应用型高等学校实验教学示范系列教材

数据结构实验指导教程

毛养红　主　编
陈坚强　江　立　副主编

清华大学出版社

砼梁结构
试验指导教程

应用型高等学校实验教学示范系列教材

数据结构实验指导教程

毛养红　主　编
陈坚强　江　立　副主编

清华大学出版社
北　京

内 容 简 介

全书共两部分：基础实验和综合实验，其中实验 1 复习数据结构与算法用到的语言基础，实验 2～实验 5 是线性结构部分，实验 6～实验 12 是树与图结构部分，实验 13～实验 16 是查找与排序部分，实验 17 是综合实验部分。全书结合理论知识指导了对应的基础实验及综合实验。

本书适合作为高等院校计算机、软件工程专业高年级本科生、研究生的教材，同时可供对数据结构比较熟悉并且对数据结构有所了解的开发人员及广大科技工作者参考。

图书在版编目(CIP)数据

数据结构实验指导教程/毛养红主编. —北京：清华大学出版社，2014
应用型高等学校实验教学示范系列教材
ISBN 978-7-302-37363-6

Ⅰ. ①数… Ⅱ. ①毛… 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(2014)第 163345 号

责任编辑：张　玥　赵晓宁
封面设计：常雪影
责任校对：时翠兰
责任印制：王静怡

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：12.75　　**字　　数**：292 千字
版　　次：2014 年 10 月第 1 版　　**印　　次**：2014 年 10 月第1次印刷
印　　数：1～2500
定　　价：26.00 元

产品编号：060196-01

前言

数据结构是软件工程及计算机相关专业的核心课，是重要的专业基础课。实验是学习本课程的非常重要的环节。目前各种“数据结构”教材较为注重理论的叙述与介绍，算法描述不拘泥某种语言的语法细节，读者已具备程序设计基础，可以在课后独立完成数据结构的实验。实际上在学习程序设计的基础并不一致，相当一部分人基础较为薄弱。多数学生反映数据结构的上机实验存在一定的困难，希望有合适的实验指导教程参考学习。数据结构的理论学习也有一定的深度，存在一定的难度。学生必须完成实验的过程才能更进一步加深相关知识点的理解，同时也提高联系实际分析解决问题和编程的能力。正是基于以上的原因才编写了这本《数据结构实验指导教程》。

本指导教程的实验1～实验16为16次实验，每次实验在清楚实验目的基础上，理解实验相关理论知识，强调本次实验的要求，再具体开始实验的内容和步骤，最后给出实验参考对自己所做的实验进行自我检验。其中，每章的第5节内容是对应本章第4节的相关实验步骤或要求，给出了实验的参考答案或参考代码。实验17为综合实验部分，其中综合实验分为阶段性知识点综合实验和本课程及相关课程关联知识的综合实验。学完阶段性相关知识点就可以开始做个相应的综合实验来加强所学知识的综合应用。本指导书综合案例的安排：线性表、栈、队列，一个提供综合理解学习，另一个自己尝试完成；树与二叉树相关知识提供一个综合实验理解学习，提供一个自己尝试完成；图相关知识提供综合实验理解，提供一个综合案例自己尝试完成；排序、查找、结合线性表提供一个综合案例自己尝试完成。实验1～实验5和实验17由毛养红编写，实验6～实验11由陈坚强编写，实验12～实验16由江立编写。由于作者水平有限，不足之处在所难免，敬请读者批评指正。

编　者

2014年8月

目录

▶基础实验

▶综合实验

基础实验

实验 1　程序设计基础与抽象数据类型

1.1　实 验 目 的

- 复习函数的使用；
- 复习结构体、指针的使用；
- 掌握抽象数据类型的设计方法及实现；
- 练习用 Visual C++ 6.0 或 Visual Studio 2010 以上版本开发环境。

1.2　实 验 原 理

1.2.1　结构体

1. 结构体类型的定义

用户建立由不同类型数据组成组合型的数据结构，称为结构体(structure)。

格式：

```
struct  结构体类型名
{
    数据类型   成员 1;
    数据类型   成员 2;
       ⋮
    数据类型   成员 n;
};
```

日期结构体类型 Date 定义：

```
struct Date
{
    int  year;
    int  month;
    int  day;
};
```

2. 结构体变量的定义

(1) 先声明结构体类型,再定义该类型变量。

例如:

```
struct Student
{
    int num;
    char name[20];
    char sex;
    int age;
    float score;
    char addr[30];
};
struct Student student1,student2;
```

student1 和 student2 结构体变量每个数据项所占的字节个数如图 1-1 所示。

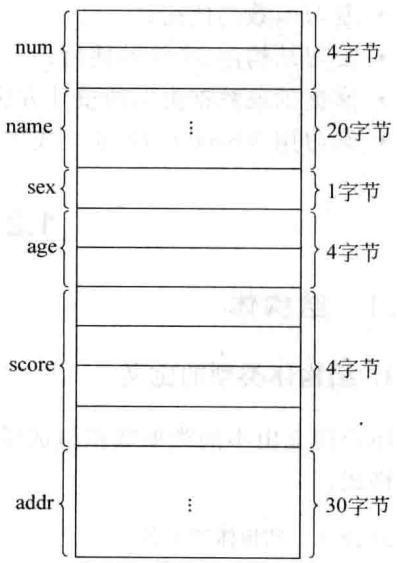

图 1-1 student1 和 student2 结构体变量每个数据项所占的字节个数

(2) 在声明类型的同时定义变量。

```
struct  结构体名
{
    成员表列
} 变量名表列;
```

例如:

```
struct Student
{
    int num;
    char name[20];
    char sex;
    int age;
    float score;
    char addr[30];
} student1,student2;
```

(3) 用 typedef 先把类型名称定义为别名,再用别名来定义变量。

```
typedef struct  结构体名
{
    成员表列
} 结构体类型名的别名;
```

例如:

```
typedef struct Student
```

```
{
    char num[11];                    //学生学号
    char name[20];                   //学生姓名
    char sex;                        //性别
    int age;                         //年龄
    float score;                     //成绩
    char addr[30];                   //地址
}STU;
STU student1,student2;
```

3. 结构体变量的初始化和引用

【例 1-1】 把一个学生的信息(包括学号、姓名、性别、住址)放在一个结构体变量中,然后输出这个学生的信息。

解题思路:建立一个结构体类型,包括有关学生信息的各成员,用它定义结构体变量,同时赋以初值,最后输出该结构体变量的各成员。

```
#include<iostream>
using namespace std;
int main(void)
{
    struct Student
    {
        char num[11];                //学生学号
        char name[20];               //学生姓名
        char sex;                    //性别
        char addr[30];               //地址
    }stu={"10101","Li Lin",'M',"123 Beijing Road"};      //变量 stu 初始化
    cout<<"NO.:<<stu.num<<endl;
    cout<<"name: "<<stu.nam<<endl;
    cout<<"sex: "<<stu.sex<<endl;
    cout<<"address: "<<stu.addr<<endl;
    return 0;
}
```

4. 结构体数组

结构体数组是数组中的每一个元素都是结构体变量。

结构体数组定义的一般形式:

格式 1:

```
struct 结构体名
{
    成员表列
} 数组名[数组长度];
```

格式 2：先声明一个结构体类型，然后再用此类型定义结构体数组。

```
结构体类型  数组名[数组长度];
```

例如：

```
struct Person leader[3];
```

【例 1-2】 有 n 个学生的信息(包括学号、姓名、成绩)，要求按照成绩的高低顺序输出各学生的信息。

解题思路：用结构体数组存放 n 个学生信息，采用选择法对各元素进行排序(进行比较的是各元素中的成绩)。

```
#include<iostream>
using namespace std;
struct Student                    //声明结构体类型 struct Student
{
    int num;
    char name[20];
    float score;
};
int main(void)
{
    struct Student stu[5]={{10101,"Zhang",78},{10103,"Wang",98.5},{10106,"Li",86},
    {10108,"Ling",73.5},{10110,"Fun",100}};     //定义结构体数组并初始化
    struct student temp;     //定义结构体变量 temp,用作交换时的临时变量
    const int n=5;
    int i,j,k;
    cout<<"The order is:"<<endl;
for(i=0;i<n-1;i++)
    {
        k=i;
        for(j=i+1;j<n;j++)
            if(stu[j].score>stu[k].score)     //进行成绩的比较
                k=j;
        if(k!=i)                              //stu[k]和 stu[i]元素互换
        {
            temp=stu[k];
            stu[k]=stu[i];
            stu[i]=temp;
        }
    }
    for(i=0;i<n;i++)
        cout<<stu[i].num<<"  "<<stu[i].name<<"  "<<stu[i].score<<endl;
    return 0;
}
```

1.2.2　指针

1. 概念与定义格式

概念：用来存放一个数据(对象)的地址的变量。指针变量的值是地址(即指针)。

定义格式：

```
数据类型 *　变量名 [=指针表达式];
```

说明：

(1) * 是指针类型的标志，不能省略；它不是指针变量名的组成部分。

(2) 数据类型是指针变量中所存地址指向的数据的类型，又称基类型，决定了数据占用的存储单元长度及存储方式，不同数据类型占用的存储单元长度可能不同，即使长度相同，存储方式也不同；数据类型是通过指针存取数据的基础，不允许定义无数据类型的指针变量。

(3) 允许对指针变量初始化，即定义时并赋地址值。

(4) 指针的两个特殊运算符(单目运算符)。

① &：取地址运算符，返回其操作对象的内存地址，通常其操作对象为一个变量，如“p_a=&a;”。

② *：间接访问运算符，存取指针所指单元的值，如“cout<< * p_a;”，可以输出变量 a 的值。

【例 1-3】 运算符 & 与 * 使用。

```
#include<iostream>
using namespace std;
int main(void)
{
    int a=10;
    int * p_a;
    p_a=&a;                    //取地址
    cout<<"变量 a 的地址: "<<p_a<<endl;
    cout<<"修改前: 直接访问 a="<<a<<", 间接访问 a="<< * p_a<<endl;
    * p_a=10000;               //通过间接访问,修改 a 值
    cout<<"修改后: 直接访问 a="<<a<<", 间接访问 a="<< * p_a<<endl
    return 0;
}
```

【例 1-4】 输入 a 和 b 两个整数，按先大后小的顺序输出 a 和 b。

解题思路：用指针方法来处理这个问题。不交换整型变量的值，而是交换两个指针变量的值。

```
#include<iostream>
using namespace std;
```

```
int main(void)
{
    int *p1,*p2,*p,a,b;
    printf("please enter two integer numbers:");
    scanf("%d %d",&a,&b);                    //输入两个整数
    p1=&a;                                   //使 p1 指向变量 a
    p2=&b;                                   //使 p2 指向变量 b
    if(a<b)                                  //如果 a<b
    {
        p=p1;p1=p2;p2=p;
    }                                        //p1 与 p2 的值互换
    cout<<"a="<<a<<",b="<<b<<endl);          //输出 a,b
    cout<<"max="<<*p1<<",min="<<*p2;         //输出 p1 和 p2 所指向的变量的值
    return 0;
}
```

2. 指针变量作为函数参数

【例 1-5】 题目要求同例 1-4，即对输入的两个整数按大小顺序输出。现用函数处理，而且用指针类型的数据作函数参数。

解题思路：定义一个函数 swap，将指向两个整型变量的指针变量作为实参传递给 swap 函数的形参指针变量，在函数中通过指针实现交换两个变量的值。

```
#include<iostream>
using namespace std;
int main(void)
{
    void swap(int *p1,int *p2);
    int a,b;
    int *pointer_1,*pointer_2;
    cout<<"please enter a and b:"<<endl;
    cin>>a>>b;
    pointer_1=&a;
    pointer_2=&b;
    if(a<b)
        swap(pointer_1,pointer_2);
    cout<<"max="<<a<<", min="<<b<<endl;
    return 0;
}
void swap(int *p1,int *p2)
{
    int temp;
    temp=*p1;
```