法性能评价和问题规模、算法复杂度分析的范例,针对同一个问题给出了两个复杂度不同的算法,便于直观地理解算法复杂度的概念和算法复杂度分析的方法。

- 在数据结构部分,为方便读者理解与应用抽象数据类型结构,建立完整的结构体系,我们提供了“C 语言实现算法模板示例”。读者可在上机实验前参阅数字课程网站上的相关资源,引用已定义的类型文件嵌入相关函数,实现抽象数据类型重用,从而掌握算法实现的方法。同时,

为紧扣考研大纲要求,我们增补了并查集与等价类划分相关内容。

- 在基本技术部分,突出了典型技术的优化处理技术,增加了数据结构典型技术的新应用,引入了B树动态索引技术应用示例,以便读者理解构建目前使用的查找索引数据结构。为引导学习者注重掌握排序思想与改进思路,加入对每类排序方法的小结。

- 全书各章均有总结与典型题例,形成梯度,便于引导提高。与前版教材相比,增大了习题与实习题的比例,对综合性强的实习加入部分提示与指导,注重引导能力的培养。

资源特色:

鉴于数据结构课程的技术性特点,同时为适应翻转课堂的教学需求,以及移动学习的环境变化,我们充实了与教材配套的数字课程资源,通过课程教学视频和课程资源网站,提供师生之间的课程教学资源与交流平台。

针对教材各章难点及重点提供了课程教学视频,可通过在数字课程网站按章节选播或手机扫描二维码的方式进行在线学习。

配套数字课程网站含有丰富的教学资源,包括多媒体教学课件与电子教案、算法实现和课程设计与实习范例、选学内容、试卷及解析等内容。其他教学资源可参考国家精品资源共享课网站,网址 http://www.icourses.cn/coursestatic/course_4246.html。

本书同时也是国家精品课、国家精品资源共享课及MOOC课程“数据结构”的配套主讲教材。

本书由耿国华主编,第1章、第6章、第7章、第9章由耿国华编写,第5章、第8章由张德同编写,第3章、第10章由周明全编写,第2章由卢燕宁编写,第4章由刘晓宁编写,全书由耿国华统稿。刘晓宁、李康老师负责数字课程资源建设,多名研究生参加了算法调试与动画多媒体课件制作。

本书前版教材已应用于多所学校的电子、数学、自动化、通信、计算机等专业的教学过程,在这里衷心感谢广大读者对我们的厚爱与信任。真诚期待大家在使用本教材的过程中提出意见和建议,使之在教学过程中逐步改进完善,更好地服务于教学。

耿国华
2015.6.20

目 录

| | | | |
|--------------------------|----|----------------------------|-----|
| 第 1 章 绪论 | 1 | 基本运算 | 42 |
| 1.1 数据结构的基础概念 | 1 | 2.3 线性表的链式存储 | 47 |
| 1.2 数据结构的内容 | 7 | 2.3.1 单链表 | 47 |
| 1.3 算法 | 9 | 2.3.2 单链表上的基本运算 | 49 |
| 1.4 算法描述 | 11 | 2.3.3 循环链表 | 57 |
| 1.5 算法性能评价 | 12 | 2.3.4 双向链表 | 59 |
| 1.5.1 算法的时间性能分析 | 12 | 2.3.5 静态链表 | 62 |
| 1.5.2 算法的空间性能分析 | 17 | 2.4 线性表应用——一元多项式的 表示及相加 | 64 |
| 1.5.3 算法性能选择 | 18 | 2.5 顺序表与链表的综合比较 | 68 |
| 1.6 数据结构与 C 语言表示 | 18 | 2.5.1 顺序表和链表的比较 | 68 |
| 1.6.1 数据结构与程序设计的 关联性 | 18 | 2.5.2 线性表链式存储方式的 比较 | 69 |
| 1.6.2 结构化程序设计与函数 的模块化 | 19 | 2.6 总结与提高 | 70 |
| 1.6.3 面向对象与抽象数据类型 | 20 | 2.6.1 主要知识点 | 70 |
| 1.6.4 算法描述规范与设计风格 | 25 | 2.6.2 典型题例 | 71 |
| 1.7 关于学习数据结构 | 32 | 习题 | 74 |
| 1.8 总结与提高 | 35 | 实习题 | 76 |
| 习题 | 35 | 第 3 章 限定性线性表——栈与队列 | 78 |
| 实习题 | 37 | 3.1 栈 | 78 |
| 第 2 章 线性表 | 38 | 3.1.1 栈的定义 | 78 |
| 2.1 线性表的概念及其抽象数据 类型定义 | 38 | 3.1.2 栈的表示和实现 | 80 |
| 2.1.1 线性表的逻辑结构 | 38 | 3.1.3 栈的应用举例 | 86 |
| 2.1.2 线性表的抽象数据 类型定义 | 39 | 3.1.4 栈与递归的实现 | 90 |
| 2.2 线性表的顺序存储 | 40 | 3.2 队列 | 97 |
| 2.2.1 线性表的顺序存储结构 | 40 | 3.2.1 队列的定义 | 97 |
| 2.2.2 线性表顺序存储结构上的 | | 3.2.2 队列的表示和实现 | 99 |
| | | 3.2.3 队列的应用举例 | 103 |
| | | 3.3 总结与提高 | 107 |

| | | | |
|------------------------------|-----|-------------------------------|-----|
| 3.3.1 主要知识点 | 107 | 6.2 二叉树 | 157 |
| 3.3.2 典型题例 | 108 | 6.2.1 二叉树的定义与基本操作 | 157 |
| 习题 | 110 | 6.2.2 二叉树的性质 | 158 |
| 实习题 | 111 | 6.2.3 二叉树的存储结构 | 160 |
| 第4章 串 | 113 | 6.3 二叉树的遍历与线索化 | 162 |
| 4.1 串的基本概念 | 113 | 6.3.1 二叉树的遍历 | 162 |
| 4.2 串的存储实现 | 115 | 6.3.2 遍历算法应用 | 165 |
| 4.2.1 定长顺序串 | 115 | 6.3.3 基于栈的递归消除 | 170 |
| 4.2.2 堆串 | 119 | 6.3.4 线索二叉树 | 175 |
| 4.2.3 块链串 | 121 | 6.3.5 由遍历序列确定二叉树 | 179 |
| 4.3 串的应用举例:简单的 行编辑器 | 122 | 6.4 树、森林和二叉树的关系 | 180 |
| 4.4 总结与提高 | 123 | 6.4.1 树的存储结构 | 181 |
| 4.4.1 主要知识点 | 123 | 6.4.2 树、森林与二叉树的相互 转换 | 183 |
| 4.4.2 典型题例 | 124 | 6.4.3 树与森林的遍历 | 186 |
| 习题 | 125 | 6.5 哈夫曼树及其应用 | 188 |
| 实习题 | 126 | 6.5.1 哈夫曼树 | 188 |
| 第5章 数组与广义表 | 127 | 6.5.2 哈夫曼编码 | 193 |
| 5.1 数组的定义与运算 | 127 | *6.6 并查集与等价类划分 | 198 |
| 5.2 数组的顺序存储与实现 | 129 | 6.7 总结与提高 | 203 |
| 5.3 特殊矩阵的压缩存储 | 132 | 6.7.1 主要知识点 | 203 |
| 5.3.1 规律分布的特殊矩阵 | 132 | 6.7.2 典型题例 | 204 |
| 5.3.2 稀疏矩阵 | 135 | 习题 | 207 |
| 5.4 广义表 | 144 | 实习题 | 209 |
| 5.4.1 广义表的概念 | 144 | 第7章 图 | 211 |
| 5.4.2 广义表的存储结构 | 145 | 7.1 图的定义与基本术语 | 211 |
| 5.4.3 广义表的操作实现 | 147 | 7.1.1 图的定义 | 211 |
| 5.5 总结与提高 | 149 | 7.1.2 基本术语 | 213 |
| 5.5.1 主要知识点 | 149 | 7.2 图的存储结构 | 216 |
| 5.5.2 典型题例 | 150 | 7.2.1 邻接矩阵表示法 | 216 |
| 习题 | 151 | 7.2.2 邻接表表示法 | 219 |
| 实习题 | 152 | 7.2.3 十字链表 | 221 |
| 第6章 树与二叉树 | 153 | *7.2.4 邻接多重表 | 223 |
| 6.1 树的定义与基本术语 | 153 | 7.3 图的遍历 | 225 |

| | | | |
|-----------------------|-----|--------------------------|-----|
| 7.3.1 深度优先搜索 | 225 | 9.2 插入类排序 | 318 |
| 7.3.2 广度优先搜索 | 229 | 9.2.1 直接插入排序 | 319 |
| 7.4 图的应用 | 231 | 9.2.2 折半插入排序 | 321 |
| 7.4.1 图的连通性问题 | 231 | 9.2.3 希尔排序 | 322 |
| 7.4.2 有向无环图的应用 | 239 | 9.2.4 小结 | 325 |
| 7.4.3 最短路径问题 | 249 | 9.3 交换类排序 | 326 |
| 7.5 总结与提高 | 255 | 9.3.1 冒泡排序 | 326 |
| 7.5.1 主要知识点 | 255 | 9.3.2 快速排序 | 328 |
| 7.5.2 典型题例 | 256 | 9.3.3 小结 | 331 |
| 习题 | 259 | 9.4 选择类排序 | 332 |
| 实习题 | 262 | 9.4.1 简单选择排序 | 332 |
| 第8章 查找 | 264 | 9.4.2 树形选择排序 | 333 |
| 8.1 查找的基本概念 | 264 | 9.4.3 堆排序 | 335 |
| 8.2 基于线性表的查找法 | 265 | 9.4.4 小结 | 341 |
| 8.2.1 顺序查找法 | 265 | 9.5 归并排序 | 342 |
| 8.2.2 折半查找法 | 266 | 9.6 分配类排序 | 344 |
| 8.2.3 分块查找法 | 269 | 9.6.1 多关键字排序 | 344 |
| 8.3 基于树的查找法 | 271 | 9.6.2 链式基数排序 | 345 |
| 8.3.1 二叉排序树 | 271 | 9.6.3 基数排序的顺序表实现 | 349 |
| 8.3.2 平衡二叉排序树 | 278 | 9.7 各种排序方法的综合比较 | 349 |
| * 8.3.3 B树 | 288 | 9.8 总结与提高 | 351 |
| 8.4 计算式查找法——哈希法 | 298 | 9.8.1 主要知识点 | 351 |
| 8.4.1 哈希函数的构造方法 | 300 | 9.8.2 典型题例 | 352 |
| 8.4.2 处理冲突的方法 | 302 | 习题 | 355 |
| 8.4.3 哈希表的查找过程 | 304 | 实习题 | 357 |
| 8.4.4 哈希法性能分析 | 305 | * 第10章 外部排序 | 358 |
| 8.5 总结与提高 | 308 | 10.1 外排序的基本方法 | 358 |
| 8.5.1 主要知识点 | 308 | 10.1.1 磁盘排序 | 358 |
| 8.5.2 典型题例 | 311 | 10.1.2 磁带排序 | 359 |
| 习题 | 314 | 10.2 总结与提高 | 360 |
| 实习题 | 315 | 习题 | 360 |
| 第9章 内部排序 | 317 | 参考文献 | 361 |
| 9.1 排序的基本概念 | 317 | | |

第1章 绪 论

陈火旺院士把计算机 60 多年的成就概括为五个“一”：开辟一个新时代——信息时代，形成一个新产业——信息产业，产生一个新学科——计算机科学与技术，开创一种新的科研方法——计算方法，开辟一种新文化——计算机文化。这一概括深刻阐明了计算机对社会发展所产生的广泛而深远的影响。

数据结构和算法是计算机求解问题过程的两大基石。著名的计算机科学家 P. Wegner 指出，“在工业革命中起核心作用的是能量，而在计算机革命中起核心作用的是信息”。计算机科学就是“一种关于信息结构转换的科学”。信息结构(数据结构)是计算机科学研究的基本课题，数据结构又是算法研究的基础。

关于数据结构理论的研究，可以追溯到 1972 年 C. A. R. Hoare 奠基性的论文“数据结构笔记”；而现代计算机所大量采用的各种数据结构，最早的系统论述应归于 D. E. Knuth 的名著《计算机程序设计技巧》。随着计算机科学的飞速发展，数据结构的基础研究也逐渐走向成熟。

在开始讲授数据结构相关内容之前，首先需要回答以下问题：

- ① 什么是数据结构？（定义）
- ② 数据结构研究什么？（内容）
- ③ 采用什么方法进行研究？（方法）
- ④ 采用什么工具对算法规则进行描述？（描述）
- ⑤ 如何对算法做性能评价？（评价）
- ⑥ 数据结构和 C 语言的关系是什么？（工具要点、规范要求）
- ⑦ 学习数据结构有什么作用？（重要性）

本章将通过对这些问题与概念的简要介绍，描述数据结构基本内容、主要概念与描述规范，作为本门课程的梗概之序。

1.1 数据结构的基础概念

下面首先介绍数据结构的相关术语。

1. 数据(Data)

数据是描述客观事物的数值、字符以及能输入机器且能被处理的各种符号集合。换句话说，数据是对客观事物采用计算机能够识别、存储和处理

电子教案
数据结构的
基础概念

的形式所进行的描述。简而言之,数据就是计算机化的信息。

数据的概念经历了与计算机发展相类似的发展过程。计算机一问世,数据作为程序的处理对象随之产生。早期计算机主要应用于数值计算,数据量小且结构简单,数据仅有进行算术运算与逻辑运算的需求,数据类型只有整型、实型、布尔型,那时程序员把主要精力放在程序设计的技巧上,而并不重视如何在计算机中组织数据。

随着计算机软件、硬件的发展与应用领域的不断扩大,计算机应用领域发生了战略性转移,非数值运算与处理所占的比例越来越高,数据的概念被大大拓展了。它包含数值、字符、声音、图像等一切可以输入到计算机中的符号集合,各种信息通过编码被划归到数据的范畴,大量复杂的非数值数据处理需求使数据的有效组织显得越来越重要。20 世纪 70 年代后,微型计算机的普及以及数据库、人工智能等研究进一步推动了计算机技术的发展,人们越来越重视运用科学工具来探索数据和程序的内部关系以及它们之间的关系,采用新的观点来设计计算机体系,使计算技术发展为一门科学。

数据的概念不再是狭义的,数据已由纯粹的数值概念发展为图像、字符、声音等各种符号。例如,对于 C 语言的编译程序来说,它加工的数据是字符流的源程序(.c),输出的结果是目标程序(.obj);对于 C 语言的链接程序来说,它加工的数据是目标程序(.obj),输出的结果是可执行程序(.exe),如图 1.1 所示。



图 1.1 C 编译过程示意图

而对于整个 C 编译过程而言,由于它在操作系统控制下接受操作系统的调度,因此相对操作系统来说它又是数据。

2. 数据元素 (Data Element)

数据元素是组成数据的基本单位,是数据集合的个体,在计算机中通常作为一个整体进行考虑和处理。一个数据元素可由一个或多个数据项组成,数据项(Data Item)是有独立含义的最小单位,此时的数据元素通常称为记录(Record)。如表 1.1 所示,学籍表是数据,每一个学生的记录就是一个数据元素。

表 1.1 学 籍 表

| 学 号 | 姓 名 | 性 别 | 籍 贯 | 出生年月 | 住 址 |
|-----|-----|-----|-----|---------|-----|
| 101 | 赵虹玲 | 女 | 河北 | 1983.11 | 北京 |
| ... | ... | ... | ... | ... | ... |

数据项
 ↓
 记录 ←

3. 数据对象 (Data Object)

数据对象是性质相同的数据元素的集合,是数据的一个子集。例如,整数数据对象是集合 $N = \{0, \pm 1, \pm 2, \dots\}$, 字母字符数据对象是集合 $C = \{A, B, \dots, Z\}$, 表 1.1 中所示的学籍表也可看作一个数据对象。由此可看出,不论数据元素集合是无限集(如整数集)、有限集(如字符集),还是由多个数据项组成的复合数据元素(如学籍表),只要性质相同,都属于同一个数据对象。

综上所述,数据概念归纳如下:

| | | |
|------|---|------------------------------|
| 数据特点 | } | 可被计算机接收(与计算机的关联性) |
| | | 可被加工(能被处理) |
| 数据构成 | } | 数据元素——组成数据基本单位(是数据集合的个体) |
| | | 数据对象——性质相同的数据元素的集合(是数据集合的子集) |

4. 数据结构 (Data Structure)

数据结构是指相互之间存在一种或多种特定关系的数据元素集合,数据结构应该包括数据元素集合及元素间关系的集合,即数据的组织形式。由此可见,计算机所处理的数据并不是数据的简单汇集,而是具有内在联系的数据集合。例如,图书馆中的图书并不是书的简单堆积,而是带有相应结构关系的书的集合。再如,表结构(如表 1.1 所示的学籍表)、树状结构(如图 1.2 所示的学校组织层次结构图)、图结构(如图 1.3 所示的交通流量图)。



微视频
什么是数据
结构

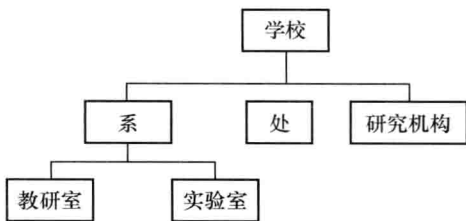


图 1.2 学校组织层次结构图

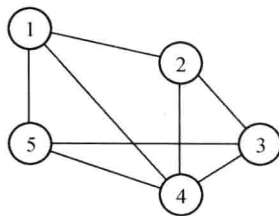


图 1.3 交通流量图

数据结构要研究数据元素之间的相互关系与组织方式,以及对其施加的运算及运算规则,并不关心数据元素的具体内容是什么值。例如,一维数组是向量 $A = (a_1, \dots, a_n)$ 的存储映像,引用时采用下标变量 $A[i]$ 的方式,关注其按序排列、按行存储的特性,并不关心 $A[i]$ 中存放的具体值。同理,二维数组 $A[i, j]$ 是矩阵 $A_{m \times n}$ 的存储映像,我们关心结构关系的特性而不涉及其数组元素本身的内容。

5. 数据类型 (Data Type)

数据类型是一组性质相同的值集合以及定义在这个值集合上的一组操作的总称。数据类型中定义了两个集合,即该类型的取值范围以及该类型中可允许使用的一组运算。例如,高级语言中的数据类型就是已经实现的数据结构的实例。从这个意义上讲,数据类型是高级语言中允许的变量种类,是程序设计语言中已经实现的数据结构(即程序中允许出现的数据形式)。例如,

在 C 语言中,整型类型的取值范围为 $-32\ 767 \sim +32\ 768$,可用的运算符集合为加、减、乘、除、取模(即+、-、*、/、%)。

从硬件的角度来看,它们的实现涉及字、字节、位、位运算等;从用户观点来看,并不需要了解整数在计算机内如何表示、运算细节如何实现,用户只需要了解整数运算的外部运算特性,就可运用高级语言进行程序设计。引入数据类型的目的,从硬件的角度是将其作为解释计算机内存中信息含义的一种手段,对使用数据类型的用户来说则实现了信息隐蔽,将一切用户不必关心的细节封装在类型中。如两整数求和问题,用户仅关注其数学求和的抽象特性,而不必关心加法运算涉及的内部位运算实现细节。

按“值”的不同特性,一般来说,高级程序设计语言中的数据类型可分为两大类:一类是原子类型,原子类型的值是不可再分的,如 C 语言中提供的整型、实型和字符型及指针;另一类是结构类型,结构类型的值是由若干成分按某种结构组合而成的,因此是可以分解的,并且它的成分可以是原子的,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,也可以是数组等其他类型。在高级程序设计语言中,数据类型包含系统定义的标准类型和用户自定义类型这两大类,对系统定义的标准类型(如 C 语言中的 int 类型)用户只需按规定的符号形式直接使用,而用户自定义类型必须由用户先定义后使用,用户在系统提供标准类型的基础上根据需要进行组合、构造新的类型(如 C 语言中提供的 typedef 定义形式)。

思考题: C 语言中的指针类型属于原子类型还是结构类型?

6. 抽象数据类型

抽象的本质是抽取反映问题的本质点,而忽略非本质的细节,这是从事计算机研究的重要方法。

(1) 数据的抽象

计算机中使用的是二进制数,汇编语言中则可给出各种数据的十进制表示,如 98.65、9.6E3 等,它们是二进制数据的抽象;程序设计人员在编程时可以直接使用,不必考虑实现细节。在高级语言中,则给出更高一级的数据抽象,出现了数据类型,如整型、实型、字符型等。抽象数据类型出现后,可以进一步定义更高级的数据抽象,如各种表、队、栈、树、图、窗口、管理等,这种数据抽象的层次为设计者提供了更有力的手段,使得设计者可以从抽象的概念出发,从整体考虑,然后自顶向下、逐步展开,最后得到所需结果。可以这样看,高级语言中提供整型、实型、字符、记录、文件、指针等多种数据类型,可以利用这些类型构造出像栈、队列、树、图等复杂的抽象数据类型。

(2) 抽象数据类型

抽象数据类型(Abstract Data Type, ADT)定义了一个数据对象、数据对象中各元素间的结构关系以及一组处理数据的操作。从某种意义上讲,抽象数据类型和数据类型实质上是一个概念,只不过 ADT 更广义,不仅限于各种不同的计算机处理器中已定义并实现的数据类型,还包括设计软件系统时用户自己定义的复杂数据类型。

ADT 包括定义和实现两方面,其中定义是独立于实现的。定义仅给出一个 ADT 的逻辑特性,不必考虑如何在计算机中实现。

抽象数据类型的定义取决于客观存在的一组逻辑特性,而与其在计算机内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,都不会影响其外部使用,从而为实现软件的部件化和可重用性提供了理论保证,进而提高了软件生产率。

ADT 通常是指由用户定义且用以表示应用问题的数据模型,由基本的数据类型组成,并包括一组相关服务操作。本门课程中将要学习的表、堆栈、队列、串、树、图等结构就是一个个不同的抽象数据类型。以盖楼为例,直接用砖头、水泥、沙子来盖,不仅建造周期长,且建造高度规模受限。如果使用按规范生产成型的水泥预制板,不仅可以高速、安全地建造高楼,水泥预制板使高楼的接缝量大大减少,从而降低了建造高楼的复杂度。由此可见,抽象数据类型是大型软件构造的模块化方法,典型的表、堆栈、队列、串、树、图抽象数据类型,就相当于设计大型软件的“水泥预制板”,用这些已由专门公司设计好的抽象数据类型就可以安全、快速、方便地设计功能复杂的大型软件。

抽象数据类型最重要的特点是数据抽象与信息隐蔽,抽象的本质是抽取反映问题的本质点,忽视非本质的细节,从而使设计的数据结构更具一般性,可以解决一类问题。信息隐蔽就是对用户隐藏数据存储和操作实现的细节,使用者仅需了解操作或服务,通过使用界面服务来访问数据,如图 1.4 所示。其中,Request() 用于检查书库,回答读者所借图书是否存在;Retrieve() 用于从书库取书,送书给读者;Return() 用于将书放入书库,将读者所还书放入书架。

抽象数据类型的特征是使用与实现分离,实现封装和信息隐蔽,也就是说,在抽象数据类型设计时,类型的定义与其实现分离。

数据类型的抽象层次越高,含有该抽象数据类型的软件复用程度就越高。ADT 定义该抽象数据类型需要包含哪些信息,并根据功能确定公共界面的服务,使用者可以使用公共界面中的服务对该抽象数据类型进行操作。从使用者的角度看,只要了解该抽象数据类型的规格说明,就可以利用其公用界面中的服务来使用这个类型,不必关心其物理实现,从而集中考虑如何解决实际问题。

ADT 物理实现作为私有部分封装在其实现模块内,使用者不能看到,也不能直接操作该类型所存储的数据,只有通过界面中的服务来访问这些数据。从实现者的角度来看,把抽象数据类型的物理实现封装起来,有利于编码、测试,也有利于修改。当需要改进数据结构时,只要界面服务的使用方式不变,只要改变抽象数据类型的物理实现,而无须修改所有使用该抽象数据类型的程序,这样就会提高系统的稳定性。

ADT 包括数据对象、数据元素间的结构关系、操作三个部分,其定义格式可以有多种,本书采用下述格式定义 ADT。

ADT <ADT 名>

{ 数据对象:<数据对象的定义>

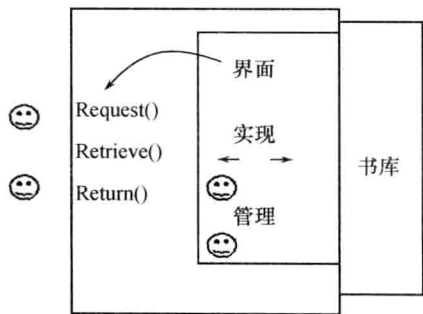


图 1.4 信息隐蔽示意图

结构关系: <结构关系的定义>

基本操作: <基本操作的定义>

| ADT <ADT 名>

其中,

- ① 数据对象定义应在已有数据类型或已定义数据对象基础上对新的数据对象进行定义。
- ② 数据对象和结构关系的定义采用数学符号和自然语言描述。
- ③ 基本操作定义包括操作名称、参数表、初始条件和操作结果四部分内容的定义和描述。

基本操作的定义格式为:

<操作名称>(参数表)

操作前提: <操作前提描述>

操作结果: <操作结果描述>

下面给出问题求解过程的两个示例,从中体会数据结构对解决问题的重要作用。

例 1.1 电话号码查询问题。

【问题描述】

编写程序实现对某市或某单位个人电话信息的查询功能。具体要求如下:

1. 按姓名进行查询。
2. 若该人装有电话,则显示其电话号码;若尚未安装,则显示提示信息。

【问题抽象】 从实际问题出发,可将市内(单位内)已装电话人员信息抽象为如表 1.2 所示结构。

表 1.2 电话簿结构

| 姓名 | 电话号码 |
|----|---------|
| 张三 | 8830259 |
| 王四 | 8830251 |

【求解方法】 可有顺序表法及索引结构法两种求解方法。

顺序表法:可用数组存储整个表,每个人的信息(即一个人的姓名和电话号码)就是数组中的一个元素。从表的第一项开始依次查找姓名,直至找到或是确定无此姓名为止。这种依次查找方法对于一个规模不大的单位是可行的,但对拥有上百万人电话的城市来说效率太低,应当构建新的数据结构支持快速查找。

索引结构法:采用类似于汉语字典的组织形式,按姓氏拼音或姓氏笔画排列构建表,并构造其姓氏或笔画索引表。查找时只需先在索引表中查找姓氏,然后根据索引表指示在登记表中核查姓名,从而提高了查找效率,如图 1.5 所示。