

计算机类主干课程系列教材

数据结构实验与 课程设计教程

◎ 李志敏 李纪平 主编

The CDIO logo is a blue sphere with the letters "CDIO" in white. It is positioned on the left side of the cover, partially overlapping a dark, circular graphic that resembles a CD or a globe with grid lines.

科学出版社

计算机类主干课程系列教材

数据结构实验与课程设计教程

李志敏 李纪平 主 编

科学出版社

北 京

版权所有,侵权必究

举报电话:010-64030229;010-64034315;13501151303

内 容 简 介

本书共分三篇。第一篇为基础实验,主要是介绍数据结构与算法基础知识的实验,包括线性表、栈和队列、串、数组、广义表、树和二叉树、图、查找、排序等内容,一共有9个实验。第二篇为应用实训,是数据结构知识的应用与提高,包括链表的应用、栈和队列的应用、树结构的应用、图结构的应用以及文本文件检索等综合性实验内容,共10个实验。第三篇为课程设计,详细介绍7个课程设计的课题,综合性较强,另外还给出了一部分实训项目,内容涉及数据结构课程的多个应用领域,以引导学生进行开发实践。

本书既可以作为高等学校“数据结构”课程的实验、实习、实训教材,也可用作教师指导大学生程序设计竞赛的参考书。

图书在版编目(CIP)数据

数据结构实验与课程设计教程/李志敏,李纪平主编. —北京:科学出版社, 2016.05

ISBN 978-7-03-048295-2

I. ①数… II. ①李… ②李… III. ①数据结构—高等学校—教材
IV. ①TP311.12

中国版本图书馆CIP数据核字(2016)第098895号

责任编辑:张颖兵 闫 陶/责任校对:董艳辉

责任印制:彭 超/封面设计:苏 波

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

武汉市首壹印务有限公司印刷

科学出版社发行 各地新华书店经销

*

开本:787×1092 1/16

2016年5月第 一 版 印张:19 1/2

2016年5月第一次印刷 字数:440 000

定价:40.00 元

(如有印装质量问题,我社负责调换)

前 言

本书基于 CDIO 工程教育理念重新设计改编了第一版的实验和课程设计项目。充分吸收了编者获得的各类教学研究成果,尤其是湖北省高等学校教学改革研究项目“基于 CDIO 工程教育理念的编译原理课程教学改革”(编号 2012367)。

数据结构是计算机专业的必修课程之一,也是计算机学科的一个重要分支研究领域。它是算法设计与分析、操作系统、软件工程、数据库概论、编译原理、计算机图形学等专业课程的基础课。透彻掌握数据结构的理论与方法,有助于合理地组织存储数据、设计高效的算法、编写高质量的程序,满足实际应用的需要。

本书共分三篇。第一篇为基础实验,介绍数据结构与算法基础知识和实验,包括线性表、栈和队列、串、数组和广义表、树和二叉树、图、查找、排序等内容,可以帮助学生熟练掌握基础知识和基本算法。第二篇为应用实训,是数据结构知识的应用与提高,包括链表的应用、栈和队列的应用、树结构的应用、图结构的应用以及文本文件检索等综合性实验内容,可以培养分析和解决实际问题的能力。第三篇为课程设计,综合性比较强,精选实际应用项目引导学生进行开发实践。内容涉及数据结构课程的多个应用领域,供各类教学和学习者参考。

本书有如下特色。

(1) 本书努力做到“门槛低,坡度缓,层次高”。在内容编排上,先从验证型实验出发,引导学生独立进行综合实验设计,最后达到能完成综合课程设计的目标。

(2) 本书内容丰富、实用性强、适用面广。本书既可作为“数据结构”课程的学习参考书和实验指导书,又可供高等院校各专业学生实验、课程设计、竞赛指导和考前复习,还可供教师和其他专业技术人员参考。

(3) 根据数据结构教学大纲精心选择基础实验内容。考虑到本课程开课时间一般在本科低年级,学生编程能力不是很强,本书对基础实验编写做到过程描述详细,代码注释完整,便于初学者模仿训练,循序渐进,稳步提高。

(4) 应用实训的内容按课程教学顺序设计。同时考虑到实际应用的要求,使综合实验既巩固大纲要求的知识点,又接近课程设计项目的需要,循序渐进训练学生的分析问题和解决问题以及编程能力。

(5) 课程设计项目选题新颖,应用性强。将课程设计过程进行模块化描述。每个项目都有设计要求、需求分析、算法设计、编程和实例测试。通过详细的实例分析,锻炼学生的动手能力,引导学生完成课程设计。

(6) 本书配套学习资源丰富,C 语言程序源代码完整,可以直接编译运行。

本书由湖北工程学院计算机与信息科学学院李志敏、李纪平执笔编写。参加本书编写和讨论工作的有涂俊英、王署霞、叶从欢、陈佛敏、桂超、李国屏、王邯、许中元、张万山、

周天宏、周云才, 贵州大学姚凯学、彭长根、汪学明、龙土工, 贵州财经大学文静华, 贵州民族大学杨承中、张儒良、潘峰, 贵州师范大学张仁津, 铜仁学院黄贻望, 西安工程大学张凯兵, 赣南师范学院申鼎才等。本书的出版得到了湖北工程学院的大力支持。在写作过程中, 参考了相关教材及网络资料, 在此一并致谢。

由于作者水平有限, 书中难免存在不足之处, 殷切希望广大读者批评指正。

编 者

2015 年 10 月于湖北工程学院

目 录

第一篇 基础实验

第 1 章 准备实验	2
1.1 C 语言实验	2
1.2 拓展训练	16
第 2 章 线性表	17
2.1 线性表的顺序表示和实现	17
2.2 线性表的链式表示和实现	26
2.3 拓展训练	35
第 3 章 栈与队列	36
3.1 栈的顺序表示和实现	36
3.2 队列的链式表示和实现	43
3.3 队列的顺序表示和实现	50
3.4 拓展训练	57
第 4 章 串	58
4.1 串的定长顺序存储表示和实现	58
4.2 拓展训练	68
第 5 章 数组和广义表	69
5.1 数组的顺序存储和实现	69
5.2 稀疏矩阵三元组顺序表存储和实现	76
5.3 广义表的表示和实现	85
5.4 拓展训练	92
第 6 章 树和二叉树	93
6.1 二叉树的链式存储和实现	93
6.2 拓展训练	100
第 7 章 图	101
7.1 图的邻接表表示及其基本操作	101
7.2 拓展训练	117
第 8 章 查找	118
8.1 顺序查找、二分查找、二叉排序树查找的实现	118
8.2 拓展训练	127

第9章 内部排序	128
9.1 插入与交换排序操作	128
9.2 选择排序与归并排序操作	135
9.3 链式基数排序	143
9.4 课外训练	150

第二篇 应用实训项目

应用实训项目一 一元多项式加、减、乘、除运算的实现	153
应用实训项目二 迷宫问题实现	164
应用实训项目三 舞伴搭配问题	170
应用实训项目四 压缩软件实现	177
应用实训项目五 校园导游咨询	187
应用实训项目六 散列表的设计与实现	193
应用实训项目七 简单文本编辑器设计与实现	202
应用实训项目八 图书馆书目检索	214
应用实训项目九 拓扑排序	223
应用实训项目十 关键路径	240

第三篇 课程设计

课程设计过程简介	245
数据结构课程设计实施方案	250
课程设计一 线性表	252
1.1 通讯录管理系统	252
1.2 学生成绩管理系统(单链表)	257
1.3 课程设计项目	262
课程设计二 栈和队列	267
2.1 停车场管理系统	267
2.2 课程设计项目	270
课程设计三 串的应用	273
3.1 文本文件的检索	273
3.2 文本文件单词的检索与计数	274
3.3 课程设计项目	276
课程设计四 树型结构的应用	278
4.1 压缩软件设计	278
4.2 课程设计项目	280
课程设计五 图结构的应用	281
5.1 交通咨询系统	281
5.2 计算机专业教学计划编制问题	284

5.3 课程设计项目	286
课程设计六 排序与查找	288
6.1 航班信息的查询与检索	288
6.2 课程设计项目	290
课程设计七 文件信息管理系统	294
7.1 图书管理信息系统的设计与实现	294
7.2 课程设计项目	302

第一篇 基础实验

本篇第 1 章准备实验是复习 C 语言程序设计,包括函数的参数传递与结果返回、动态存储分配等基础知识和方法。教师根据实际情况取舍,不要求提交实验报告。

第 2~9 章是数据结构及其有关算法的基础实验,必须按教学要求提交独立完成的实验报告。每次实验完成后,提交电子版的实验报告和程序源代码(含上机练习题)。压缩包命名为姓名-学号-实验名称。实验报告的主要内容如下。

实验报告页眉: 数据结构课程实验报告

实验题目(居中,四号,黑体)

姓名 _____ 学号 _____ 指导教师姓名 _____
实验时间 _____ 实验地点 _____

实验目的(左对齐,小四号,黑体)(正文用五号宋体)

1. 问题描述(小四号,宋体)(正文用五号宋体)
2. 数据结构设计(小四号,宋体)(正文用五号宋体)
3. 功能(函数)设计(小四号,宋体)(正文用五号宋体)
包括函数名称和接口,主要函数的初始条件与操作结果,程序流程图。
4. 界面设计(小四号,宋体)(正文用五号宋体)
5. 编码实现(小四号,宋体)(正文中文用五号宋体,英文用 Times New Roman)
包括程序预处理、数据结构、功能函数、菜单显示、主函数等完整程序。
6. 运行测试(小四号,宋体)(正文用五号宋体)
运行程序,测试各功能函数,并将运行效果截图。
7. 上机训练题(小四号,宋体)(正文用五号宋体)
按教师要求,选做若干题目。
8. 实验小结(小四号,宋体)(正文用五号宋体)

小结本次实验过程中出现的错误、面临的困难以及这些错误和困难的解决方案和今后需要努力的方向。

第 1 章 准备实验

1.1 C 语言实验

1.1.1 函数的参数传递与结果返回

1. 函数参数传递

1) 形式参数和实际参数

函数定义中的参数是形式参数(形参)。函数的调用者提供给函数的参数称为实际参数(实参)。在函数调用之前,实际参数的值将被复制到形式参数中。

2) 参数传递

在运行程序的过程中,把实际参数的值传给形式参数,这就是函数参数的传递。C 语言中函数参数的传递有三种形式:值传递、地址传递、引用传递。注意:形参和实参可能不止一个,如果多于一个,函数声明、调用、定义的形式都要一一对应,不仅个数要对应,参数的数据类型也要对应。

(1) 变量与变量之间的值传递。

/*第 1 章验证实验 1

功能:从键盘输入一个整数,并在屏幕上显示该整数*/

#include <stdio.h>

void fun1(int num_back); /*注意函数声明的形式*/

void main()

{

int num;

printf("请从键盘输入一个整数\n");

scanf("%d", &num);

fun1(num); /*注意调用形式*/

}

void fun1(int num_back) /*注意定义形式*/

{

printf("%d\n", num_back);

}

该程序在主函数中,先定义一个变量,然后输入一个值,在 fun1() 函数中输出。当程序运行 a(num); 这一步时,把 num(实参)的值赋值给 num_back(形参)。

/*第1章验证实验2

功能:从键盘先输入一个整数,再输入一个浮点数,并在屏幕上显示*/

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

```
void fun2(int num1_back, float num2_back); /*注意函数声明的形式*/
```

```
void main()
```

```
{
```

```
    int num1;
```

```
    float num2;
```

```
    printf("请从键盘输入一个整数\n");
```

```
    scanf("%d", &num1);
```

```
    printf("请从键盘输入一个小数\n");
```

```
    scanf("%f", &num2);
```

```
    fun2(num1, num2); /*注意调用形式*/
```

```
}
```

```
void fun2(int num1_back, float num2_back) /*注意定义形式*/
```

```
{
```

```
    printf("%d, %f\n", num1_back, num2_back);
```

```
}
```

函数有两个参数,一个是整型,一个是浮点型,那么在声明、调用、定义的时候,参数个数要一样,类型也要对应。如果不对应,有可能出现编译错误。有时即使通过了编译,也会在数据传递过程中出现错误。

/*第1章验证实验3

功能:从键盘输入两个整数 a 和 b,运行 fun3(a,b),并在屏幕上显示运行结果*/

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

```
void fun3(int x, int y)
```

```
{
```

```
    int tmp;
```

```
    tmp=x;
```

```
    x=y;
```

```
    y=tmp;
```

```
    printf("x=%d,y=%d\n", x, y);
```

```
}
```

```
void main()
```

```
{
```

```
    int a;
```

```
    int b;
```

```
    printf("请从键盘输入一个整数 a\n");
```

```
    scanf("%d", &a);
```

```
    printf("请从键盘输入一个小数 b\n");
```

```
    scanf("%d", &b);
```

```
    printf("运行 fun3 (a,b) 前\n a= %d, b= %d\n", a, b);
```

```

printf("运行 fun3 (a,b) 后\n");
fun3(a,b);
printf("a=%d,b=%d\n", a, b);
}

```

fun3(int x, int y)的功能是交换 x 和 y 的值。但观察到调用 fun3 (a,b) 后,a 和 b 的值没有发生预期的改变。这说明实参的值传递给函数局部工作区相应的副本中,这里 $x=a, y=b$ 。函数使用副本执行运算。因此函数修改的是副本的值,即 x 和 y 的值。实参 a 和 b 的值不变。

(2) 数组与数组之间的地址传递。由于数组名是指向数组的指针,数组之间的值传递本质上是地址传递。函数调用数组时,实参就是数组名。

/*第1章验证实验4 一维数组作为函数参数

功能:从键盘先输入 5 个整数元素的一维数组,并在屏幕上显示*/

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

```
void fun4(int array[]) ; /*注意函数声明的形式*/
```

```
void main()
```

```
{
```

```
int array[5],i;
```

```
for(i=0;i<5;i++) scanf("%d",&array[i]);
```

```
fun4(array); /*注意调用形式,实参是数组名*/
```

```
}
```

```
void fun4(int array[])/*注意定义形式*/
```

```
{
```

```
int i;
```

```
for(i=0;i<5;i++) printf("%d\t",array[i]);
```

```
printf("\n");
```

```
}
```

注意:一维数组的数组名 array 是指向这个数组首地址的指针。数组作为参数在函数原型中一定要写"void fun4(int array[])"。在调用时去掉[],只需写成 fun3(array),将 array 数组首地址传递给函数 fun4()。

/*第1章验证实验5 二维数组作为函数参数

功能:在屏幕上显示如下图案

```

X! O! X
+---+
! X! O
+---+
X! ! O */

```

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

```
void fun5(char array[][3]); /*注意函数声明的形式*/
```

```
main(){
```

```
char array[3][3]={{'X','O','X'},{' ','X','O'},{'X',' ','O'}};
```

```

fun5(array); /*注意调用形式,实参是数组名*/
return 0;
}

void fun5(char array[][3]) /*注意定义形式*/
{ int row,column;
  for(row=0;row<3;row++)
  { if(row!=0)
    printf("---+---+---\n");
    for(column=0;column<3;column++)
    { if(column!=0) printf("|");
      printf("%3c",array[row][column]);
    }
    printf("\n");
  }
}

```

注意:二维数组的数组名 array 是指向这个数组首行地址的指针。在调用的时候,数组作为参数在函数原型中一定要写“void fun5(char array[][])”。

在调用的时候,函数要求的是输入数组。只需写成 fun5(array),将 array 数组首地址传递给函数 fun5()。

(3) 指针与指针之间的地址传递。函数实参是指针变量时,将指针变量的地址传到被调用函数中,使形参指针变量也指到实参所指向的存储单元。

/*第1章验证实验6 指针变量作为函数参数

功能:从键盘输入两个整数 a 和 b,交换 a 和 b 的值。在屏幕上显示运行结果*/

#include <stdio.h>

void fun6(int *px, int *py) /*注意函数申明形式,形参是指针变量*/

```

{
    int tmp=*px;
    *px=*py;
    *py=tmp;
    printf("px=%d,py=%d\n",*px,*py);
}

```

void main()

```

{
    int a;
    int b;
    printf("请从键盘输入一个整数 a\n");
    scanf("%d",&a);
    printf("请从键盘输入一个小数 b\n");
    scanf("%d",&b);
    printf("运行 fun6 (a,b) 前 a=%d,b=%d\n", a, b);
    fun6( &a,&b); /*注意调用形式,实参是变量 a 和 b 的地址*/
}

```

```
printf("运行 fun6 (a,b) 后 a=%d,b=%d\n", a, b);
```

```
}
```

与第1章验证实验3比较,该程序实现了交换a和b的值。

/*第1章验证实验7 指向一维数组的指针作为函数参数

功能:从键盘输入N个整数,用选择法从大到小排序。在屏幕上显示运行结果*/

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

```
#define m 1000
```

```
void fun7(int x[], int n) /*注意函数定义形式,形参是一维数组*/
```

```
{
```

```
int i,j,k,t;
```

```
for(i=0;i<n-1;i++)
```

```
{k=i;
```

```
for(j=i+1;j<n;j++)
```

```
if (x[j]>x[k])k=j;
```

```
if(k!=i)
```

```
{t=x[k]; x[k]=x[i]; x[i]=t;}
```

```
}
```

```
}
```

```
void main()
```

```
{
```

```
int i,N,a[m];
```

```
printf("请从键盘输入待排序整数个数 N<=1000:\n");
```

```
scanf("%d",&N);
```

```
int *p=a;
```

```
printf("请输入 N=%d 个待排序整数:\n",N);
```

```
for(i=0;i<N;i++) scanf("%d",&a[i]);
```

```
printf("你输入 N=%d 个待排序整数是:\n",N);
```

```
for(i=0;i<N;i++) printf("%d\t",*p++);/*注意本循环结束时,指针 p 不再指向  
数组首地址*/
```

```
printf("\n");
```

```
p=a; /*设置指针 p 指向首地址*/
```

```
fun7(p, N); /*注意函数申明形式,形参是一维数组*/
```

```
printf("从大到小排序结束后的顺序是:\n");
```

```
for(i=0;i<N;i++) printf("%d\t",*p++);
```

```
printf("\n");
```

```
}
```

形参是一维数组,可以用指向数组首地址的指针作为实参。本例也可以用数组名作为实参。

注意定义数组时,a[m]中的m是事先给定的常数。

/*第1章验证实验8 指向二维数组的指针作为函数参数

功能:从键盘输入一个 3×4 阶矩阵,转置后在屏幕上显示运行结果*/

```
#include <stdio.h>

void InputM(int (*a)[4], int n, int m); //形参是指向二维数组的指针变量。
void OutputM1(int (*a)[4], int n, int m);
void OutputM2(int (*b)[3], int n, int m);
void fun8(int (*a)[4], int (*b)[3]);

void main()
{
    int a[3][4], b[4][3]; //a 存放原矩阵, b 存放转置矩阵
    printf("请从键盘输入一个  $3 \times 4$  阶矩阵:\n");
    InputM(a, 3, 4);
    printf("你输入的  $3 \times 4$  阶矩阵是:\n");
    OutputM1(a, 3, 4); //实参是二维数组名
    fun8(a, b);
    printf("转置矩阵:\n");
    OutputM2(b, 4, 3);
}

void InputM(int (*a)[4], int n, int m)
{
    int i, j;
    for(i=0; i<n; i++)
        for(j=0; j<m; j++)
            scanf("%d", *(a+i)+j);
}

void OutputM1(int (*a)[4], int n, int m)
{
    int i, j;
    for(i=0; i<n; i++)
        { for(j=0; j<m; j++) printf("%d\t", *(a+i)+j);
          printf("\n");
        }
}

void OutputM2(int (*b)[3], int n, int m)
{
    int i, j;
    for(i=0; i<n; i++)
        { for(j=0; j<m; j++) printf("%d\t", *(b+i)+j);
          printf("\n");
        }
}

void fun8(int (*a)[4], int (*b)[3]) //注意函数定义形式,形参是指针变量
{
```

```

    int i, j;
    for(i=0; i<4; i++)
        for(j=0; j<3; j++)
            b[i][j]=a[j][i];
}

#include <stdio.h>

void Exchg2(int &x, int &y)
{
    int tmp=x;
    x=y;
    y=tmp;
    printf("x=%d, y=%d\n", x, y);
}

void main()
{
    int a=4;
    int b=6;
    Exchg2(a, b);
    printf("a=%d, b=%d\n", a, b);
}

```

本例形参是指向二维数组的指针变量,实参是二维数组名。

/*第1章验证实验9 指向结构体的指针作为函数参数

功能:输入5名学生的学号、姓名、三门课程成绩,计算平均成绩。

输入的5名学生成绩信息如下

学号	姓名	课程一	课程二	课程三
201401	Lilin	50.0	70.0	90.0
201402	Zhaomei	70.0	80.0	90.0
201403	Liuli	60.0	70.0	80.0
201404	Zhangyan	50.0	70.0	60.0
201405	Sunjing	60.0	80.0	90.0

5名学生平均成绩计算如下

学号	姓名	平均成绩
201401	Lilin	70.0
201402	Zhaomei	80.0
201403	Liuli	70.0
201404	Zhangyan	60.0
201405	Sunjing	76.7

```

#include <stdio.h>

struct score
{
    int num;
    char name[20];
    float sc[3];
};

float average(struct score *st)
{
    int i;

```

```

float t;
t=0;
for(i=0;i<3;i++)t+=st->sc[i]; //计算三门课成绩的和
t=t/3; //计算每个人的平均成绩
return t;
}
void main( )
{
    struct score s[5]=
    {
        {201401,"Lilin",{50,70,90}},
        {201402,"Zhaomei",{70,80,90}},
        {201403,"Liuli",{60,70,80}},
        {201404,"Zhangyan",{50,70,60}},
        {201405,"Sunjing",{60,80,90}},
    };
    struct score *p;
    float avg[5];
    int i;
    p=s;
    printf("输入的 5 名学生成绩信息如下\n");
    printf("学号\t姓名\t\t\t课程一\t\t课程二\t\t课程三\n");
    for(i=0;i<5;i++)
        printf("%d\t%-18s%.1f\t%.1f\t%.1f\n",s[i].num,
            s[i].name,s[i].sc[0],s[i].sc[1],s[i].sc[2]);
    for(i=0;i<5;i++)avg[i]=average(p+i); //形参是指针,实参是地址
    printf("\n");
    printf("5 名学生平均成绩计算如下\n");
    printf("学号\t姓名\t\t\t平均成绩\n");
    for(i=0;i<5;i++)
        printf("%d\t%-18s%.1f\n", (p+i)->num,
            (p+i)->name,avg[i]);
}

```

本例形参是指向结构体数组的指针变量,实参是结构体数组的地址。

(4) 引用传递。

/*第 1 章验证实验 10 引用传递

功能:从键盘输入两个整数 a 和 b,交换 a 和 b 的值,在屏幕上显示运行结果*/

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

```
void fun10(int &px, int &py) /*注意函数定义形参为引用*/
```

```
{
```

```
    int tmp=px;
```