高等院校计算机应用技术系列教材

数据结构与算法分析（Java版）

王世民 主编
朱建方 孔凡航 编著

- 赠送教师完整的电子教案
- 提供课后习题及参考答案

清华大学出版社

数据结构与算法分析
(Java 版)

王世民　主编
朱建方　孔凡航　编著

清华大学出版社
北　京

内 容 简 介

本书以 Java 语言为基础，讨论了数据结构的线性结构和非线性结构及其实现，全书以 Java 语言作为数据结构的算法描述。

本书概念表述严谨，逻辑推理严密。既可以作为计算机或信息类及相关专业的教材，也可供学习数据结构及其算法的 Java 语言程序设计者参考。

图书在版编目(CIP)数据

数据结构与算法分析(Java 版)/王世民主编；朱建方，孔凡航编著. —北京：清华大学出版社，2005.7
(高等院校计算机应用技术系列教材)
ISBN 7-302-11007-7
Ⅰ.数… Ⅱ.①王… ②朱… ③孔… Ⅲ. ①数据结构—高等学校—教材 ②算法分析—高等学校—教材 ③Java 语言—程序设计—高等学校—教材 Ⅳ.TP311.12

中国版本图书馆 CIP 数据核字(2005)第 049493 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
组稿编辑：胡伟卷
文稿编辑：刘金喜
封面设计：王　永
版式设计：康　博
印 刷 者：北京鑫丰华彩印有限公司
装 订 者：三河市李旗庄少明装订厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：15.25　**字数**：352 千字
版　　次：2005 年 7 月第 1 版　2005 年 7 月第1 次印刷
书　　号：ISBN 7-302-11007-7/TP·7293
印　　数：1～5000
定　　价：22.00 元

前　言

数据结构是计算机程序设计重要的理论技术基础，它不仅是计算机学科的核心课程，而且已经成为计算机相关专业必要的选修课。其要求是学会分析、研究计算机加工的数据结构的特性，初步掌握算法的时间和空间分析技术，并能够编写出结构清晰、正确易读的算法，达到培养数据抽象能力的目的。学习数据结构可以使读者碰到具体问题时，能够找到一个优化的存储结构和解决方法。本书利用目前流行的开发工具 Java 语言进行数据结构设计，包含了数据结构的全部内容，符合大学的教学大纲，既可以作为大学数据结构课程的教材，又可以为程序设计者学习数据结构提供帮助。

本书以数据结构为主线，是在 Java 语言的基础之上编写的，希望读者在阅读本书之前，最好具备 Java 语言基础。这样，在学习数据结构时，能够比较容易地建立正确的数据结构中的存储和逻辑概念。

本书共分 10 章，第 1 章综述了数据结构中的基本概念；第 2 章主要描述了线性结构的存储与实现；第 3 章描述了特殊的线性结构的存储及其实现；第 4 章着重描述了数组的存储及数组的运算；第 5 章描述了层次结构的各种运算；第 6 章描述了网状结构的存储及实现算法；第 7 章介绍了各种排序的方法及算法比较；第 8 章主要介绍了查找方法；第 9 章介绍了操作系统中涉及的动态存储管理的基本技术；第 10 章介绍了常用文件结构。本书的内容突出了抽象数据类型的概念，对每一种数据结构都给出了相应的抽象数据类型的规范说明和实现。

我们向使用本教材的教师免费提供本书的电子教案，其下载网址为 http://www.tupwk.com.cn/downpage/index.asp。需要本书习题参考答案的教师请发邮件至 cwkbook@tup.tsinghua.edu.cn，邮件的主题请设为“获取《数据结构与算法分析》参考答案”。

本书的第 1~4 章、第 9 和第 10 章由王世民编写，第 5 和第 6 章由朱建方编写，第 7 和第 8 章由孔凡航编写，疏漏之处敬请读者提出宝贵意见或建议。

作　者

目　　录

第1章　数据结构概论

信息是计算机科学的基础，信息必须以数据的方式，按照一定的规则存储在计算机的存储器中。对数据的操作方法如增加、插入、删除、查询等既要考虑数据的存储，又要考虑数据操作中数据的处理速度等各方面的因素。学习数据结构及其相关算法可以有效地完成数据处理并提高数据的处理效率。

本章的学习目标：

- 数据结构的研究内容；
- 数据结构中的基本概念、常用术语；
- 学习数据结构的意义；
- Java 语言描述；
- 对算法进行描述和分析的方法。

1.1　什么是数据结构

众所周知，计算机是一种信息处理装置，信息中的各个数据元素并不是孤立存在的，它们之间存在着一定的结构关系。如何表示这些结构关系，如何在计算机中存储数据和信息，采用什么样的方法和技巧加工与处理这些数据，都是数据结构课程需要努力解决的问题。

一般来说，使用计算机解决具体问题时，通常需要几个步骤：分析具体问题得到数学模型，设计解决数学模型的算法，编制程序并调试，最后得到最终答案。

要想得到数据模型，必须了解数据之间的关系，在数据结构中数据之间的关系主要有两种，它们分别是线性关系和非线性关系，其中非线性关系又可以分为树型关系和图关系。确定了数据之间的关系后，可以将数据存储在计算机内，所以说数据的逻辑结构和存储结构是密不可分的两个方面，在实现算法时，首先应解决数据的存储问题。

为了说明这些，可以通过以下的例子说明。

例 1：图 1-1 是一个班级若干名学生的登记表，使用计算机管理该表格，可以把所有学生的登记表看作一个文件，它由若干条记录组成，每个学生对应一条记录，每条记录占一行，每行中有若干列，包括学号、姓名、性别、年龄等属性，反应了每个学生的基本情况，每条记录按学号从小到大排列。

学号	姓名	性别	年龄	民族	……
001	刘晓东	男	18	汉	……
002	张立	男	19	汉	……
003	韩娟	女	18	汉	……
……	……	……	……	……	……

图 1-1　线性结构图

因为每个学生的属性相同，将每个学生的各种属性集合抽象为一个独立的数据单位，与每个数据单位相邻的前一个数据单位最多只能有一个(第一个数据单位没有)，与每个数据单位相邻的后一个数据单位最多只能有一个(最后一个数据单位没有)。这时可以将文件中的数据单位的集合称为数据集合，我们将这种数据单位之间的关系称为线性关系。也就是说，该数据文件的逻辑结构为线性结构，也称该文件为一个线性表。在这种结构中，计算机内可以采用多种方法存储，在存储数据时，只要能够体现出数据元素之间的线性关系即可。

对于这个文件，我们可以进行的操作有：如果某个学生离开学校，应从该文件中将此学生删除，对应的操作就是删除一个数据元素；如果新来一位同学，则相应的操作是在该线性表中插入一个数据元素；新年伊始，每位同学的年龄应增加一岁，对应的操作就是修改所有同学的年龄；如果不要求按学号从小到大排列，需要按姓名的字典顺序排列，对应的操作就是排序等。

所以数据之间既要考虑存储，又要考虑数据单位之间的关系，在确定了存储结构后，根据存储的结构再来确定相应操作的实现方法。

例 2：我们使用的微机中，每个逻辑盘都有一个根目录，根目录下可以建立若干个子目录，而每个子目录下又可以创建若干个子目录(如图 1-2 所示)，形成一个对应多个的关系称为层次关系(树型关系)。

如果将每个目录作为一个独立的数据单位，那么每个子目录只有一个父目录，但是它又允许有若干个子目录。我们可以将目录的集合看作是数据的集合，目录之间的关系看作是非线性关系中的层次模型。

在此例中，比较典型的是根目录不能删除，每个子目录可以删除，并且每个目录下可以存储若干文件。此时的存储像一棵倒树，根目录像大树的树根(只有一个)，子目录像大树的树枝(允许多个)，而每个目录下的文件又像树枝上的树叶，此种结构层次分明，所以又称层次结构。

对应此种结构的操作有：如果某个目录需要删除(根目录例外)，对应的操作就是删除该目录以及该目录下的文件，又称删除结点(子树)；如果在此结构下增加一个管理目录，对应的操作是在树结构中添加一个结点；查找某个子目录，我们一般的操作是从根目录开始，依次往下查找，对应的数据结构的操作就是查找等。

在此种结构中比较关键的是各种对应的操作一般从根目录开始，对应数据结构中的操作就是从根结点开始。

例 3：如果以每台计算机为结点，组成计算机网络，在计算机网络中，每台计算机之

间可以实现通信，此时每台计算机之间可以与多台计算机进行通信(如图 1-3 所示)。

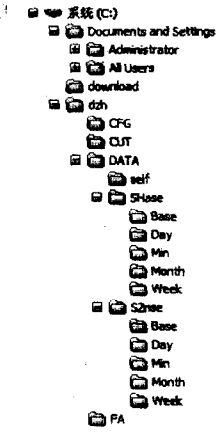


图 1-2　层次结构图

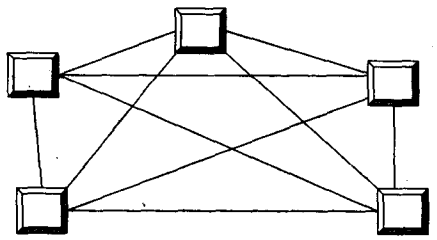


图 1-3　图结构

如果以每台计算机为结点，每台计算机可以通过传输介质与其他计算机进行通信，此时每台计算机之前可以有多台计算机，每台计算机之后也可以有多台计算机。简单将计算机之间的关系可以理解为多对多的关系，属于非线性关系中的图关系。

在此种结构中以每台计算机作为结点，如果它为其他计算机提供服务，则该计算机就是被提供服务的计算机的前趋，接受服务的计算机则称为该计算机后继；同时为其他计算机提供服务的计算机可能又接受多台计算机为它提供的服务，所以说每台计算机既是多台计算机的前趋，又是多台计算机的后继。

如果将此结构存储后，所对应的操作有：如果在网络中增加一台计算机，对应的操作是添加一个结点；如果在此结构中删除一个计算机用户，对应的操作是删除一个结点；在网络中搜寻一台计算机，对应的操作是查询等。

将以上的 3 个例子简单总结，可以说在处理数学模型时，首先应该保存数学模型所需要的必须的数据，根据数据在计算机中的存储，以及各个独立的数据元素之间的关系，可以将数据结构分为线性结构和非线性结构。

在客观世界中，如何体现数据之间的关系，需要计算机专业人员对数据进行分析时分类总结，以便更加深入地了解各种结构的特性以及关系。从不同的角度对数据结构进行分类，最终的目的就是更好地认识各种结构的特性及关系，而在设计某个操作时，首先应该清楚数据元素之间具有的逻辑关系，它适合什么样的存储结构来具体实现这种操作。同时同一种操作在不同的存储结构中实现的方法可以不同，而有的实现则完全依赖于所采用的存储结构，所以研究数据之间的关系直接影响了问题的求解，由此可见数据结构的重要性。

简单地说，数据结构是研究数据的存储、数据之间的关系及对数据实现各种操作的一门学科。

数据结构的定义可以记作：

Data-Structure=(D,R)

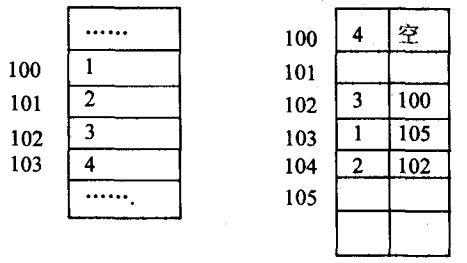
其中，D 是数据元素的有限集合，R 是 D 上的关系。

一般情况下，“关系”是指数据元素之间存在的逻辑关系，也称为数据的逻辑结构。数据在计算机内的存储表示(或映像)称为数据的存储结构或物理结构。

逻辑结构体现的是数据元素之间的逻辑关系，换句话说就是从操作对象中抽象出来的数学模型，因此又称为抽象结构，通常我们习惯说的数据结构一般就是指的逻辑结构。然而讨论数据结构的目的是为了在计算机中实现对数据的操作，因此还需要研究数据的存储结构。

存储结构是数据在计算机内的表示(映像)，又称物理结构。它包括数据元素的表示和关系的表示。由于映像的方法不同，所以同一种逻辑结构可以映像成两种不同的存储结构：顺序映像(顺序存储结构)和非顺序映像(非顺序存储结构)。顺序映像的特点是在顺序存储结构(一般用一维数组)中体现数据之间的关系；而非顺序存储结构则一般采用指针实现数据之间的关系，包括链式存储结构(链表)和散列结构等。数据的存储结构要能够正确反映数据元素之间的逻辑关系。也就是说，数据的逻辑结构和数据的存储结构是密不可分的两个方面，任何一种算法的设计取决于选定的逻辑结构，而算法的实现则依赖于采用的存储结构。

顺序映像(顺序存储结构)是借助元素在存储器中位置表示数据元素之间的逻辑关系，或逻辑上相邻的结点存储在物理位置上相邻的存储单元里，结点的逻辑关系由存储单元的邻接关系来体现；而非顺序映像(链式存储结构)是借助元素存储地址的指针表示元素之间的逻辑关系，或逻辑上相邻的结点在物理位置上可相邻，可不相邻，逻辑关系由附加的指针段表示。例如图 1-4 所示。



(a) 顺序映像示意图　　(b) 非顺序映像示意图

图 1-4　顺序存储和非顺序存储逻辑图

数据的非顺序存储结构除包括链式存储结构(简称链表)以外，还有散列存储结构、索引存储结构等。

链式存储结构利用指针直接表示数据元素之间的关系。

散列结构的基本思想是根据结点的关键字，利用散列函数直接计算出该结点的存储地址。

索引存储结构是指在存储结点信息的同时，还建立附加的索引表。索引表的每一项称为索引项，索引项的一般形式是：(关键字，地址)。关键字是能够惟一标识一个结点的那些数据项集合。索引存储结构分为稠密索引和稀疏索引，其中稠密索引是指每个结点在索引表中都有一个索引项的索引表；而稀疏索引是指一组结点在索引表中对应一个索引项的索引表。

针对存储结构，数据元素存储在计算机中，应对每个数据元素确定其取值范围以及在值上定义的一组操作就是数据类型。数据类型是和数据结构密切相关的一个概念，用以刻画(程序)操作对象的特征。在用高级语言编写的程序中，每个变量、常量或表达式都有一个它所属的数据类型，类型明显或隐含地规定了在程序执行期间变量或表达式所有可能的取值范围，以及在这些值上允许执行的操作。因此数据类型是指一个值的集合以及在这些值上定义的一组操作的总称。例如 Java 语言中有整数类型、字符类型、逻辑类型等。

数据类型根据是否允许分解分为原子类型和结构类型，其中原子类型是指其值不可再分的数据类型，例如整型、字符型等；而结构类型是指其值可以再分解为若干成分(分量)的数据类型，例如数组的值由若干分量组成。

实际上计算机中数据类型的概念不仅局限在高级语言中，从计算机和程序设计用户的角度来说，“位”、“字节”、“字”是计算机硬件的原子类型，程序设计用户中使用的高级语言提供的数据类型需要编译或解释转化为更低级语言(汇编或机器语言)的数据类型来实现。这样对使用者来说，不必了解数据类型在计算机内的表示，也不必了解数据类型的操作实现，实现了信息的隐蔽，方便了用户使用。

根据数据的结构(逻辑结构和存储结构)特性在数据的生存期间的变动情况，可将数据结构分为静态结构和动态结构。静态结构是指在数据存在期不发生任何变动，例如高级语言中的静态数组；而动态结构是指在一定范围内结构的大小可以发生变动，如使用的堆栈。

总之，数据结构所要研究的主要内容简单归纳为以下 3 个方面：

- 研究数据元素之间的客观联系(逻辑结构)；
- 研究数据在计算机内部的存储方法(存储结构)；
- 研究如何在数据的各种结构(逻辑的和物理的)上实施有效的操作或处理(算法)。

所以数据结构是一门抽象地研究数据之间的关系的学科。

1.2　数据结构的发展史及其在计算机科学中的地位

数据结构作为一门独立的课程始于 1968 年，在我国，数据结构作为一门独立课程是在 20 世纪 80 年代初，早期的数据结构对课程的范围没有明确的规定，数据结构的内容几乎和图论、树的理论是相同的，在 20 世纪 60~70 年代，随着大型程序的出现，软件也相对独立，结构程序设计逐步成为程序设计方法学的主要内容，人们已经认识到程序设计的实质就是对所确定的问题选择一种好的结构，从而设计一种好的算法。

目前在我国的高等教育体系中，数据结构不仅作为计算机专业教学计划中的重点核心课程之一，而且也开始成为许多非计算机专业的主要选修课程。在计算机专业中它是在程序设计之后的各专业课程之前的一门最重要的专业基础课，它为从事各类系统软件和应用软件的设计提供了必备的知识和方法。

数据结构在计算机科学领域中有着十分重要的地位，它有自己的理论、研究对象和应

用范围，它的内容也随着计算机技术的飞速发展而不断地扩充，从包括网络、集合代数论等离散结构的内容到包括数据库的内容等。

数据结构课程一般的前序课程是离散数学和程序设计基础，后序课程有操作系统、编译原理、数据库原理等。数据结构的研究不仅涉及到计算机硬件(特别是编码理论、存储装置和存取方法等)的研究范围，而且和计算机软件有着更密切的关系，无论是编译原理还是操作系统，都涉及数据元素在存储器中的分配问题。即使在研究信息检索时也必须考虑如何组织数据，以便查找和存取数据元素更为方便。因此数据结构是介于数学、计算机硬件和计算机软件之间的一门核心课程(如图 1-5 所示)。

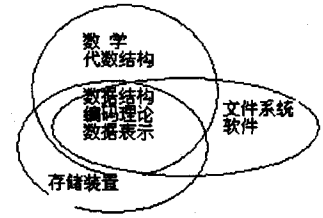


图 1-5　数据结构所处的位置

值得注意的是，数据结构的发展并未终结，一方面，面向专门领域中的特殊问题的数据结构得到研究和发展，如多维图形数据结构等；另一方面，从抽象数据类型观念讨论数据结构，已越来越被人们所重视。

1.3　基本概念和术语

数据在计算机科学中是指输入到计算机中并能够被计算机识别、存储和加工处理的符号的总称，例如一个整数或一个实数。对计算机科学而言，数据的定义非常广泛，计算机所能接受的所有符号，如我们说话，写的字，日常生活中看到的图像等，都可以通过编码的方式存储在计算机中，所以它们统称为数据结构中的数据。

数据元素是数据的基本单位，通常把数据元素作为一个整体进行考虑和处理。例如文件中的记录可以看作是一个整体。数据元素通常可以理解为逻辑意义上的数据的基本单位，而不是物理意义上的基本数据单位。

数据元素可由一个或多个数据项组成。也就是说，数据的基本物理单位是数据项。数据项(或数据元素)是指具有独立含义的最小识别单位(数据中不可分割的最小单位)。例如文件记录中的某一列，所以数据项又称项或字段。

数据项有逻辑形式(logical form)和物理形式(physical form)两个方面。用 ADT 给出的数据项的定义是它的逻辑形式，数据结构中对数据项的实现是它的物理形式。

结构通常是指关系，所以数据结构是指数据之间的关系的一门学科，它表示的是数据之间的相互形式，即数据的组织形式。数据结构可以从不同的角度进行分类，一般从用户的角度和计算机的角度来划分数据结构的分类：从用户的角度来说，主要关心的是数据元素之间的关系，所以数据结构称为逻辑结构(或抽象结构)，其作用是研究数据元素之间的逻辑(或抽象)关系；而从计算机的角度来说，主要关心的是数据元素及其关系在计算机内如何存储，称为存储结构(或物理结构)，其作用是数据元素及其关系在计算机内的表示。

如果从数据结构是否变化来分类，可以把数据结构划分为静态结构和动态结构。所谓静态结构是指数据结构在存在期间不会发生变动，如静态数组，不会随着操作的进行而改变数组的大小。而动态结构是指在一定范围内结构的大小要发生变动，如堆栈、队列以及树型结构等都属于动态结构的范畴。

根据数据结构是研究数据及其关系的一门学科的定义，了解数据是对客观事物的符号表示是非常必要的。数据在计算机中存储必须按照数据的分类存储，数据分类应该用数据类型划分，数据类型用以刻画(程序)操作对象的特性，在高级语言编写的程序中，每个变量或表达式都有一个它所属的确定的数据类型，类型明显或隐含地规定了在程序执行期间变量或表达式所有可能的取值范围，以及在这些值上允许进行的操作。因此数据类型是一个值的集合和定义在这个值集合上的一组操作的总称。如程序语言中的整型变量、字符型变量，都确定了数据的取值范围。

按照值的不同属性，数据类型可以分为两类：原子类型和结构类型。原子类型的值是不可分解的，如高级语言中基本数据类型(整型、字符型等)；结构类型是由若干成分按某种结构组成的，是可以分解的。

实际上，在计算机中，数据类型的概念并非局限在高级语言中，计算机的硬件和软件系统都提供了一组原子类型或结构类型，例如“位”、“字节”、“字”等属于原子类型，它们的操作通过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则是通过编译或解释器转化为低层(汇编语言或机器语言)的数据类型来实现。

在存储结构中包括了数据元素的表示和数据元素之间的关系表示。数据元素存储在计算机中，计算机中表示信息的最小单位是二进制数的一位，称为位(bit)，由若干位组成一个位串表示一个数据元素，称为元素或结点，当数据元素由若干个数据项组成时，对应于各个数据项的子位串称为数据域。在非顺序映像中借助指示元素存储地址，描述数据元素之间逻辑关系的部分称为指针域。结点也可以描述为数据处理的数据单位，它可能是一条记录、一个数据项或组合数据项。对应结点定义根据结点所处位置的不同可以将表中的结点分为前趋和后继结点，对表中任意结点，处于该结点之前的所有结点称为该结点的前趋结点，而处于该结点之后的所有结点称为该结点的后继结点，与之相邻的前趋结点称为直接前趋结点，而与之相邻的后继结点称为直接后继结点。表中的第一个结点称为开始结点，表中最后一个没有后继的结点称为终端结点。

1.4　抽象数据类型和数据结构

数据类型是指一个值的集合以及在这些值上定义的一组操作的总称。抽象数据类型(Abstract Data Type，ADT)是指抽象数据组织和与之相关的操作。每一个操作由它的输入和输出定义。抽象数据类型的定义取决于它的一组逻辑特性，而与其在计算机内的表示和实现无关，也就是说，无论其内部结构如何变化，只要其数学特性不变，都不影响其外部

的使用。或者说一个 ADT 的定义不涉及它的实现细节，这些实现细节对 ADT 的用户是隐藏的。隐藏实现细节的过程称为封装。数据结构是 ADT 的物理实现。

抽象数据类型和数据类型实质上是一个概念。例如计算机中的“整数”类型是一个抽象类型，尽管它们在不同的处理器上实现的方法不同，但由于其定义的数学特性相同，在用户看来都是相同的。因此抽象的意义在于数据类型的数学抽象特性。

抽象数据类型的定义仅取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关，即不论其内部结构如何变化，只要其数学特性不变，都不影响其外部的使用。

另一方面，抽象数据类型比数据类型的范畴更广，它不仅局限在处理器中已经定义并实现的数据类型，还包括用户在设计软件时自己定义的数据类型。一个软件系统的框架应建立在数据之上，而不应该建立在操作之上。所以定义的数据类型的抽象的层次越高，含有该抽象数据类型的软件模块的复用程度也就越高。

抽象数据类型可以定义为：

(D,S,P)

其中，D 表示数据对象，S 是 D 上的关系集，P 是对 D 的基本操作集。

ADT 使用伪码描述为：

```
ADT 抽象数据类型名{
    数据对象:<数据对象的定义>
    数据关系:<数据关系的定义>
    基本操作:<基本操作的定义>
}ADT 抽象数据类型名
```

抽象数据类型的定义由一个值域和定义在该值域上的一组操作组成，如果按照其值的不同特性，可细分为 3 种类型。

- 原子类型：属于原子类型的变量的值是不可再分的。
- 固定聚合类型：属于该类型的变量的值由确定数目的成分按照某种结构组成。
- 可变聚合类型：属于该类型的变量的值的成分数目不确定，其中序列的长度是可变的。

后两种类型可统称为结构类型。

例 1：整数的数学概念和施加到整数的运算构成一个 ADT。Java 的变量类型 int 就是对这个抽象类型的物理实现。但 int 型变量有一定的取值范围，所以对这个抽象的整型的实现并不完全正确，如果无法接受该限制，就必须引进一些其他的实现方法来实现这个抽象类型。

例 2：一个整数线性表的 ADT 应包含下列操作。

- 数据的存储结构确定线性关系。
- 把一个新整数插入到表尾。
- 按线性表的元素顺序依次打印。