高等院校系列教材
河南科学技术出版社

21世纪高等院校软件技术专业规划教材

MIANXIANG DUIXIANG DE SHUJU JIEGOU

面向对象的数据结构

（Java版）

主编　车战斌　李占波

河南科学技术出版社

21 世纪高等院校软件技术专业规划教

面向对象的数据结构

(Java 版)

主　编　车战斌　李占波

河南科学技术出版社
·郑州·

图书在版编目(CIP)数据

面向对象的数据结构:Java 版/车战斌,李占波主编.
郑州:河南科学技术出版社,2008.4
(21 世纪高等院校软件技术专业规划教材)
ISBN 978-7-5349-3819-1

Ⅰ.面… Ⅱ.①车…②李… Ⅲ.Java 语言-程序设计-高等学校:技术学校-教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2008)第 013660 号

出版发行:河南科学技术出版社
地址:郑州市经五路 66 号 邮编:450002
电话:(0371)65737028 65788613
网址:www.hnstp.cn
策划编辑:范广红
责任编辑:杨艳霞
责任校对:张小玲 王晓红
封面设计:张 伟
版式设计:南 妮
印 刷:黄委会设计院印刷厂
经 销:全国新华书店
幅面尺寸:185 mm×260 mm 印张:16.5 字数:382 千字
版 次:2008 年 4 月第 1 版 2008 年 4 月第 1 次印刷
定 价:28.00 元

编审委员会名单

主　任　王世卿

副主任　车战斌　刘黎明　吴勇军　李　波
李占波　李　捷　张素智

委　员　（以姓氏笔画为序）
于立红　王世卿　车战斌　刘黎明
孙　杰　李　波　李　捷　李占波
吴勇军　张素智　陈桂生　秦国防
郭长庚

编写人员名单

BIANXIEREN YUAN MINGDAN

主　编　车战斌　李占波

编　委　（以姓氏笔画为序）

车战斌　邢静宇　任两品　李占波

李俊峰　高　亮　高艳霞

编写说明

BIANXIESHUOMING

近年来，我国国民经济快速发展，信息产业更是突飞猛进，再加上国际化软件技术专业人才的巨大市场需求，都为软件技术专业人才的培养提供了强大的驱动力。2001 年 12 月，教育部与国家计划经济委员会联合提出了建立 35 所国家示范性软件学院的决定，并于 2003 年又批准 35 所高职高专院校为示范性软件职业技术学院。除此之外，各省又相继成立了一批软件学院或软件职业技术学院。目前，这些高校已成为高等职业教育的一个重要组成部分，培养了大批优秀的软件技术专业人才，为我国软件产业的可持续健康发展提供了强有力的支撑。

教材的建设对人才培养起着至关重要的作用。就如何做好课程体系的建设和人才培养工作，中国计算机学会教育委员会、高等学校计算机教育研究会联合组建的“中国计算机科学与技术学科教程 2002 研究组”推出的《中国计算机科学与技术学科教程 2002》给出了 16 门核心课程描述，建议以此构建专业人才的公共平台；2006 年，高等学校计算机教育研究会结合高职院校计算机教育的特点，又推出了《中国高职院校计算机教育课程体系》蓝皮书，给出了目前流行的计算机类各专业的参考课程体系架构。目前，国内各出版机构围绕这两本书出版了很多软件类教材，但是多偏重于编程语言理论的教学，大多为传统的教学模式，结果导致学生的编程设计能力和应用能力不够。为此，河南科学技术出版社组织郑州大学、河南大学、中原工学院、郑州轻工业学院、南阳理工学院、平顶山学院等省内软件技术学院教学一线的教师及软件开发公司具有丰富经验的工程技术人员共同编写了一批教材，共包括理论课教材 20 种，实训课教材 13 种。

本套教材以《中国计算机科学与技术学科教程 2002》和《中国高职院校计算机教育课程体系》蓝皮书为

指导，以“就业为导向、能力为本位”要求为原则，以“淡化理论，强化能力，体现创新，灵活多用”为出发点，突出实际动手能力和实用性，突出案例和任务驱动等技能训练。为了培养外向型软件技术专业人才，还编写了《计算机专业英语》、《计算机专业日语》。

为满足不同学校、不同层次、不同基础水平进行教学安排和人才培养的实际需要，本套教材尽量采用富有弹性的模块化内容结构，对知识传授与能力培养采用有目的的整合、融合和综合的编写方法，将若干知识点组成模块，每个模块既是教材的有机组成部分，又是一个相对完整而开放的单元，以便于教师组织教学与学生自主学习。

同时，本套教材具有系列化、立体化特征，即在编写教材的同时，开发出一些好的电子课件，通过教学资源库、课程网站等供老师、学生使用。

本套教材既适合作软件学院、软件职业技术学院以及计算机相关专业的本、专科生教材，也可作为实训机构的培训教材和相关技术人员的学习参考书。

要编写一套推动和促进应用型人才培养的教材是一项艰巨的任务，加上软件技术专业的招生时间还比较短，可以借鉴的经验不多，尽管编审委员会与各位专家都已尽力，恐仍存在疏漏之处，恳请各位读者批评指正。

郑州大学软件技术学院　**王世卿**

2008 年 1 月

前言

数据结构是计算机专业的传统专业基础课之一，在计算机类专业中占有很重要的地位，是学生学习编程最重要的课程。传统数据结构课程的教学目的有三个方面，一是掌握和了解典型的数据结构，主要是三大类结构及其计算机存储方法；二是具体了解针对这些数据结构的操作算法；三是通过对数据结构及其算法的学习，积累编程经验，提高编程能力。

传统数据结构课程的基础是面向过程的程序设计，一般以C语言作为描述算法的工具。随着Java语言越来越普及，有很多作者对传统数据结构与算法课程进行了改写，使用Java语言描述过程算法，但是大部分仅仅停留在简单地把使用Java的方法替代C语言的函数。这样，对于真正使用Java编程序的人来说，其指导作用就显得太片面，没有将Java面向对象的编程思想充分反映出来。本教材与传统教材的重要不同，就是在前述三个教学目的基础上加入了另外两个教学目的，即掌握面向对象的数据结构的类设计和通过课程的学习提高学生面向对象的程序设计能力。

另外，针对软件技术专业，如何在学时有限的情况下，真正让学生掌握最基本的内容，如何裁减普通程序员使用相对较少的数据结构内容，都是目前教学中没有解决的问题。现在迫切需要一本将Java面向对象的编程思想融入教材，能真正帮助学生提高编程能力，并且其内容选取符合软件类专业的数据结构与算法教材。

本教材正是根据这种需要编写的。在以下几个方面作了一些尝试：

首先，教材编排上完全按数据结构类型编排，共分8章，第1章为绪论；第2章、第3章为线性表，因为线性表是本教材的重点，所以讲解得比较详细；第4章为栈；第5章为队列；第6章为矩阵；第7章为树；第8章为图。编

排的特点是:以线性结构为重点,也就是重点放在第2章到第6章。其次是树,对树的存储和各种操作也都作了比较详细的讲解。对图结构以了解为主,重点介绍了图的概念和基本操作。

其次,以面向对象的方法设计数据结构的存储和操作,并且以面向对象的方法举例说明数据结构的应用。通过典型例子的阅读,达到提高读者面向对象的编程和设计能力。

第三,采取分类习题和编程训练,循序渐进地提高学生的编程能力。本书的习题分为基础知识、模仿练习、动手设计三个部分,难度逐渐增加。

第四,个别章节加入了扩展阅读指导。主要是指导读者阅读Java类库中有关数据结构的内容。通过大量阅读程序提高学生的代码设计能力。

本教材的例题和数据结构的全部代码都经过调试,读者可到以下网址下载:www.hnstp.cn/download.asp。为提高读者调试程序的能力,编排时嵌入了一些语法错误和逻辑错误的例题,促使读者必须认真调试才能得到运行结果。

本教材由中原工学院的车战斌、高亮、高艳霞,郑州大学的李战波、任两品、李俊峰,南阳理工学院的邢静宇共同编写。车战斌、李占波担任主编,第1章和第2章由车战斌编写,第3章由高艳霞编写,第4章、第5章由邢静宇编写,第6章由高亮编写,第7章、第8章由李俊峰、任两品共同编写。最后由车战斌负责统稿。刘黎明、韩玉民等同志对本教材的编写工作提出了很多宝贵意见,在此表示感谢。

本教材试图把面向对象的程序设计思想与传统的数据结构课程内容进行结合,因为能力和水平有限,一定有很多错误和不当之处,请读者批评指正。

编者

2007年12月

第1章 绪 论

1.1 数据结构的概念

1.1.1 数据结构与分类

在学习软件技术的过程中，我们会提出这样的问题：什么是软件？什么是程序？什么是算法？实际上如果我们能够很好地回答这些问题，那我们距离一个真正的软件工程师可能就不远了。在很多书籍中对程序都有这样的定义：程序 = 算法 + 数据。算法就是为了达到某种计算目的而设计的计算步骤。数据的概念较简单，被算法处理的一些信息就是数据。可见，算法是为处理数据服务的，有什么样的数据就应该设计与之相适应的算法。实际问题中数据千变万化，特别是数据之间的关系非常复杂，实际问题对数据处理的要求也非常复杂，使得处理这些数据变得很困难。

那么，数据到底有没有什么规律可循？能否有一些有规律的数据处理方法？答案是肯定的。数据的复杂性表现在数据本身的复杂性和数据之间关系的复杂性。数据本身的复杂性主要表现在一个数据元素有很多成员，或者成员之间关系很复杂。如果把一个数据元素看成一个不可分割的数据，则数据关系就决定了处理这些数据的模式和方法。例如一个班级的成绩表，可以把每个学生及其成绩看成是一个数据元素，那么，这些成绩数据之间就是一个简单的线性表关系。这种数据之间不同的数据关系，我们称之为不同的数据结构，这就是本教材主要研究的内容。

研究数据及其处理方法，要解决几个问题：

第一，数据结构能否非常有规律地分成类？如何分类？即数据的逻辑结构分类问题。

第二，每一种逻辑结构需要哪些对数据的操作？即不同逻辑数据结构对数据的操作要求问题。

第三，不同类型的逻辑结构数据，如何在计算机中存储即数据的存储结构问题。

第四,某一种逻辑数据结构,在不同存储结构下的操作问题。

1.1.2 数据的逻辑结构和基本操作

人们经过多年的实践和总结,基本上已经解决了数据结构的分类问题。虽然实际问题中,数据结构非常复杂,多是复合型的数据结构,但是基本的数据结构可以分为三个大类,即线性数据结构、树状数据结构和图状数据结构。线性数据结构较容易理解,也是最常用的数据结构。树状结构也是实际问题中经常出现的结构,例如组织结构图就是典型的树状结构。图状数据结构是最普遍的数据结构也是最复杂的数据结构,例如,城市交通网络、人际关系网等都是典型的图状数据结构。当然,在这些大的分类之下,根据数据的特征还可以继续分类。例如,根据线性结构的用途不同,可以分为普通线性表、栈、队列、矩阵等,树还可以分为一般的树和二叉树,等等。

逻辑结构确定了,根据实际的用途,对数据的基本操作就基本上定下来了。例如,对于一个栈,基本上可以确定对其进行的操作主要有判断栈是否为空或为满、压栈、弹出栈等基本操作。

总之,在数据分类这个问题上已经基本形成了大家公认的结论,我们只需要了解这些结论就可以了。

1.1.3 数据的存储结构及其操作

数据的逻辑结构确定以后,在实际编程中最重要的就是数据的存储结构了。所谓存储结构,就是某种逻辑数据结构在计算机中的存储办法。有编程经验的人都知道,数据的存储结构设计的好坏,直接关系着后面编写程序的难易、效率的高低、程序的可维护性、程序的可读性。决定了数据的存储办法后,编写程序就是对这些数据结构进行操作。

在面向过程的程序设计中,数据的存储结构和作用于数据的操作算法是分离的,也就是本节开始时所说的,程序 = 数据 + 算法。这样的设计,不利于程序的封装,不利于程序的可读性和可维护性。在以面向过程为主的程序设计时代,与数据结构相关的课程都是以面向过程设计为基础,先设计存储结构然后再设计操作函数。

面向对象的编程技术把程序设计和现实生活中的实体紧密地结合在一起,使得程序和现实中的概念有了一对一的关系。具体到数据来说,就是把数据及其相关的操作联系在一起。这样使得设计出来的程序概念更清晰、可读性更强、更易维护。本教材就是立足于面向对象的思想阐述数据结构及其程序设计的。

1.2 算法与算法分析

上节中,我们提到决定了数据的存储结构后就可以设计对数据结构的操作算法。本教材的每一章中都有对数据结构操作的具体算法,特别是进行查找、遍历等操作时,算法显得更加重要。本节简单介绍算法的有关知识。

1.2.1 算法

1. *算法的定义* 算法(algorithm)就是一个有穷规则的集合。这是图灵奖获得者、著

名科学家D. Knuth对算法作的一个简单的定义。算法的实现,就是一组问题的操作序列。上述算法定义中的规则一般来说需要具有以下特征:

(1)有穷性。这是最关键的,算法必须在执行了有穷步以后结束。

(2)确定性。算法的每一步执行的操作必须是确定的。

(3)具有输入和输出。算法是为了解决问题的。所以,一般都有0个或多个输入,至少有一个输出。

(4)可行性。算法应该是在有限的和问题相适应的时间内可以完成的。例如,天气预报的计算,如果需要计算三天,那就是不可行的。

2. *算法的描述* 算法的描述有多种方法,高级语言本身就是算法的描述。在程序设计阶段,人们总是会使用伪代码、框图等办法描述算法。

3. *数据结构与算法的关系* 算法是建立在数据的存储结构基础上的。数据结构设计的合理,就可以使得算法变得简单明了并且高效。研究数据结构离不开研究算法,只有研究算法才能说明这种数据结构的意义。

1.2.2 算法设计

随着计算机解决问题的规模越来越大、复杂性越来越高,算法的设计显得更加重要。数据结构与算法的设计直接决定了软件最终的性能。什么是一个好的算法?根据问题的不同,评价可能也完全不同。一般来说,从下边几个方面评价算法的优劣:

(1)正确性。算法必须是正确的。

(2)高时间效率。即解决问题所花的时间要尽可能短。

(3)高空间效率。即解决问题所花费的存储资源要尽可能少。

(4)可读性、逻辑性。即算法要容易理解。这一点,主要是因为随着问题复杂性的提高,解决问题往往是多人合作,或者花很长时间,如果不降低问题的逻辑复杂性,就会导致系统无法完成。

(5)健壮性。算法在异常情况时应该有安全的可预料的反应。

这些要求,对于不同的问题规模,不同问题性质其重要性是不同的。计算机发展的过程中,人们对什么是一个好的算法的认识,逐渐发生了变化。在早期,计算机仅解决复杂性较小的重复计算问题的时候,高时间效率(也称时间复杂度)、高空间效率(也称空间复杂度)是第一位的。那时,CPU资源、内存资源是奇缺资源。例如20世纪80年代,一般计算机内存只有几MB,硬盘才有几十MB,CPU主频也都只有几十MB,这些资源的水平也使得人们不得不重视时间复杂度和空间复杂度。但是随着计算机解决问题复杂度的提高,同时由于CPU资源和存储资源的飞速发展,对于大规模的软件,特别是对时效性要求不高的软件,人们更加重视可读性和程序设计的逻辑清晰。

其实,面向对象的程序设计就是一个很好的例子。一般来说,为了可通用、可维护,面向对象的设计一般来说,都需要花费更多的存储资源,运行效率一般来说也都较低,但是,它的最大好处是逻辑清晰,分析设计容易,维护方便。

1.2.3 算法分析

算法分析包括时间复杂度分析和空间复杂度分析,本教材主要是时间复杂度。这里