

## <<数据结构>>

### 图书基本信息

书名：<<数据结构>>

13位ISBN编号：9787302225065

10位ISBN编号：7302225060

出版时间：2010-6

出版时间：清华大学出版社

作者：段恩泽，肖守柏 主编

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

自从美国唐·欧·克努特教授用汇编语言编写的《计算机程序设计技巧》第一卷《基本算法》问世以来,已经出现了用Pascal、Java、C、C++、C#等语言编写的数据结构方面的书。

总体说来,这些语言基本上分为面向过程的语言和面向对象的语言两大类,也出现过采用两种语言描述数据结构的书籍,如C和C++语言描述,但作者实际上是按照C++语言描述。

同时采用面向过程和面向对象语言描述数据结构,目前国内基本上是空白。

对于同一种数据结构与算法,同时采用面向过程和面向对象语言进行描述,可以从中更深刻理解这两种思想的不同,这对于深刻理解计算机语言和思想有着重要的作用。

C语言是现在最流行的面向过程的语言,在业界使用非常广泛。

而C#语言作为微软在新一代开发平台(.NET)上推出的、完全面向对象的语言,凭着其简洁、高效、模板、标准化的特性,使得C#语言像程序设计语言中的一件艺术品,也吸引着越来越多的开发人员。

当然,C#语言也吸收了C语言的一些东西,如语法等,所以,在有些方面,C#与C是相似的。

鉴于此,编者决定编写本书,使用C和C#语言来描述数据结构与算法。

本书共分为8章,第1章介绍了数据结构和算法的基本概念及本书用到的数学知识、C语言知识和C#语言的知识;第2~6章分别讨论了线性表、栈、队列、串、数组、树、二叉树及图等常用的数据结构及其应用;第7、8两章分别讨论了排序和查找常用的各种方法及其应用。

由于本书采用C和C#两种语言对数据结构进行描述,为节省篇幅,本书对内容的处理如下:先对数据结构进行分析,如数据结构的概念、逻辑结构、物理结构等,然后给出每种数据结构的两种语言描述。

这样,把重点放在了数据结构本身上,而不只是考虑其语言实现。

这体现了“数据结构”课程的目的,即理解数据结构的特性,培养计算机的数据抽象能力和计算机思维能力。

本书由成都东软信息技术职业学院段恩泽、江西蓝天学院肖守柏两位老师主编,江西蓝天学院蔡爱平、江西吉安市信息化工作办公室习爱民两位老师共同完成。

其中,C语言部分,第1、2章和第3~5章分别由肖守柏、蔡爱平两位老师编写;第6~8章由习爱民老师编写。

C#语言部分由段恩泽老师编写,全书由段恩泽老师统稿、整理。

## <<数据结构>>

### 内容概要

“数据结构”是计算机及相关专业必修的核心基础课程。

本书采用C和C#两种语言作为算法描述的语言，对常用的数据结构与算法作了系统的介绍，力求概念清晰简单，注重实际应用。

本书通过两种语言对数据结构与算法的不同描述来揭示面向过程和面向对象两种不同的思想。

全书共分为8章，依次介绍了数据结构与算法及本书用到的数学、C和C#知识、线性表、栈和队列、串和数组、树型结构和图结构，以及排序和查找等基本运算。

本书主要面向高职高专院校计算机专业的学生，也可作为非计算机专业学生的选修教材及计算机应用技术人员自学参考书。

## &lt;&lt;数据结构&gt;&gt;

## 书籍目录

第1章 绪论	1.1 数据结构	1.1.1 学习数据结构的必要性	1.1.2 基本概念和术语	1.2 算法
1.2.1 算法的特性	1.2.2 算法的评价标准	1.2.3 算法的时间复杂度	1.3 数学预备知识	1.4 C预备知识
集合	1.3.2 常用的数学术语	1.3.3 对数	1.3.4 递归	1.4.1 指针
结构体	1.5 C#预备知识	1.5.1 接口	1.5.2 泛型编程	本章小结 习题
表的逻辑结构	2.1.1 线性表的定义	2.1.2 线性表的基本操作	2.2 顺序表	第2章 线性表
2.2.2 顺序表数据关系的语言描述	2.2.3 顺序表数据操作的语言描述	2.2.4 顺序表应用举例	2.3 单链表	2.3.1 单链表的定义
2.3.2 单链表数据关系的语言描述	2.3.3 单链表数据操作的实现	2.3.4 单链表应用举例	2.4 其他链表	2.4.1 双向链表
语言描述	2.4.2 循环链表	2.4.3 链表的存储和运算实现	2.4.4 链表的存储和运算实现	2.4.5 链表的存储和运算实现
习题	第3章 栈和队列	3.1 栈	3.1.1 栈的定义及基本运算	3.1.2 顺序栈的存储和运算实现
3.1.3 链栈的存储和运算实现	3.1.4 栈的应用举例	3.2 队列	3.2.1 队列的定义及基本运算	3.2.2 循环顺序队列的存储和运算实现
3.2.3 链队列的存储和运算实现	3.2.4 队列的应用举例	第4章 串和数组	4.1 串	4.1.1 串的基本概念及基本运算
4.1.2 串存储及运算实现	4.1.3 串的基本操作的实现	4.1.4 模式匹配	4.2 数组	4.2.1 数组的逻辑结构
4.2.2 数组的内存映像	本章小结	习题	第5章 树和二叉树	第6章 图
			第7章 排序	第8章 查找参考文献

## &lt;&lt;数据结构&gt;&gt;

## 章节摘录

在通用的计算机高级语言中，一般都有整型、实型、字符型、数组、枚举、结构体等数据类型。数据类型可分为两类：一类是非结构的原子类型，如整型、实型、字符型等基本类型；另一类是结构类型，它的成分可以由多个结构类型组成，并可以分解。

结构类型的成分可以是非结构的，也可以是结构的。

例如，数组的成分可以是整型等基本类型，也可以是数组等结构类型。

C#语言的数据类型比C多。

C#不仅有字符串这种基本的数据类型，还有类、委托、接口、事件等数据类型。

其中原因在于两者出现的背景不同。

C语言是随着UNIX操作系统的出现而流行起来的，主要用来书写系统代码，应用在底层上，速度快、效率高。

然而，C语言存在一些缺陷，如类型检查机制相对较弱、缺少支持代码重用的语言结构等，造成用C语言开发大程序比较困难。

而C#语言是Microsoft.NET Framework的首选语言，在Internet的大环境下出现的，为了编写面向网络的应用。

所以，C#语言的抽象程度更高，应用范围更广，能够开发大程序，因此，C#语言的数据类型的种类更多。

**5.抽象数据类型** 抽象数据类型（Abstract Data Type，ADT）是指一些数据以及对这些数据所进行的操作的集合。

数据之间有一定的关系，不同的抽象数据类型的数据之间的关系不同。

对数据进行的操作由数据间的关系决定。

数据间的关系不同，对数据的操作也就不同。

这些操作既向程序的其余部分描述了这些数据是怎么样的，也允许程序的其余部分改变这些数据。

一个ADT可能是一个图形窗体以及所有能影响到该窗体的操作，也可以是一个文件以及对这个文件进行的操作，或者是一张保险费率表及相关操作等。

传统的教科书在讲到抽象数据类型时，总会用一些数学中的概念进行描述。

典型的描述是：“你可以把抽象数据类型想成一个定义有一组操作的数学模型。

”这种描述给人的感觉好像你从不会真正用到抽象数据类型似的——除非拿它来催眠。

把抽象数据类型解释得这么空洞是完全丢了重点。

抽象数据类型可以让你像在现实世界中一样操作实体，而不必在底层的实现上摆弄实体，如不用再向链表中插入一个结点了，而是可以在电子表格中添加一个数据单元，或向一组窗体类型中添加一个新类型，或给火车模型加挂一节车厢。

也就是说，抽象数据类型使你可以在问题域中思考问题，而不管底层是如何实现的。

这就是“抽象数据类型”概念中“抽象”的含义。

也就是说，它把底层的信息隐藏起来，修改底层信息不会影响该抽象数据类型的使用，因为它提供的公有操作没有改变。

并且，如果数据类型发生改变，也只需在一处修改而不会影响到整个程序。

## <<数据结构>>

### 编辑推荐

《数据结构(C/C#语言版)》特色：通过C/C#对数据结构与算法的描述来揭示面向过程和面向对象的思想，理解数据结构特性，培养计算机数据抽象能力和计算机思维能力

<<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>