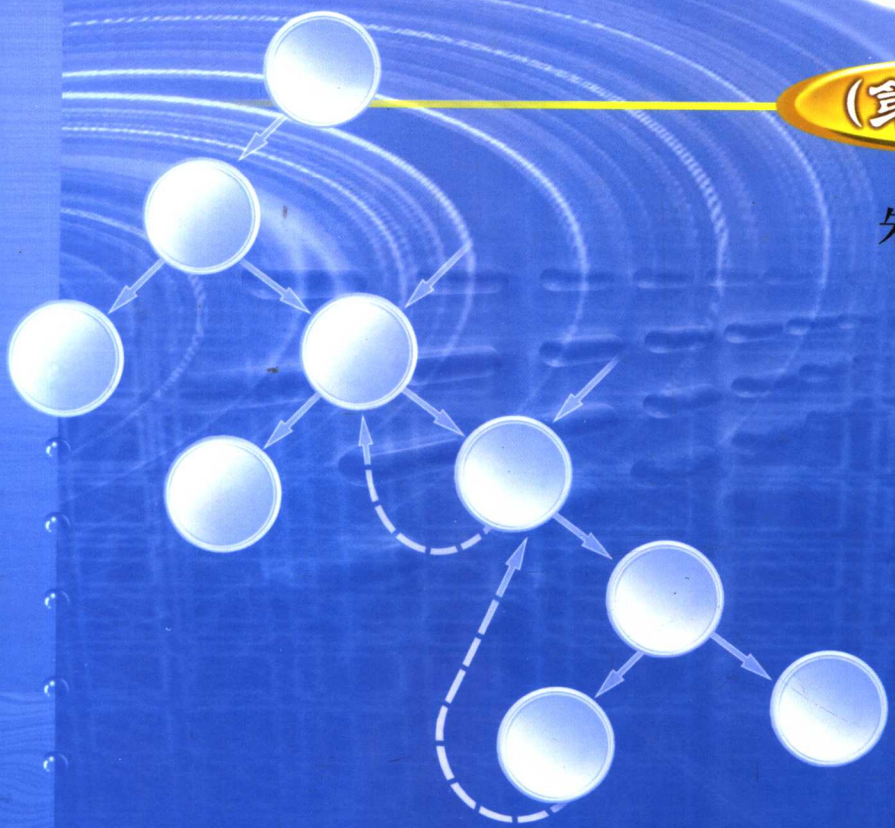
数据结构

——使用C语言

（第3版）

朱战立　编著



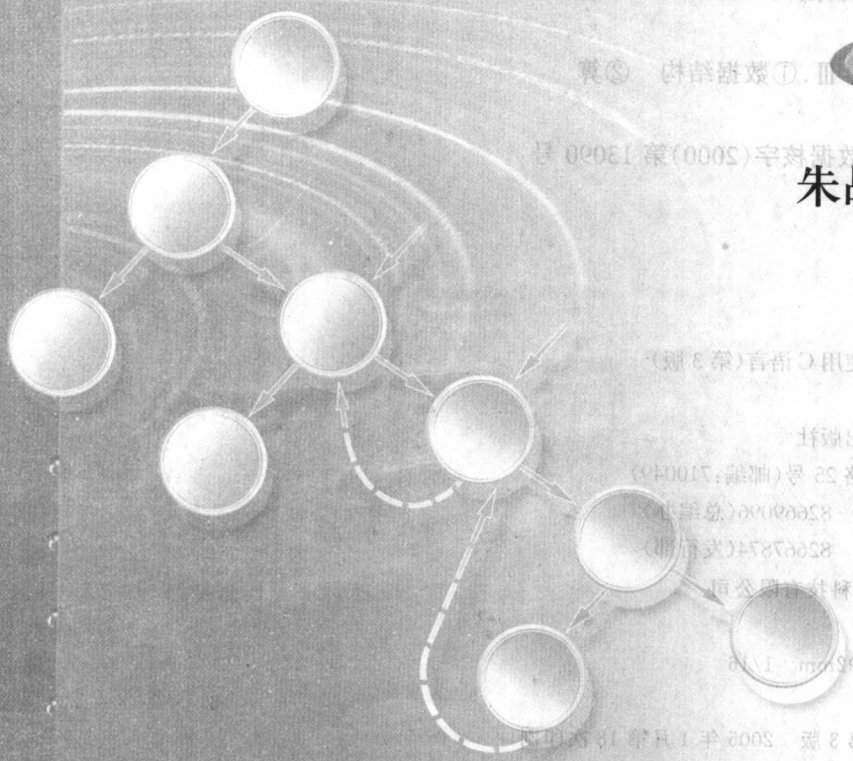
西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

数据结构

——使用C语言

（第3版）

朱战立　编著



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS
·西安·

内 容 简 介

数据结构是计算机专业和其他一些与计算机技术关系密切专业必修的核心课程。本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。

本书既可作为大专院校计算机等专业的教科书，也可作为从事计算机应用的工程技术人员的自学参考书。

图书在版编目(CIP)数据

数据结构——使用C语言/朱战立编著. —第3版.
—西安：西安交通大学出版社，2004.1
ISBN 7-5605-0883-9

Ⅰ.数… Ⅱ.①朱… Ⅲ.①数据结构 ②算法理论 Ⅳ.TP311.12

中国版本图书馆CIP数据核字(2000)第13090号

书　　名：数据结构——使用C语言(第3版)
编　　著：朱战立
出版发行：西安交通大学出版社
地　　址：西安市兴庆南路25号(邮编：710049)
电　　话：(029)82668315　82669096(总编办)
　　(029)82668357　82667874(发行部)
印　　刷　陕西江源印刷科技有限公司
字　　数　515千字
开　　本　787 mm×1 092mm　1/16
印　　张　21.375
版　　次　2004年1月第3版　2005年1月第18次印刷
印　　数　78 001 ～ 83 000
书　　号　ISBN 7-5605-0883-9/TP·142
定　　价　25.00元

第 3 版前言

“数据结构”是计算机学科的一门核心课程，也是其他一些与计算机学科关系密切学科或专业的一门必修或选修课程。

本书是第 3 版。本书的前两版受到众多读者的欢迎，被许多高校定为数据结构课程的教科书。作者近年来仍一直从事数据结构课程的教学工作，在教学过程中，感觉原书中的一些叙述方法和材料组织方法有改进或更新的必要，因此，重写了此书。第 3 版的改动主要包括：各章均以抽象数据类型为核心，对材料进行了重新编排组织；对全书的文字进行了仔细推敲，并为方便读者学习，把重要术语以黑体字格式给出；按统一的工程格式重写了全部算法；修改完善或重新设计了各章的应用举例；补充了一些习题，并把习题按难度和类型划分成基本概念题、复杂概念题、算法设计题和上机实习题；把原书散见于各章的 C 语言基础知识内容统一放在了第 0 章；重新设计了附录的内容。

作者进行上述改动是基于以下几个原因：(1)抽象数据类型是软件设计的核心，以抽象数据类型为核心组织材料是目前数据结构课程教学的要求。(2)计算机学科发展速度很快，很多内容需要更新，有些内容也需要淘汰。(3)算法设计是学生学习数据结构课程普遍感觉困难的一个问题，根据作者长期教学的经验，可以从三个方面来缓解学生学习的难度：首先是准确的算法思想叙述和合适的算法思想举例说明；其次是工程化的函数设计(工程化的函数设计是指函数设计中包括充分的注释语句)，这既可以方便学生学习理解，也符合软件设计的可读性要求；最后是完整的测试程序设计，通过测试程序对函数的调用，可以帮助学生更好地理解函数中参数的含义。本书就是从这三个方面来组织算法设计方面的材料的。

还需要说明的是本书的习题设计。本书在原书习题的基础上，补充了一些新习题，并把所有习题按类型和难度分成基本概念题、复杂概念题、算法设计题和上机实习题四大类，这既可以方便学生的学习，也可便于教师布置作业。数据结构课程既要培养学生软件设计方面的理论水平，也要培养学生基本的上机动手能力。上机实习是培养学生上机动手能力的重要环节。然而，完成上机实习作业是学生普遍感觉困难的一个问题，为此，附录 1 给出了上机实习内容规范和两个上机实习范例。为方便学生自学，附录 2 给出了部分习题的解答。

根据作者的经验，使用本教材授课约需 54～70 课时，其中包括 10 个左右课时的课内上机实践。课时数较少时，目录中标有“*”号的内容可不讲授。

尽管作者在写作过程中非常认真和努力，但错误和不足之处仍在所难免，敬请读者批评指正。

朱战立

2003.10

第 2 版前言

数据结构是计算机学科本科生、专科生的核心课程，是计算机程序设计的重要理论技术基础。

本书是第 2 版。较之第 1 版，其主要的改动包括：重写了全部算法并规范了算法的书写格式，改写和补充了各章的应用实例，给出了习题的部分答案，添加了若干上机实验题目的分析和程序设计。

本书用 C 语言作为算法描述语言。目前，数据结构教材采用 C 语言作为算法描述语言已渐成趋势。数据结构课程对学生编程能力的训练包括两方面的内容：一个方面是训练学生理解各种基本数据结构的概念，并能理解和编写出各种典型的算法；另一个方面是训练学生应用各种典型算法进行具体应用问题的程序设计，这包括程序中变量设计、函数中参数设计、程序的书写格式等方面的训练。目前数据结构教材采用 C 语言作为算法描述语言的也有两种：一种是采用类 C 语言，另一种是直接采用 C 语言。采用类 C 语言有利于学生把注意力放在算法的设计和分析上；采用 C 语言能兼顾两个方面的训练，这对高等工科院校强调对学生实际动手能力的训练是非常适宜的。

第 2 版在附录中添加了部分书写习题答案和若干上机实验题目的分析及程序设计。添加书写习题答案有利于读者自学，添加上机实验题目的分析和程序设计既有利于学生实验报告的书写和实际编程能力的训练，也有利于教师安排和设计学生的上机实验题目。

本书讲授学时可为 44～60 学时。讲授时间低于 60 学时时可适当删去带“*”的章节内容。

本书是在第 1 版基础上，由刘天时完成第 2,3 章和 4.5 节全部算法的改写和完善工作，罗进元重新调试了第 7,8,10 章的几个算法，其余章节的算法改写工作和全部章节的文字修改工作均由朱战立完成，全书由朱战立修改定稿。

编著者

1999.10

前　言

数据结构是一门计算机专业和计算机应用专业本科、专科学生必修的专业基础课。它是许多涉及数据结构设计和算法设计课程，如操作系统、编译原理、计算机图形学、人工智能等的先导课。数据结构也是非计算机专业培养学生软件设计能力重点选择的一门专业课。

本书共分 11 章。第 1 章介绍了数据结构的基本概念，概述了 C 语言的数据类型，说明了算法设计目标和算法效率度量；第 2 章到第 5 章分别介绍了线性结构中线性表、堆械、队列、串和数组的基本概念，以及存储结构和一些基本操作的实现；第 6 章介绍了递归概念以及递归算法的设计方法和用非递归算法模拟递归算法的基本方法；第 7 章和第 8 章分别介绍了非线性结构中树和二叉树以及图的基本概念、存储结构、一些基本操作的实现及它们的典型应用；第 9 章和第 10 章分别介绍了排序和查找的各种常用算法，并对各种算法进行了简单的性能分析；第 11 章介绍了文件的概念和文件的几种组织方法。

本书在内容组织上力求丰富充实，结合实际；在语言描述上力求深入浅出、简洁明了。为了便于学习和理解，书中列举了大量例题，并在一些章后设有专门小节讨论较大型的综合应用问题，每章都配有适量习题。

本书既可作为计算机应用专业本科、专科学生的教材，也可作为非计算机专业本科学生的教材。对从事计算机应用的工程技术人员，本书也是十分有价值的参考书。

目前国内数据结构教材基本使用类 PASCAL 语言或用 PASCAL 语言作为算法描述语言，这与 C 语言的广泛使用和计算机应用专业教学的发展变化要求极不相符。本书是编著者在多年来讲授数据结构和 C 语言程序设计课程的基础上编写的。本书采用 C 语言作为算法描述语言，书中全部算法都通过了上机调试。本书第 1 章至第 6 章及第 11 章由朱战立编写，第 7 章至第 10 章由李文编写，全书由朱战立修改定稿。

本书由西安交通大学冯博琴教授审阅，西安交通大学出版社对本书的出版给予了大力支持，在此一并表示谢意。

编著者

1997.1

目 录

第0章 C语言程序设计

数据结构课程的一个主要内容是讨论各种算法，算法需要用某种语言来书写，本书采用C语言描述算法。另外，数据结构课程既要培养学生软件设计方面的理论水平，也要培养学生基本的上机动手能力。鉴于以上两个原因，使用本书作为教材时，需要学生有C语言程序设计的基础知识。考虑到有些学生C语言程序设计知识的不足或不系统，作为全书的预备知识，本章简要介绍在算法描述中和程序设计中要用到的一些C语言程序设计的基本知识。

0.1 程序的结构

写文章要考虑文章的结构，编写程序也要考虑程序的结构。程序是处理特定问题的用程序设计语言书写的语句序列。当要处理的问题复杂时，程序将会很长。因此，程序的结构就非常重要。

程序结构是以模块化为基础的。所谓模块化就是把一个较大的程序划分成许多块，每块完成一个较小的、特定的功能。在C语言中，函数就是完成特定功能的程序块。函数是不能直接被执行的，需要由一个称为主函数的函数来组织整个程序的执行顺序。因此，C语言程序的结构一般如下：

函数1
函数2
⋮
函数n
主函数

由于函数中的参数是一个变量，其具体取值由主函数调用时传送的数据决定，因此，一个函数可以被许多主函数用不同的数据调用。这样，程序的模块化结构就既可以使程序的结构清晰，也可以实现模块的共享。

函数分为库函数和用户自定义函数两大类。**库函数**也称作**系统函数**，是包括在高级语言软件中的提供基础功能的函数。**用户自定义函数**是软件开发者根据所处理的特定问题设计的

提供特定功能的函数。

通常,库函数很多,为方便用户使用,库函数一般是按照函数的类型组织成文件形式提供给用户的。也就是说,一个库函数的文件中包括了同类型的许多库函数。库函数文件通常以.h 作为文件后缀。例如,C 语言的 stdio.h 文件中就包括了 printf(),scanf(),open(),close()等函数,stdlib.h 文件中包括了 exit()等函数,malloc.h 文件中包括了 malloc(),free()等函数。

如果 C 语言的主函数中要使用某个库函数,需要用文件包含命令把包含该库函数的整个文件包含(或称加载)到用户的程序中。文件包含命令的语句格式为:

```
#include <库函数文件名>
```

例如,若用户的主函数中要使用库函数 printf()和 scanf(),用户的程序中就要有如下的文件包含命令:

```
#include <stdio.h>
```

用户设计自定义函数时要符合 C 语言函数的格式。用户自定义的函数如果很多时,也可以按照类型组织成用户库函数形式,即组织成用户自己命名的文件形式。如果 C 语言的主函数中要使用某个用户库函数,也需要用文件包含命令把包含该用户库函数的整个文件包含到用户的程序中。此时,文件包含命令的语句格式为:

```
#include "用户库函数文件名"
```

文件包含命令#include 后边使用符号<>和符号""的差别是:当使用符号<>时,系统将直接在系统的库函数文件目录下寻找该文件;当使用符号""时,系统将首先在用户当前的工作目录下寻找该文件。用户当前的工作目录是指包含主函数的程序存放的目录。因此,为方便起见,用户的库函数文件通常应该和用户的程序放在一个目录中。

0.2 函数

C 语言函数的一般形式为:

```
数据类型 函数名(数据类型 参数 1,……, 数据类型 参数 n)
{
    函数体
}
```

可见,一个函数由函数名、参数和函数体三部分组成。一个函数的函数名既是该函数的代表,也可以用返回值带回函数的处理结果。函数的参数称为**形参**或**虚参**,函数的参数有输入型参数和输出型参数两种。调用函数的函数(最常见的是主函数)称为调用函数,被调用函数调用的函数称为被调用函数。调用函数中使用的参数称为**实参**。

C 语言中若函数形参的形式不同,则实参和形参间传送数据的功能就不同,从而可以实现输入型参数和输出型参数两种不同的功能。

0.2.1 返回值

一个函数的函数名既是该函数的代表,同时也是一个变量。系统为函数名变量分配有与定义的数据类型相一致的内存单元空间。由于函数名变量是像全局变量一样定义的,所以调

用函数可以使用保存在函数名变量中的数据。由于函数名变量通常用来把函数的处理结果数据带回给调用函数，所以一般把函数名变量称为**返回值**。

C语言规定没有返回值的函数要定义成空类型。空类型的标识符是 void。

例0-1　设计一个从两个整数类型数据中得到较大数值的函数，并设计一个主函数调用该函数。

设计　包括函数和主函数的程序设计如下：

```
#include <stdio.h>

int Max1(int x1, int x2)
{
    if(x1 >= x2) return x1;
    else return x2;
}

void main(void)
{
    int t1 = 5, t2 = 8, max;

    max = Max1(t1, t2);
    printf("Max = %d\n", max);
}
```

在上述算法的设计中，函数名具有整数类型的返回值，用于返回得到的较大数值。由于函数名变量 Max1 是定义在主函数外，并且位于主函数的前边，所以主函数 main()中可以像使用全局变量一样使用函数名变量 Max1。

0.2.2　输入型参数

输入型参数是指调用函数只通过该参数传送数据给被调用函数。由于调用函数只向被调用函数传送数据，所以称为输入型参数。输入型参数也称作值参。

C语言函数实现输入型参数的方法是值传送，即系统把实参的数据直接复制给形参。

例0-1中的两个形参只是从调用函数(即主函数)向被调用函数(即 Max1()函数)中传送数据，所以 Max1()函数中的两个形参 x1 和 x2 都是输入型参数。

例0-1的主函数中实际输入数据(即实参)t1 和函数中的形式输入数据(即形参)x1 的代换过程如图0-1所示。在函数调用时，实参 t1 把数值5直接复制给形参 x1；在函数调用结束时，形参 x1 分配的内存单元被系统收回。实参 t2 和形参 x2 的代换过程与此类同。

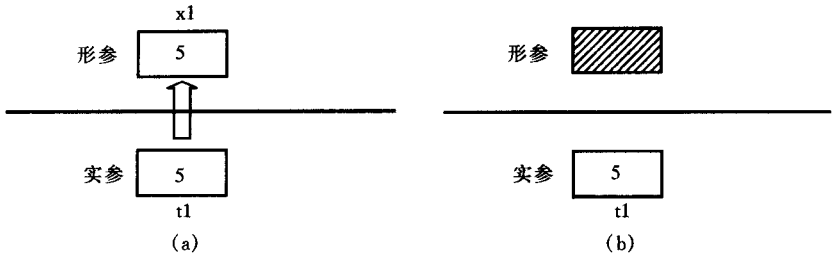


图 0-1　值传送的实参和形参代换过程
(a) 函数调用时；　(b)调用结束时

0.2.3　输出型参数

输出型参数是指调用函数只通过该参数把被调用函数处理后得到的结果数据传送给调用函数。由于这样的参数数据是离开被调用函数的，所以称为输出型参数。输出型参数也称作变参。

C 语言函数实现输出型参数的方法是地址传送，即系统把实参的地址复制给形参，被调用函数把处理的结果数据存放在形参地址所指的内存单元中。由于此时调用函数的实参地址与被调用函数的形参地址相同，所以调用函数可以得到被调用函数处理的结果数据。C 语言函数地址传送的具体实现方法是把形参设计成指针类型，实参用某个变量的地址。

例 0-2　设计一个从三个整数类型数据中得到最大数值和次大数值的函数。

设计　函数要有三个输入型参数，设为 x1，x2，x3，分别表示三个原始数据。如果函数只带回一个返回值时，通常的方法是用返回值带回处理的结果数据，但此时函数要带回两个返回值，因此必须再设计两个输出型参数，设为 y1 和 y2。

程序设计如下：

```
#include <stdio.h>

void Max2(int x1, int x2, int x3, int *y1, int *y2)
/* 注意:输出型参数 y1 和 y2 定义为指针类型 */
{
    if(x1 >= x2 && x1 >= x3)
    {
        *y1 = x1;
        if(x2 >= x3) *y2 = x2;
        else *y2 = x3;
    }

    if(x1 >= x2 && x1 < x3)
    {
        *y1 = x3;
```

```
        if(x1 >= x2) *y2 = x1;
        else *y2 = x2;
    }

    if(x1 < x2 && x2 >= x3)
    {
        *y1 = x2;
        if(x1 >= x3) *y2 = x1;
        else *y2 = x3;
    }

    if(x1 < x2 && x2 < x3)
    {
        *y1 = x3;
        if(x1 >= x2) *y2 = x1;
        else *y2 = x2;
    }
}

void main(void)
{
    int v1 = 5, v2 = 9, v3 = 7, f1, f2;
    .
    Max2(v1, v2, v3, &f1, &f2);                    /* &f1 表示 f1 的地址 */
    printf("f1 = %d f2 = %d\n", f1, f2);           /* 可取到 f1 的数值 */
}
```

例0-2的主函数中实参 &f1 和形参 y1 的代换过程如图0-2所示。在函数调用时,实参 &f1 把数值(即变量 f1 的地址)复制给形参 y1,因此形参 y1 也指向变量 f1;在函数调用结束时,形参 y1 分配的内存单元被系统收回,但主函数中实际存在的变量 f1 中保存了函数处理得

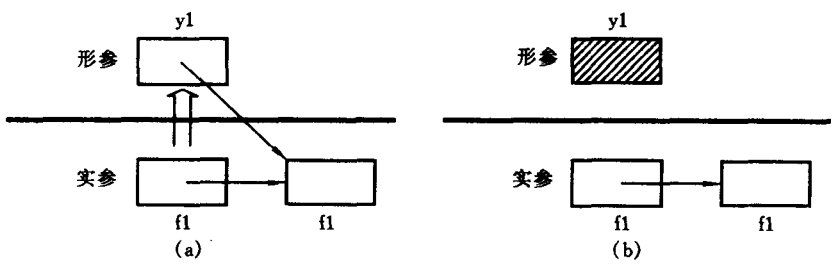


图0-2 地址传送的实参和形参代换过程
(a) 函数调用时; (b)调用结束时