



“十二五”普通高等教育本科国家级规划教材
“十二五”江苏省高等学校重点教材

数据结构 (Java版) (第4版)

◆ 叶核亚 编著 ◆ 陈本林 主审



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

“十二五”普通高等教育本科国家级规划教材
“十二五”江苏省高等学校重点教材

数据结构 (Java 版)

(第 4 版)

叶核亚 编著

陈本林 主审

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内容简介

本书是“十二五”普通高等教育本科国家级规划教材。

本书全面系统地介绍数据结构的基础理论和算法设计方法,包括线性表、树、图等数据结构以及查找和排序算法。本书采用 Java 语言以面向对象方法设计并实现了全部的数据结构及算法。本书精选基础理论内容,重点是数据结构设计和算法设计,通过降低理论难度和抽象性、加强实践环节等措施,力求增强学生的理解能力和应用能力。本书内容涉及的广度和深度符合本科培养目标的要求,配套教学资源丰富。

本书可作为普通高等学校计算机及相近专业本科的数据结构课程教材,也可作为从事计算机软件开发和工程应用人员的参考书。

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。

图书在版编目(CIP)数据

数据结构:Java 版 / 叶核亚编著. —4 版. —北京:电子工业出版社,2015.7

ISBN 978-7-121-26188-6

I. ① 数… II. ① 叶… III. ① 数据结构—高等学校—教材 ② JAVA 语言—程序设计—高等学校—教材
IV. ① TP311.12 ② TP312

中国版本图书馆 CIP 数据核字(2015)第 117514 号

策划编辑:章海涛

责任编辑:章海涛 特约编辑:张 玉

印 刷:涿州市京南印刷厂

装 订:涿州市京南印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱邮编 100036

开 本:787×1092 1/16 印张:21.25 字数:590 千字

版 次:2004 年 5 月第 1 版

2015 年 7 月第 4 版

印 次:2015 年 7 月第 1 次印刷

定 价:45.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

第4版前言

本书是“十二五”普通高等教育本科国家级规划教材，可作为普通高等学校计算机及相近专业本科的“数据结构”课程的教材。

数据结构是软件设计的重要理论和实践基础，数据结构设计和算法设计是软件系统设计的基础和核心。“数据结构”课程讨论的知识内容是软件设计的理论基础，介绍的技术方法是软件设计中使用的�基本方法。“数据结构”是理论与实践并重的课程，既要掌握数据结构的基础理论知识，又要掌握运行和调试程序的基本技能。因此，“数据结构”课程在计算机学科本科培养体系中具有非常重要的地位，是计算机类各专业的核心课程，是培养程序设计能力必不可少的重要环节。

“数据结构”课程内容多，概念抽象，理论深奥，递归算法难度大，是计算机专业最难学的课程之一。根据计算机专业本科的培养目标，本书内容覆盖了数据结构课程的基础理论和典型算法，重点突出数据结构设计和算法设计，通过降低理论难度和抽象性，加强实践环节等措施，增强学生的理解能力和应用能力。配套教学资源齐全。

本书特色说明如下。

(1) 内容全面、理论基础厚实

本书全面系统地介绍数据结构的基础理论和算法设计方法，阐明线性表、树、图等数据模型的逻辑结构，讨论它们在计算机中的存储结构，讨论每种数据结构所能进行的多种操作，以及这些操作的算法设计和实现；针对软件设计中应用频繁的查找和排序问题，根据不同数据结构对操作的实际需求，给出多种查找和排序算法，并分析算法的执行效率。

本书内容选择适合工科院校，理论叙述精练、简明扼要，结构安排合理、由浅入深、层次分明、重点突出，算法分析透彻，程序结构严谨规范，内容涉及的广度和深度符合计算机专业本科培养目标。

(2) 采用 Java 语言描述数据结构和算法

描述数据结构和算法所采用的程序设计思想和方法，以及实现它们所采用的程序设计语言，都必须随着软件时代的发展而发展。面向对象程序设计方法是目前软件开发的主流方法；Java 语言是目前功能最强、应用最广泛的一种完全面向对象程序设计语言，具有成熟而严密的语法体系和强大的应用系统设计能力，其特有的面向对象、跨平台特性、内存自动管理、异常处理、多线程等机制，使其更健壮、更安全、更高效。今日 Java 应用无处不在，其重要性毋庸置疑。

Java 语言放弃了 C++语言的全局函数、结构、指针、友元类、运算符重载、多重继承等，这些丝毫没有影响 Java 功能，而 Java 提供的机制具有更强的功能和更高的性能。Java 提倡简单性原则，对一个问题只提供一种简单精炼的表达方法，这样使程序简单、直接并且不造成歧义，它所提供的各种机制均体现出简单性原则。例如，使用下标形式对数组元素进行操作，则不需要使用指针；方法（函数）采用返回值或引用类型参数返回结果，也不需要使用指针；有了类，则不需要结构类型；构造方法采用重载方式，则不需要采用参数默认值形式，避免产生歧义；通过成员方法实现类的操

作，则不需要运算符重载；通过“单重继承+接口”方式实现多重继承，通过内存自动管理机制自动释放动态分配的存储空间等。

Java 语言提供类的封装、继承、多态和抽象特性，提供数组和对象引用模型，提供具有类模板功能的泛型，以及函数能够实现递归等等，这些不仅具备表达数据结构和算法的基本要素，而且能使算法更简明、更直接，性能更好。因此，本书采用 Java 语言描述数据结构和实现算法不仅可行，作为面向对象的程序设计方法训练也是十分恰当的，更是数据结构课程教学改革的必然发展趋势，完全符合本科培养目标的要求。依托一种功能强大的程序设计语言，充分表达和实现复杂的算法设计，体现软件模块化、可重用的设计思想，是提高程序设计能力的有效手段。

本书充分展示了使用 Java 语言的优越性。

① 在设计线性表、二叉树、树、图等数据结构时，以泛型类为设计基础，将封装、继承、多态、抽象等面向对象技术贯穿始终，采用面向对象思想刻画和表达数据结构设计，这是这个时代的软件设计所必需的。

② 以 Java 的集合框架为背景，采用多种数据结构实现了多种不同特性的集合类，以及各种迭代器。并将 Java 集合运用于实际工程应用。

(3) 加强实际应用和实践环节

理论与实践相结合是提高软件设计能力的有效途径。数据结构是一门理论和实践紧密结合的课程，要在透彻理解理论知识的基础上，通过实践性环节，逐步锻炼程序设计能力。

本书不仅注重传授基础理论知识，更注重在实践环节中培养程序设计的基本技能。将程序设计能力的锻炼和提高设计为一个循序渐进的过程，从原理叙述、例题、思考题等课堂讲授环节，到课后习题、上机实验、课程设计等实践性环节，精心安排“先见识、再模仿、最后自主创新设计”的学习过程。精选并设计一系列例题、习题、实验题、课程设计题等，选题之间前后衔接，逐步深入，层层推进，既突出 Java 语言的特点，又与现实生活息息相关，使原本枯燥难懂的理论变得生动，充分说明数据结构和算法必要性，以及对实际应用程序设计的指导作用；同时，还引导学生在生活中发现问题并解决问题，提高学习兴趣，力求在潜移默化中使学生理解和体会理论知识的重要性，并掌握如何使用它们的方法。

每章的实验题均给出详细的训练目标、设计内容和设计要求，针对课程设计实践性环节，给出多种算法设计与分析的综合应用程序设计实例，详细说明需求方案、设计思想、模块划分、功能实现、调试运行等环节的设计方法，贯彻理论讲授和案例教学相结合的教学方法。

程序设计有其自身的规律，不是一蹴而就的，也没有捷径。程序员必须具备基本素质，必须掌握程序设计语言的基本语法以及算法设计思想和方法，并且需要积累许多经验。这个逐步积累的过程需要一段时间，需要耐心，厚积而薄发。

本书第 1~9 章是“数据结构”课程的主要内容，包括线性表、树、图等数据结构设计以及查找和排序算法设计；第 10 章综合应用设计，包括课程设计的范例和参考选题。

本书所有程序已在 MyEclipse 2014 (JDK 8u25) 集成开发环境中调试通过。熟练掌握集成开发环境的各种操作以及程序调试技术是程序设计的一项基本技能，也需要经过一个逐步积累的过程。

使用 JDK 和 MyEclipse 集成开发环境的详细说明见《Java 程序设计实用教程（第 4 版）》（ISBN 978-7-121-21733-3，叶核亚编著，“十二五”普通高等教育本科国家级规划教材）。

本书由叶核亚编著，在写作过程中得到了许多帮助和支持。感谢南京大学计算机科学与技术系陈本林教授，陈老师认真细致地审阅了全稿，提出了许多宝贵意见；感谢黄纬、徐金宝、彭焕峰、廖雷、阚建飞、陈立等老师给予的帮助，感谢电子工业出版社的支持，感谢众多读者朋友对本书前 3 版提出的宝贵意见。

对书中存在的不妥与错漏之处，敬请读者朋友批评指正。

本书为任课教师提供配套的教学资源（包含电子教案和例题源代码），需要者可[登录华信教育资源网](http://www.hxedu.com.cn)（<http://www.hxedu.com.cn>），注册之后进行[下载](#)。

读者反馈：unicode@phei.com.cn，或 yeheya@x263.net。

作 者

目 录

第 1 章 绪论	1
1.1 数据结构的基本概念	1
1.1.1 为什么要学习数据结构	1
1.1.2 什么是数据结构	1
1.1.3 数据类型与抽象数据类型	5
1.2 算法	8
1.2.1 什么是算法	8
1.2.2 算法分析	9
1.2.3 算法设计	12
习题 1	15
实验 1 算法设计与分析	16
第 2 章 线性表	17
2.1 线性表抽象数据类型	17
2.2 线性表的顺序存储和实现	18
2.2.1 线性表的顺序存储结构	18
2.2.2 顺序表	18
2.2.3 排序顺序表	29
2.3 线性表的链式存储和实现	37
2.3.1 线性表的链式存储结构	37
2.3.2 单链表	38
2.3.3 双链表	47
2.4 线性表应用：多项式的表示及运算	51
2.4.1 一元多项式的表示及运算	52
2.4.2 二元多项式的表示及运算	56
习题 2	58
实验 2 线性表的基本操作	58
第 3 章 串	61
3.1 串抽象数据类型	61
3.2 串的存储和实现	62
3.2.1 串的存储结构	62
3.2.2 常量字符串类	63
3.2.3 变量字符串类	71
3.3 串的模式匹配	74
3.3.1 Brute-Force 算法	74

3.3.2	模式匹配应用	76
3.3.3	KMP 算法	79
习题 3		85
实验 3	串的基本操作及模式匹配算法	85
第 4 章	栈和队列	89
4.1	栈	89
4.1.1	栈抽象数据类型	89
4.1.2	顺序栈	90
4.1.3	链式栈	90
4.1.4	栈的应用	91
4.2	队列	97
4.2.1	队列抽象数据类型	97
4.2.2	顺序队列	98
4.2.3	链式队列	100
4.2.4	队列的应用	102
4.2.5	优先队列	103
4.3	递归	105
习题 4		110
实验 4	栈和队列以及递归算法	111
第 5 章	数组和广义表	113
5.1	数组	113
5.2	特殊矩阵的压缩存储	118
5.2.1	三角矩阵、对称矩阵和对角矩阵的压缩存储	118
5.2.2	稀疏矩阵的压缩存储	120
5.3	广义表	127
5.3.1	广义表抽象数据类型	127
5.3.2	广义表的存储结构	129
5.3.3	广义表双链表示的实现	131
习题 5		134
实验 5	特殊矩阵和广义表的存储和运算	135
第 6 章	树和二叉树	137
6.1	树及其抽象数据类型	137
6.1.1	树定义	137
6.1.2	树的术语	138
6.1.3	树抽象数据类型	139
6.2	二叉树	139
6.2.1	二叉树定义	139
6.2.2	二叉树性质	140
6.2.3	二叉树的遍历规则	141

6.2.4	二叉树抽象数据类型	142
6.2.5	二叉树的存储结构	142
6.2.6	二叉树的二叉链表实现	144
6.3	线索二叉树	158
6.3.1	线索二叉树定义	159
6.3.2	中序线索二叉树	160
6.4	Huffman 树	163
6.4.1	Huffman 编码	163
6.4.2	Huffman 树及其构造算法	164
6.5	树的表示和实现	171
6.5.1	树的遍历规则	171
6.5.2	树的存储结构	171
6.5.3	树的父母孩子兄弟链表实现	172
	习题 6	175
	实验 6 树和二叉树的基本操作	177
第 7 章	图	180
7.1	图及其抽象数据类型	180
7.1.1	图的基本概念	180
7.1.2	图抽象数据类型	183
7.2	图的表示和实现	184
7.2.1	图的邻接矩阵表示和实现	184
7.2.2	图的邻接表表示和实现	191
7.2.3	图的邻接多重表表示	196
7.3	图的遍历	197
7.3.1	图的深度优先搜索遍历	197
7.3.2	图的广度优先搜索遍历	200
7.4	最小生成树	201
7.4.1	生成树	201
7.4.2	最小生成树的构造算法	203
7.5	最短路径	206
7.5.1	单源最短路径	207
7.5.2	每对顶点间的最短路径	210
	习题 7	214
	实验 7 图的表示和操作	215
第 8 章	查找	216
8.1	查找的基本概念	216
8.2	二分法查找	218
8.3	基于索引表的分块查找	220
8.4	散列	224

8.4.1	散列表	224
8.4.2	散列映射	230
8.5	二叉排序树和平衡二叉树	233
8.5.1	二叉排序树	234
8.5.2	树映射	241
8.5.3	平衡二叉树	242
习题 8		245
实验 8	查找算法	247
第 9 章	排序	248
9.1	插入排序	248
9.1.1	直接插入排序	248
9.1.2	希尔排序	250
9.2	交换排序	251
9.2.1	冒泡排序	252
9.2.2	快速排序	253
9.3	选择排序	256
9.3.1	直接选择排序	256
9.3.2	堆排序	257
9.4	归并排序	260
9.5	线性表的排序算法	262
9.5.1	顺序表的排序算法	262
9.5.2	单链表的排序算法	263
9.5.3	循环双链表的排序算法	264
习题 9		267
实验 9	排序算法设计及分析	267
第 10 章	综合应用设计	269
10.1	JAVA 集合框架	269
10.1.1	Comparator 比较器接口	269
10.1.2	Arrays 数组类	269
10.1.3	集合	270
10.1.4	映射	276
10.2	实现迭代器	277
10.2.1	提供迭代器的类	277
10.2.2	基于迭代器的操作	280
10.3	算法设计策略	281
10.3.1	分治法	282
10.3.2	动态规划法	283
10.3.3	贪心法	285
10.3.4	回溯法	295

10.4 课程设计的目的、要求和选题..... 309

附录 A ASCII 字符与 Unicode 值..... 316

附录 B Java 关键字..... 317

附录 C Java 基本数据类型..... 319

附录 D Java 运算符及其优先级..... 320

附录 E Java 类库（部分）..... 321

 E.1 java.lang 语言包..... 321

 E.2 java.util 实用包..... 324

附录 F MyEclipse 常用菜单命令..... 326

参考文献..... 328

第 1 章

绪 论

计算机数据处理的前提是数据组织，如何有效地组织数据和处理数据是软件设计的基本内容，也是“数据结构”课程的基本内容。

作为绪论，本章勾勒“数据结构”课程的一个轮廓，说明“数据结构”课程的目的、任务和主要内容。本章主要介绍数据结构概念所包含的数据逻辑结构、数据存储结构和对数据的操作等；介绍抽象数据类型概念；介绍算法概念、算法设计目标、算法描述和算法分析方法。

1.1 数据结构的基本概念

1.1.1 为什么要学习数据结构

软件设计是计算机学科的核心内容之一。进行软件设计时要考虑的首要问题是数据的表示、组织和处理方法，这直接关系到软件的工程化程度和软件的运行效率。

随着计算机技术的飞速发展，计算机应用从早期的科学计算扩大到过程控制、管理和数据处理等各领域。计算机处理的对象也从简单的数值数据发展到各种多媒体数据。软件系统处理的数据量越来越大，数据的结构也越来越复杂。因此，针对实际应用问题，如何合理地组织数据，如何建立合适的数据结构，如何设计好的算法，是软件设计的重要问题，而这些正是“数据结构”课程讨论的主要内容。

在计算机中，现实世界中的对象用数据来描述。“数据结构”课程的任务是，讨论数据的各种逻辑结构、在计算机中的存储结构以及各种操作的算法设计。“数据结构”课程的主要目的是培养学生掌握处理数据和编写高效率软件的基本方法，为学习后续课程及进行软件开发打下坚实基础。

数据结构是软件设计的重要理论和实践基础，数据结构设计和算法设计是软件系统设计的基础和核心。“数据结构”课程讨论的知识内容，是软件设计的理论基础；“数据结构”课程介绍的技术方法，是软件设计中使用的的基本方法。“数据结构”是一门理论与实践并重的课程，既要掌握数据结构的基础理论知识，又要掌握运行和调试程序的基本技能。因此，“数据结构”课程在计算机学科本科培养过程中的地位十分重要，是计算机类各专业本科的核心课程，是培养程序设计能力必不可少的重要环节。

在计算机界流传的经典名言“数据结构+算法=程序设计”（瑞士 Niklaus Wirth 教授），这句话简洁、明了地说明了程序（或软件）与数据结构和算法的关系，以及“数据结构”课程的重要性。

1.1.2 什么是数据结构

数据（Data）是描述客观事物的数字、字符以及所有能输入到计算机中并能被计算机接受的各

种符号集合的统称。数据是信息的符号表示，是计算机程序的处理对象。除了数值数据外，计算机能够处理的数据还有字符串等非数值数据，以及图形、图像、音频、视频等多媒体数据。

表示一个事物的一组数据称作一个**数据元素**（Data Element），数据元素是数据的基本单位。一个数据元素可以是一个不可分割的原子项，也可以由多个数据项组成。**数据项**（Data Item）是数据元素中有独立含义的、不可分割的最小标识单位。例如，一个整数、一个字符都是原子项；一个学生数据元素由学号、姓名、性别和出生日期等多个数据项组成。一个数据元素中，能够识别该元素的一个或多个数据项称为**关键字**（Keyword），能够唯一识别数据元素的关键字称为主关键字（Primary Keyword）。

在由数据元素组成的数据集合中，数据元素之间通常具有某些内在联系。研究数据元素之间存在的关系并建立数学模型，是设计有效地组织数据和处理数据方案的前提。

数据的结构指数据元素之间存在的关系。一个**数据结构**（Data Structure）是由 n ($n \geq 0$) 个数据元素组成的有限集合，数据元素之间具有某种特定的关系。

数据结构概念包含三方面：数据的逻辑结构、数据的存储结构和对数据的操作。

1. 数据的逻辑结构

数据的逻辑结构是指数据元素之间的逻辑关系，用一个数据元素的集合和定义在此集合上的若干关系来表示，常被简称为数据结构。

根据数据元素之间逻辑关系的不同数学特性，数据结构分为三种：线性结构、树结构和图，其中树和图是非线性结构。以图示法表示数据的逻辑结构如图 1.1 所示，一个圆表示一个数据元素，圆中的字符表示数据元素的标记或取值，连线表示数据元素之间的关系。

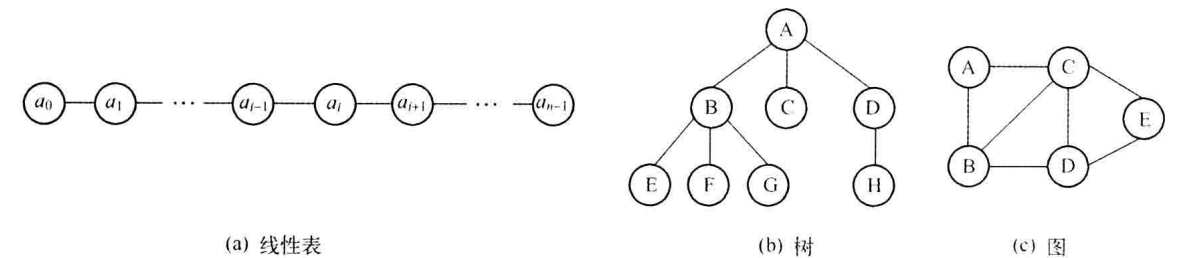


图 1.1 线性表、树和图的数据逻辑结构表示

(1) 线性结构

线性结构是数据元素之间具有线性关系的数据结构。线性表($a_0, a_1, \dots, a_{n-1}$)是由 n (≥ 0) 个类型相同的数据元素 $a_0, a_1, \dots, a_{n-1}$ 组成的有限序列，若 $n=0$ ，为空表；若 $n>0$ ， a_i ($0 \leq i \leq n-1$) 有且仅有一个前驱元素 a_{i-1} 和一个后继元素 a_{i+1} ， a_0 没有前驱元素， a_{n-1} 没有后继元素。采用序号（Index）确定数据元素在线性结构中的逻辑次序。

数据元素可以是一个数、字符、字符串或其他复杂形式的数据。例如，整数序列{1,2,3,4,5}，字母序列{'A','B','C',..., 'Z'}，表 1-1 所示的学生序列都是线性表，数据元素之间具有顺序关系。

表 1-1 学生信息表					
学 号	姓 名	年 龄	学 号	姓 名	年 龄
20120001	王红	18	20120003	吴宁	18
20120002	张明	19	20120004	秦凤	17

表 1-1 中，学生数据元素由“学号”、“姓名”、“年龄”等多个数据项组成，“姓名”可以作为标识一个学生的关键字；“学号”是能够唯一标识一个学生的主关键字。

(2) 树结构

树结构是数据元素之间具有层次关系的一种非线性结构，树中数据元素通常称为结点。树结构的层次关系是指，根（最顶层）结点没有前驱结点（称为父母结点），除根之外的其他结点有且仅有一个父母结点，所有结点可有零到多个后继结点（称为孩子结点）。在图 1.1(b)中，A 是树的根结点，B 结点有一个父母结点 A，有 3 个孩子结点 E、F 和 G。家谱、Windows 文件系统的组织方式、淘汰赛的赛果等都是树结构。淘汰赛的赛果是一棵满二叉树，如图 1.2 所示，其中数据是 2010 年南非世界杯足球赛淘汰赛的赛果。

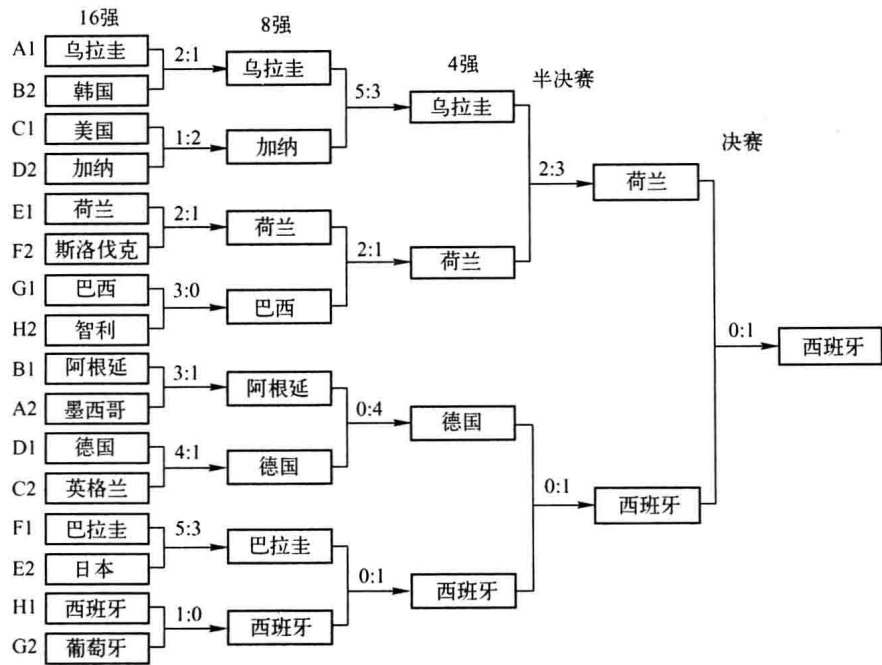


图 1.2 淘汰赛的赛果是一棵满二叉树

(3) 图

图也是非线性结构，每个数据元素可有多多个前驱元素和多个后继元素。例如，交通道路图、飞机航班路线图等都具有图结构。图 1.3 是从南京飞往昆明的航班路线图，有直飞航班，也有经停重庆或长沙的航班，边上的数值表示两地间的千米数。

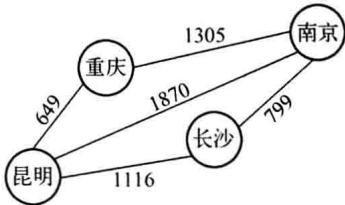


图 1.3 南京飞往昆明的航班路线图

2. 数据的存储结构

数据元素及其关系在计算机中的存储表示或实现称为数据的存储结构，也称为物理结构。软件系统不仅要存储所有数据，还要正确地表示出数据元素之间的逻辑关系。

数据的逻辑结构从逻辑关系角度观察数据，它与数据的存储无关，是独立于计算机的。而数据的存储结构是逻辑结构在计算机内存中的实现，它是依赖于计算机的。

数据存储结构的基本形式有两种：顺序存储结构和链式存储结构。

(1) 顺序存储结构

顺序存储结构使用一组连续的内存单元依次存放数据元素，数据元素在内存中的物理存储次序与它们的逻辑次序相同，即每个元素 a_i 与其前驱元素 a_{i-1} 及后继元素 a_{i+1} 的存储位置相邻。这样，数据元素的物理存储次序体现它们之间的逻辑关系。

通常，使用程序设计语言中的数组实现顺序存储结构。

(2) 链式存储结构

链式存储结构使用若干地址分散的存储单元存储数据元素，逻辑上相邻的数据元素在物理位置上不一定相邻，数据元素间的关系需要采用附加信息特别指定。通常，采用指针变量记载前驱或后继元素的存储地址，由数据域和地址域组成的一个结点表示一个数据元素，通过地址域将相互直接关联的结点链接起来，结点间的链接关系体现数据元素间的逻辑关系。

线性表可采用上述两种存储结构。线性表 $(a_0, a_1, \dots, a_{n-1})$ 的两种存储结构如图 1.4 所示。

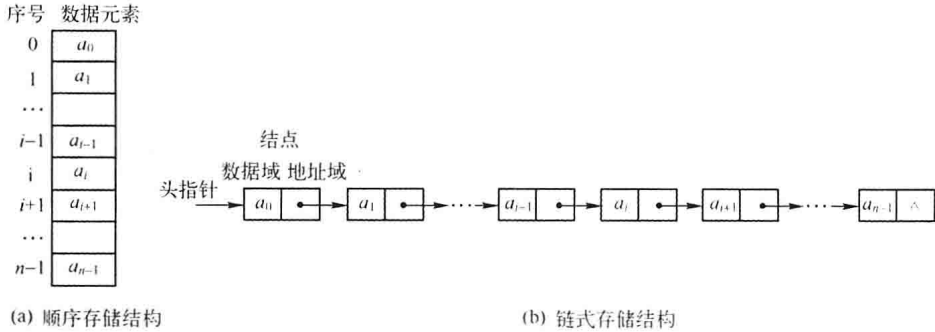


图 1.4 线性表 $(a_0, a_1, \dots, a_{n-1})$ 的两种存储结构

① 采用顺序存储结构存储线性表 $(a_0, a_1, \dots, a_{n-1})$ ，数据元素占用所有存储空间，各元素 a_i 连续存储，逻辑上相邻的数据元素 a_{i-1} 、 a_i 和 a_{i+1} 在存储位置上也相邻，数据的存储结构体现数据的逻辑结构。

② 采用链式存储结构存储线性表 $(a_0, a_1, \dots, a_{n-1})$ ，各元素 a_i 分散存储，每个元素 a_i 占用一个结点(Node)存储单元，结点结构如下：

结点(数据域, 地址域)

其中，数据域存储数据元素 a_i ，地址域存储元素 a_i 的前驱和(或)后继结点的地址。结点间的链接关系体现数据的逻辑结构。一条链式存储的线性表必须使用一个地址域(称为头指针)记住元素 a_0 的结点地址。

如果一个数据元素由多个数据项组成，则数据域有多个。例如，学生信息线性表的顺序和链式存储结构如图 1.5 所示。

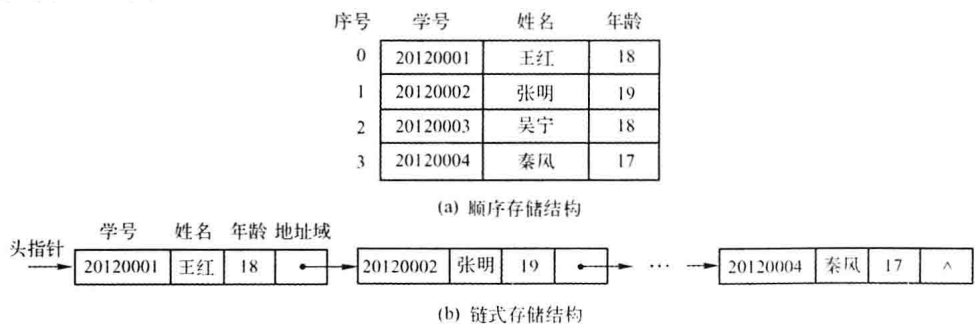


图 1.5 学生信息表的两种存储结构

顺序存储结构和链式存储结构是两种最基本、最常用的存储结构。除此之外，将顺序存储结构和链式存储结构进行组合，还可以构造出一些更复杂的存储结构。

3. 对数据的操作

每种数据结构都需要一组对其数据元素实现特定功能的操作(运算或处理)，包含以下一些基本操作，此外根据其自身特点，还需要一些特定操作。

- ① 初始化。
 - ② 判断是否空状态。
 - ③ 存取，指获得、设置指定元素值。
 - ④ 统计数据元素个数。
 - ⑤ 遍历 (Traverse)，指按照某种次序访问一个数据结构中的所有元素，并且每个数据元素只被访问一次。遍历一个数据结构，将得到一个所有数据元素的线性序列。
 - ⑥ 插入 (Insert)、删除 (Remove) 指定元素。
 - ⑦ 查找 (Search)，指在数据结构中寻找满足给定条件的数据元素。
 - ⑧ 排序 (Sort)，指将数据元素按照指定关键字值的大小递增 (或递减) 次序重新排列。
- 对数据的操作定义在数据的逻辑结构上，实现对数据的操作依赖于数据的存储结构。例如，线性表包含上述一组对数据的操作，采用顺序存储结构或链式存储结构，都可实现这些操作。

1.1.3 数据类型与抽象数据类型

1. 数据类型

类型 (Type) 是具有相同逻辑意义的一组值的集合。数据类型 (Data Type) 是指一个类型和定义在这个类型上的操作集合。数据类型定义了数据的性质、取值范围以及对数据所能进行的各种操作。例如，Java 语言的整数类型 `int`，除了数值集合 $[-2^{31}, \dots, -2, -1, 0, 1, 2, \dots, 2^{31} - 1]$ 之外，还包括在这个值集上的操作集合 $[+, -, *, /, \%, =, ==, !=, <, <=, >, >=]$ 。

程序中的每个数据都属于一种数据类型，确定数据的类型意味着确定了数据的性质以及对数据进行的运算和操作，同时数据也受到类型的保护，确保对数据不能进行非法操作。

高级程序设计语言通常预定义一些基本数据类型和构造数据类型。基本数据类型的值是单个的、不可分解的，它可直接参与该类型所允许的运算。构造数据类型是使用已有的基本数据类型和已定义的构造数据类型，按照一定的语法规则组织起来的较复杂的数据类型，构造数据类型的值由若干元素组合而成，这些元素按某种结构组织在一起。

Java 语言的基本数据类型有整数类型、浮点数类型、字符类型、布尔类型，构造数据类型 (称为引用类型) 有数组、类和接口。

数据类型与数据结构两个概念的侧重点不同。数据类型研究的是每种数据所具有的特性，以及对这种特性的数据能够进行哪些操作；数据结构研究的是数据元素之间具有的相互关系，数据结构与数据元素的数据类型无关，也不随数据元素值的变化而改变。

2. 抽象数据类型

抽象数据类型 (Abstract Data Type, ADT) 是指一个数学模型以及定义在该模型上的一组操作。例如，复数是数学中常用的一种类型，一个复数 $a+ib$ 由实部 a 和虚部 b 两部分组成， i 是虚部标记。复数抽象数据类型描述如下：

ADT Complex	//复数抽象数据类型
{	
double real, imag;	//复数的实部和虚部
Complex (double real, double imag)	//指定实部和虚部构造一个复数
Complex add(Complex c)	//加法，返回当前复数与 c 相加之后的复数
Complex sub(Complex c)	//减法，返回当前复数与 c 相减之后的复数
}	

大多数程序设计语言没有提供复数类型。程序员需要实现 ADT Complex 所声明的操作。

(1) 数据抽象

抽象数据类型和数据类型本质上是一个概念，它们都表现数据的抽象特性，**数据抽象**是指“定义和实现相分离”，即将一个类型上的数据及操作的逻辑含义与具体实现分离。程序设计语言提供的数据类型是抽象的，仅描述数据的特性和对数据操作的语法规则，并没有说明这些数据类型是如何实现的。程序设计语言实现了它预定义数据类型的各种操作。程序员按照语言规则使用数据类型，只考虑对数据执行什么操作（做什么），而不必考虑怎样实现这些操作（怎样做）。

例如，赋值语句的语法定义为“变量 = 表达式”，表示先求得指定表达式的值，再将该值赋给指定变量。程序员需要关注所用数据类型的值能够参加哪些运算、表达式是否合法、表达式类型与变量类型是否赋值相容等；至于如何存储一个整数、变量的存储地址是什么、如何求得表达式值等实现细节则不必关注，这些操作由语言的实现系统完成。

数据抽象是研究复杂对象的基本方法，也是一种信息隐蔽技术，从复杂对象中抽象出本质特征，忽略次要细节，使实现细节相对于使用者不可见。抽象层次越高，其软件复用程度也越高。抽象数据类型是实现软件模块化设计思想的重要手段。一个抽象数据类型是描述一种特定功能的基本模块，由各种基本模块可组织和构造起来一个大型软件系统。

(2) 声明抽象数据类型

与程序设计语言中使用数据类型描述数据特性原理相同，本课程使用抽象数据类型描述数据结构，后续章节将声明线性表、树、图等抽象数据类型，每种抽象数据类型描述一种数据结构的逻辑特性和操作集合，与其存储结构及实现无关。

只有实现了这些抽象数据类型，才能在实际应用中使用它们。实现抽象数据类型依赖于数据的存储结构。例如，线性表可分别采用顺序存储结构或链式存储结构实现，详见第 2 章。

声明抽象数据类型包括 ADT 名称定义、数据定义和操作集合，其中，数据定义描述数据元素的逻辑结构，操作集合描述该数据结构所能进行的各种操作声明，约定操作名、初始条件和操作结果等操作要求。

【例 1.1】集合的表示与实现。

(1) 集合抽象数据类型

数学中的集合，元素间没有次序、不可重复、不排序。声明集合抽象数据类型 Set<T>如下，为集合运算约定了共同的方法声明。

ADT Set<T>	//集合抽象数据类型
{	
数据: 集合中的数据元素，数据元素的数据类型为 T	
操作:	
boolean isEmpty()	//判断集合是否为空
int size()	//返回元素个数
T search(T key)	//返回查找到的关键字为 key 元素
boolean contains(T key)	//判断是否包含关键字为 key 元素
boolean add(T x)	//增加元素 x
T remove(T key)	//删除关键字为 key 元素，返回被删除元素
void clear()	//删除所有元素
String toString()	//返回集合所有元素的描述字符串
boolean equals(Object obj)	//比较 this 与 obj 引用集合是否相等
Object[] toArray()	//返回包含集合所有元素的数组