

<<数据结构>>

图书基本信息

书名：<<数据结构>>

13位ISBN编号：9787040141498

10位ISBN编号：7040141493

出版时间：2004-2

出版时间：朱战立 高等教育出版社 (2004-02出版)

作者：朱战立 著

页数：291

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;数据结构&gt;&gt;

## 前言

数据结构是计算机专业以及其他一些和计算机技术关系密切的专业的一门核心必修课程。

数据结构课程的任务是讨论现实世界中数据的各种逻辑结构、在计算机中的存储结构以及实现各种操作的算法设计问题。

数据结构课程的目的是使学生掌握组织数据、存储数据和处理数据的基本方法，从而为以后进行软件开发和应用、进一步学习后续专业课程打下坚实的基础。

计算机软件开发方法是不断发展的，数据结构课程内容也应随着软件开发方法的不断发展而发展。

目前面向对象的软件分析和设计技术已发展成为软件开发的主流方法，因此，用面向对象的方法描述数据结构就成为数据结构课程改革的必然。

用面向对象的观点描述具体的数据结构问题时，首先涉及选用什么样的面向对象的高级语言问题。

C++语言是目前最普遍使用的面向对象程序设计的最底层语言，因此本书采用C++语言作为数据结构的描述语言。

数据结构课程是一门理论和实践结合密切的课程。

数据结构教材的编写方法有两种：一种是注重理论基础，另一种是注重实践应用。

考虑到高职高专和成人教育等特点，本书采用理论叙述简明、应用实例丰富的方法编写。

数据结构主要分为线性结构、树结构和图结构，本书讨论的内容主要包括：线性表、堆栈、队列、串、数组、树、二叉树、图、排序、查找以及递归。

其中，线性表、堆栈、队列、串和数组属于线性结构，树和二叉树属于树结构，图属于图结构，排序和查找是线性结构问题中两个应用最广泛的算法设计问题。

树结构和图结构属于非线性结构。

很多非线性结构的算法设计问题需要使用递归方法，因此，本书专设一章讨论递归的概念和递归算法的设计方法。

## &lt;&lt;数据结构&gt;&gt;

## 内容概要

《数据结构（C++语言描述）》为普通高等教育“十五”国家级规划教材。全书系统地介绍了各种类型的数据结构和查找、排序的各种方法。对于每一种类型的数据结构，都详细阐述了基本概念、各种不同的存储结构和不同存储结构上一些主要操作的实现算法，并给出了许多设计实例，以帮助读者理解。另外，书中还介绍了递归算法的设计方法。全书采用C++语言作为算法描述语言。为方便学习，附录中还给出了部分典型习题解答。

《数据结构（C++语言描述）》既可作为高等学校应用型本科计算机相关专业、成人及高职高专计算机相关专业的教材，也可作为从事计算机应用的工程技术人员的自学参考书。

## &lt;&lt;数据结构&gt;&gt;

## 书籍目录

第0章 C++程序设计基础0.1 程序的结构0.2 函数0.2.1 函数参数0.2.2 函数的返回值0.2.3 重载0.3 类0.3.1 访问权限0.3.2 构造函数和析构函数0.3.3 运算符重载0.3.4 友元0.3.5 分符号0.3.6 内联函数0.3.7 默认值0.3.8 派生类和继承性0.3.9 多态性和虚函数0.3.10 纯虚函数和抽象类0.3.11 结构体0.3.12 对象0.4 通用化的软件设计0.4.1 抽象数据类型0.4.2 模板0.5 动态申请和动态释放内存习题第1章 绪论1.1 数据结构的基本概念1.2 抽象数据类型和软件构造方法1.3 算法及其时间复杂度1.3.1 算法1.3.2 算法设计目标1.3.3 算法时间效率的度量1.4 算法书写规范习题第2章 线性表2.1 线性表抽象数据类型2.1.1 线性表的定义2.1.2 线性表抽象数据类型2.2 顺序表类2.2.1 顺序表的存储结构2.2.2 顺序表类定义2.2.3 顺序表类实现2.2.4 顺序表类方法的效率分析2.2.5 顺序表类应用举例2.3 单链表类2.3.1 单链表的结构2.3.2 单链表的动态内存分配方法2.3.3 结点类的定义和实现2.3.4 单链表类的定义和实现2.3.5 单链表操作的效率分析2.3.6 单链表应用举例2.4 循环单链表2.5 双向链表2.6 静态链表2.7 设计举例2.7.1 顺序表设计举例2.7.2 单链表设计举例习题第3章 堆栈和队列3.1 堆栈3.1.1 堆栈的基本概念3.1.2 堆栈抽象数据类型3.1.3 顺序堆栈类3.1.4 链式堆栈类3.2 堆栈应用3.2.1 括号匹配问题3.2.2 表达式计算问题3.3 队列3.3.1 队列的基本概念3.3.2 队列抽象数据类型3.3.3 顺序队列3.3.4 顺序循环队列类3.3.5 链式队列类3.3.6 队列的应用3.4 优先级队列3.4.1 顺序优先级队列类3.4.2 优先级队列的应用习题第4章 串4.1 串的基本概念、抽象数据类型和c++语言的串函数4.1.1 串的基本概念4.1.2 串的抽象数据类型4.1.3 C++语言的串函数4.2 串的存储结构4.2.1 串的顺序存储结构4.2.2 串的链式存储结构4.3 顺序串类4.3.1 顺序串类的定义4.3.2 构造函数和析构函数4.3.3 插入、删除和取子串成员函数4.3.4 常用操作符重载4.3.5 逻辑操作符重载4.3.6 顺序串类的测试4.4 串的模式匹配算法4.4.1 Brute—Force算法4.4.2 KMP算法4.4.3 Brute—Force算法和KMP算法的运行效率比较习题第5章 数组5.1 数组的基本概念5.1.1 数组的定义5.1.2 数组的实现机制5.1.3 数组抽象数据类型5.2 动态数组类5.3 特殊矩阵5.3.1 特殊矩阵的压缩存储5.3.2 n阶对称矩阵顺序表类5.4 稀疏矩阵5.4.1 稀疏矩阵的压缩存储5.4.2 三元组顺序表类5.4.3 三元组链表习题第6章 递归算法6.1 递归的概念6.2 递归算法的执行过程6.3 递归算法的设计方法6.4 递归过程和运行时栈6.5 递归算法的效率分析6.6 递归算法到非递归算法的转换6.7 设计举例6.7.1 一般递归函数设计举例6.7.2 回溯法及其设计举例习题第7章 树和二叉树7.1 树7.1.1 树的定义7.1.2 树的表示方法7.1.3 树的抽象数据类型7.1.4 树的存储结构7.2 二叉树7.2.1 二叉树的定义7.2.2 二叉树抽象数据类型7.2.3 二叉树的性质7.2.4 二叉树的存储结构7.3 以结点类为基础的二叉树设计7.3.1 二叉树的结点类7.3.2 二叉树的遍历7.3.3 二叉树遍历的应用7.3.4 应用举例7.3.5 非递归的二叉树遍历算法7.4 二叉树类7.5 二叉树的分步遍历7.5.1 二叉树遍历游标类7.5.2 二叉树中序遍历游标类7.5.3 二叉树层序遍历游标类7.6 线索二叉树7.7 哈夫曼树7.7.1 哈夫曼树的基本概念7.7.2 哈夫曼编码问题7.7.3 哈夫曼编码的软件设计7.8 树与二叉树的转换7.9 树的遍历习题第8章 图8.1 图的基本概念和抽象数据类型8.1.1 图的基本概念8.1.2 图的抽象数据类型8.2 图的存储结构8.2.1 图的邻接矩阵存储结构8.2.2 图的邻接表存储结构8.3 邻接矩阵图类8.4 图的遍历8.4.1 图的深度和广度优先遍历算法8.4.2 图的深度和广度优先遍历函数实现8.5 最小生成树8.5.1 最小生成树的基本概念8.5.2 普里姆算法8.5.3 克鲁斯卡尔算法8.6 最短路径8.6.1 最短路径的基本概念8.6.2 从一个顶点到其余各顶点的最短路径8.6.3 每对顶点之间的最短路径习题第9章 排序9.1 排序的基本概念9.2 插入排序9.2.1 直接插入排序9.2.2 希尔排序9.3 选择排序9.3.1 直接选择排序9.3.2 堆排序9.4 交换排序9.4.1 冒泡排序9.4.2 快速排序9.5 归并排序9.6 基数排序9.7 性能比较习题第10章 查找10.1 查找的基本概念10.2 静态查找表10.2.1 顺序表10.2.2 有序顺序表10.2.3 索引顺序表10.3 动态查找表10.3.1 二叉排序树10.3.2 B—树10.4 哈希表10.4.1 哈希表的基本概念10.4.2 哈希函数构造方法10.4.3 哈希冲突解决方法10.4.4 哈希表类习题附录部分典型习题解答参考文献

## &lt;&lt;数据结构&gt;&gt;

## 章节摘录

插图：0.3.2构造函数和析构函数构造函数是当定义对象时被自动调用的特殊成员函数，构造函数是用来在内存中建立类的具体对象（即在内存中为该对象分配适当的空间）并对其进行初始化赋值的成员函数。

构造函数的名字必须和类的名字相同，构造函数没有返回值。

构造函数允许有默认值，当构造函数有默认值时，若定义该类的对象时没有给出初始化值，按默认值处理。

例0—4中对象h有初始化值（plus, 3, 30），对象g和hg没有初始化值，因此对象g和hg的初始化值为默认值（plus, 0, 0）。

一个类允许有多个构造函数，当一个类有多个构造函数时，系统将根据定义对象时的参数类型和参数个数选择恰当的构造函数来建立对象和对该对象进行初始化。

析构函数是当对象被撤销时被自动调用的特殊成员函数。

当一个对象被撤销时，析构函数提供了释放该类对象占用内存空间的方法。

析构函数的名字是在类的名字前面加上前缀~。

析构函数没有返回值。

如果对象的内存空间是用非动态方法建立的，该类对象被撤销时系统能自动识别出这些对象占用的内存空间，此种情况下析构函数的函数体为空。

当析构函数的函数体为空时，C++允许省略析构函数。

如果对象的内存空间是用动态方法建立的，该类对象被撤销时系统不能识别出这些对象占用的内存空间，需要通过系统自动调用析构函数来释放此类对象占用的内存空间，此种情况下析构函数的函数体不能为空。

每个类只能有一个析构函数。

## <<数据结构>>

### 编辑推荐

《数据结构(C++语言描述)》为普通高等教育十五国家级规划教材之一。

<<数据结构>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>