

<<数据结构与算法实验教程>>

图书基本信息

书名：<<数据结构与算法实验教程>>

13位ISBN编号：9787040302141

10位ISBN编号：7040302144

出版时间：2011-1

出版时间：高等教育出版社

作者：张铭 等

页数：393

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>

<<数据结构与算法实验教程>>

内容概要

“数据结构与算法”课程是一门理论和实践紧密结合的课程，上机实习题的设计、实习训练的数量和质量都非常重要。

与作者编著的《数据结构与算法》主教材相呼应，张铭等编著的《数据结构与算法实验教程》主要介绍程序设计实践和技巧，常用的问题建模方法、基本算法，以及数据结构与算法应用，目的在于指导学生灵活地运用数据结构和算法解决具体问题，以便更好地巩固“数据结构与算法”课程所学的内容。

《数据结构与算法实验教程》给出了大量的数据结构与算法实习题，可用作相关知识点的实验设计题目。

尤其是第8章给出了10道综合性较强的中小型应用题，适合于布置期末综合实习，或者作为综合的课程设计作业项目。

《数据结构与算法实验教程》概念清楚，逻辑性强，内容新颖，可作为普通高校计算机科学技术专业与计算机应用专业学生的教材和参考书，还可供相关的信息技术领域工程技术人员参考。

<<数据结构与算法实验教程>>

书籍目录

第1章 “数据结构与算法”教学实施方案1.1 “数据结构与算法”的理论体系1.1.1 课程的基本定位1.1.2 知识体系1.2 “数据结构与算法”学习重点1.2.1 概论1.2.2 线性表1.2.3 栈与队列1.2.4 字符串1.2.5 二叉树1.2.6 树1.2.7 图1.2.8 内排序1.2.9 文件与外排序1.2.10 检索1.2.11 索引1.2.12 高级数据结构第2章 面向对象程序设计与C++概述2.1 面向对象程序设计概述2.1.1 面向对象程序设计：类和对象2.1.2 面向对象程序设计的特点2.2 C++编程概述2.2.1 C++中的类和对象2.2.2 对象的定义2.2.3 类的成员函数2.2.4 构造函数和结构函数2.2.5 友元2.2.6 类的继承2.3 重载2.3.1 函数重载2.3.2 运算符重载2.4 动态存储分配习题第3章 STL简介3.1 泛型编程简介3.1.1 泛型编程的需求3.1.2 C++中模板的使用3.2 STL容器简介3.2.1 vector3.2.2 deque3.2.3 list3.2.4 set和multiset3.2.5 map和multimap3.2.6 stack3.2.7 queue3.3 STL算法3.3.1 STL算法简介3.3.2 非变动性算法3.3.3 变动性算法3.3.4 变序型算法和排序算法3.3.5 已排序区间算法3.3.6 数值算法3.4 STL迭代器3.4.1 迭代器简介3.4.2 迭代器类型3.4.3 迭代器函数3.4.4 迭代器配接器3.5 文件流与输入输出3.5.1 全局性的Stream对象3.5.2 标准操作符3.5.3 标准10函数习题第4章 程序设计实践4.1 程序设计风格4.1.1 命名4.1.2 语句4.1.3 注释4.1.4 程序组织原则4.1.5 文档4.1.6 实践和原则4.2 界面4.3 测试、性能和可扩展性4.3.1 软件测试基本概念4.3.2 软件测试原则4.3.3 软件测试策略4.3.4 软件测试方法4.3.5 测试实例4.3.6 性能和可扩展性习题第5章 问题建模5.1 数学模型和数学建模5.1.1 数学模型5.1.2 数学模型示例——雨中行问题5.1.3 生产计划问题——线性规划模型5.1.4 预测疾病的发展变化趋势——马尔可夫链模型5.1.5 Buffon投针实验——蒙特卡罗方法5.1.6 公交最优路线查询系统设计问题5.2 设计模式5.2.1 设计模式的概念5.2.2 MVC的设计模式5.2.3 设计模式举例——工厂模式习题第6章 经典算法设计6.1 状态空间6.2 时间复杂度计算6.2.1 算法时间复杂度分析6.2.2 递推方程求解6.3 穷举法6.4 贪心法6.5 递归和回溯6.5.1 递归法6.5.2 回溯法6.5.3 回溯法的分支限界6.6 搜索与剪枝6.6.1 盲目搜索算法6.6.2 剪枝6.6.3 搜索的效率问题6.7 分治法6.7.1 分治策略6.7.2 降低递归算法复杂性的途径6.8 动态规划6.9 算法思想小结习题第7章 问题求解实践7.1 问题求解7.2 线性结构7.2.1 数组元素循环右移k位——时空权衡7.2.2 火车调度——栈的应用7.2.3 KMP模式匹配算法的应用7.3 树形结构7.3.1 二叉树遍历算法框架在问题求解中的应用7.3.2 树的应用7.3.3 选择树的应用7.3.4 树与二叉树的计数7.4 线段树7.4.1 线段树的定义及特征7.4.2 线段树的基本操作7.5 图的应用7.5.1 图的抽象7.5.2 图的搜索7.5.3 基于深度优先的拓扑排序7.5.4 第二最短路径7.5.5 唯一最小生成树7.5.6 有向图的强连通性问题7.6 排序与检索7.6.1 统计逆序对的归并思想7.6.2 求两个等长有序序列中位数的二分思想7.7 算法优化习题第8章 数据结构与算法技术应用实例8.1 搜索引擎中的数据结构技术8.1.1 概述8.1.2 抓取系统8.1.3 索引系统8.1.4 检索系统8.2 在线评测算法实习范例8.3 综合实习范例习题第9章 试题及参考答案9.1 期中考试9.1.1 2007年期中考试试题9.1.2 2007年期中考试参考答案9.1.3 2008年期中考试试题9.1.4 2008年期中考试参考答案9.2 期末考试9.2.1 2007年期末考试试题9.2.2 2007年期末考试参考答案9.2.3 2008年期末考试试题9.2.4 2008年期末考试参考答案9.3 高级专题考试9.3.1 2007年高级专题考试试题9.3.2 2007年高级专题考试参考答案9.3.3 2008年高级专题考试试题9.3.4 2008年高级专题考试参考答案9.4 实习课程考试9.4.1 2007年实习课考试试题9.4.2 2007年实习课考试参考答案9.4.3 2008年实习课考试试题9.4.4 2008年实习课考试参考答案参考文献

<<数据结构与算法实验教程>>

章节摘录

2.1.2 面向对象程序设计的特点 面向对象的程序设计试图用人类的思维方式去理解现实世界中对象的实质，从而进行程序设计，这种程序设计方法具有极大的优越性。面向对象的程序设计方法使开发人员摆脱了具体的数据格式与过程的限制，得以集中精力研究所处理的对象。

面向对象程序设计具有封装性、继承性、多态性三大特点，下面分别加以介绍。

1.封装性（Encapsulation） 封装性是指将关于一个对象的数据属性和操作方法封装在该对象中，使得对象之间相互独立。

封装的数据不能被外界访问，只能被封装在同一个类中的函数访问。

这些函数提供了对象数据和程序之间的接口。

避免数据被程序直接访问的概念被称为“信息屏蔽”或者“信息隐藏”，就是说把一部分数据属性、操作和处理对外界隐藏起来，使得外界无法看到这些隐蔽的信息。

这是面向对象方法的中心，是面向对象程序设计的基础。

对象就是封装了数据和操作这些数据的函数的逻辑实体，对象的使用者只能看到外部公用的接口，而具体操作的实现细节对外界不透明。

即对象的内部实现和外部行为通过封装区别开来。

这一特性降低了操作对象的复杂性，使用对象的程序员可以不知道对象内部的实现细节，只需要了解外部公用的接口就能够自由地操作对象。

现实世界中封装性的例子很多，比如汽车具有发动机这一属性，而作为用户在不知道这个属性的前提下也能照常驾驶汽车。

封装性使得对象对内部数据和方法提供了不同级别的保护，以防止数据在程序中被意外地改变或错误地使用。

这样就减少了不必要的破坏，保证了软件的模块化，有利于数据安全，提高了系统的可靠性和独立性。

2.继承性（Inheritance） 现实世界中经常碰到这样的情况：工厂生产某一系列的汽车，后续的车型只需要在前一车型的基础上做一定的改进或补充一些新的功能，而其他大部分部件都可以借鉴前者，因而不需要从头开始重新设计一款新的汽车。

继承性是指直接获得已有的特性而不必重新定义。

即已经定义了一个类，又想另外定义一个基本相同但需要改变或增加一些属性和方法的新类，这样只需在前者的基础上指明新类具有哪些新的特性即可，而不必重新设计新对象的全部特性。

前者称为父类或基类，后者称为子类或派生类。

一个父类具有它所派生的所有类的共同特征和行为。

⋮

<<数据结构与算法实验教程>>

编辑推荐

数据结构与算法实习课程注重实践能力和工程能力的培养，提高学生的实际动手能力，为后续专业基础课和专业课程学习打下坚实的基础。

张铭等编著的《数据结构与算法实验教程》旨在指导学生灵活运用数据结构和算法解决具体问题。以更好地巩固理论课程所学的内容。

《数据结构与算法实验教程》主要内容包括数据结构与算法教学实施方案、C++概述及基础知识、泛型化程序设计思想和标准模板库、STL基本概念、程序设计实践和技巧、问题建模专题讨论、问题求解实践和数据结构与算法应用实例。

《数据结构与算法实验教程》附有北京大学数据结构与算法课程期中、期末、高级专题以及期末试题与参考答案。

同时提供配套的教学讲义、课程网站等教学资源。便于教师教学。

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>