# 第 3 章　用数组和指针实现公司职员的信息管理程序

## 3.1　系 统 概 要

数组和指针是 C 和 C++ 语言的基础知识，是我们必须要掌握的知识。本章的实例设计了一个公司职员的信息管理程序，用数组和指针来实现。

本程序实现的功能是增加职员信息、查找职员信息、浏览所有的职员信息、删除职员信息。功能菜单有如下内容：(1)增加职员信息，(2)查找职员信息，(3)浏览职员信息，(4)删除职员信息。

程序的功能很简单，功能结构图如图 3.1 所示。

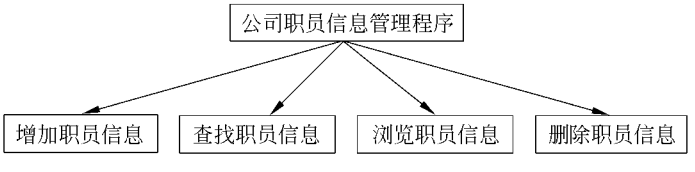


图　3.1

## 3.2　基本功能要求

程序要求不用链表，用数组和指针来实现。设计要求主要如下。

(1) 功能选择可以用菜单来实现。用户根据自己的不同选择进入不同的菜单。

(2) 程序不要求将信息保存到文件中，程序开始时数组内容为空，用户需要先增加职员信息，之后才可以实现查找、删除等功能。

(3) 职员信息的关键字是职员的职员号，加入时职员号重复的记录不能加入。查找、删除按照职员号进行操作。

## 3.3 主要知识点

程序主要用数组和指针来实现职工信息的添加、查找、删除和显示。实质就是在数组中增加元素、查找元素、删除元素和显示数组中的所有元素。数组是由类型相同、逻辑意义相关的一组数据构成的。每个数组都有一个由标识符来充当的名字，也称为数组名。数组中的每个元素叫做数组元素，它们按顺序分配在内存中一片连续的内存单元中，每个元素占用相同数目的单元。数组元素是按顺序排列的，顺序号叫做数组元素的下标。数组元素通过数组名和下标来表示。数组可以有多个下标，下标的个数叫数组的维数。

无论是在C语言，还是在C++语言中，指针都极其重要。指针变量就是存放另一变量地址的变量。它也占用存储空间，也可以进行运算。

## 3.4 系统设计思路与算法

### 3.4.1 关键技术

下面分析程序中主要的数据结构和函数的实现。

(1) 数据结构

程序中设计了两个类 Staff 和 Company。定义如下。

① Staff 类的定义

```
class Staff
{
public:
    char name[10];              //职员姓名
    char no[5];                 //职员号
    char department[10];        //职员所在部门
    int money;                  //职员工资
    char work[10];              //职员职位

    Staff();
    Staff(char * name,char * no,char * dep,int money,char * work);
    ~Staff();
};
```

Staff 类主要的成员变量是职员的一些信息。职员的主要信息由类 Staff 表示。

② Company 类的定义

```
class Company
{
public:
    int count;                  //公司职员的人数
```

```
    int add[30];              //整型数组,数组中的元素是职员的地址

    Staff * Sta;
    Company();
    ~Company();
    bool AddStaff(char * name,char * no,char * dep,int money,char * work);
    bool DeleteStaff(char * no);
    bool FindStaff(char * no);
    void DispAll();
};
```

Company 类主要的成员变量 count 用来表示职员数，int add[30]的每一个元素用来表示每一个职员的信息的地址。主要的成员函数 AddStaff(char * name,char * no,char * dep,int money,char * work)实现增加职员的功能；DeleteStaff(char * no)实现删除职员的功能；FindStaff(char * no)实现查找职员信息的功能；DispAll()实现显示所有职员信息的功能。

下面介绍类 Company 成员函数的实现。

(2) 增加职员信息函数 AddStaff(char * name,char * no,char * dep,int money,char * work)的实现

函数设计的主要思想的实质就是往一维数组中添加元素。

① 函数中 ptr= new Staff(name,no,dep,money,work)用来新建了一个 Staff，并令指针 ptr 指向它。

② 查找数组中现有元素，是否添加的职工号在数组中已经存在，若存在则不能添加。

③ 如果要添加的元素在数组中不存在，则追加为数组的最后一个元素。需要注意的是 add[count]=(int)ptr 使得数组 add[ ]中的内容是 Staff 的指针。也就是说，一维数组 add[ ]中的每一个元素其实是职员类的每一个实例的指针，也就是每一个职员的信息的地址。

(3) 显示所有职员信息 DispAll()的实现

函数的实质就是一位数组的遍历。

数组的每一个元素是职员信息的地址。

函数中 Sta=(Staff * )add[i]将数组的内容转化为 Staff 指针。用 for 循环依次输出数组中 Staff 指针所指向的所有对象。

(4) 删除职员信息 DeleteStaff(char * no)的实现

函数的实质就是从一维数组中删除元素。

① 函数中需要先找到要删除的职工在数组中的位置，将位置用 j 保存。

② 删除时只需要把 j 到 count 的所有元素前移一位即可，即 add[j]= add[j+1]。

③ 如果查找了数组中所有的元素，没有找到要删除的元素，则表明没有这个职员。

(5) 查找职员信息 FindStaff(char * no)的实现

函数的实质是一维数组的查找。

从数组的第一个元素开始查找，如果找到，则显示该职员的所有信息；如果没有找到，

则表明没有这个职员。

### 3.4.2 运行结果

程序运行时，首先出现菜单，由用户进行功能选择，选择不同的数字，进入不同的功能区，如图 3.2 所示。

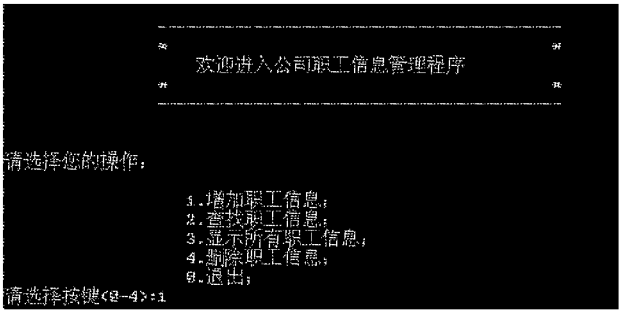


图 3.2

增加客户信息时，输入客户信息，如图 3.3 所示。

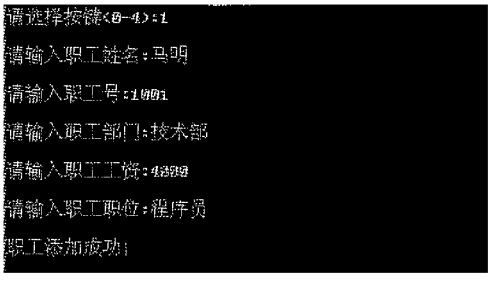


图 3.3

显示所有职工信息时，如图 3.4 所示。

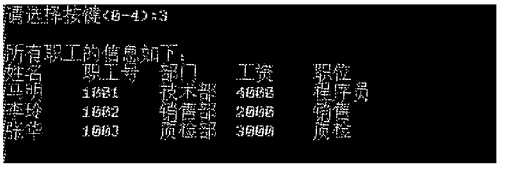


图 3.4

查找某个职员的信息时，需要输入职工号，如图 3.5 所示。

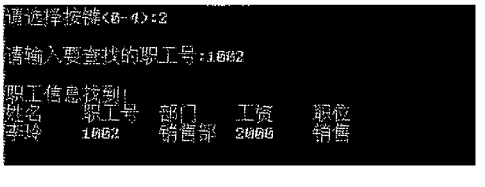


图 3.5

删除时和查找一样，这里略去。

退出系统时，如图 3.6 所示。


图　3.6

## 3.5　源程序代码

```
/*****************************************************************************/
    程序由两个文件组成：cpp3.h 和 cpp3.cpp。
    1. cpp3.h 文件
    #include <stdlib.h>
    #include <stdio.h>
    #include <string.h>
    #include <iostream.h>
/*****************************************************************************/
    /*
    类名称：Staff
    功能描述：定义类 Staff
    修改记录：
    */
    class Staff
    {
    public:
        char name[10];
        char no[5];
        char department[10];
        int money;
        char work[10];

        Staff();
        Staff(char * name,char * no,char * dep,int money,char * work);
        ~Staff();
    };
```

```
/*****************************************************************************/
    /*
    类名称：Company
    功能描述：定义类 Company
    修改记录：
    */
    class Company
    {
    public:
        int count;
        int add[30];

        Staff * Sta;

        Company();
        ~Company();
        bool AddStaff(char * name,char * no,char * dep,int money,char * work);
        bool DeleteStaff(char * no);
        bool FindStaff(char * no);
        void DispAll();
    };
/*****************************************************************************/
```

2. cpp3.cpp 文件

```
    #include "cpp3.h"
    /*
    函数名称：Staff()
    功能描述：类 Staff 的构造函数
    修改记录：
    */
    Staff::Staff()
    {
        name[0]='0';
        no[0]='0';
        department[0]='0';
        money=0;
        work[0]='0';
    }
/*****************************************************************************/
    /*
    函数名称：Staff(char * name1,char * no1,char * dep1,int money1,char * work1)
    功能描述：类 Staff 的带参数的构造函数
    修改记录：
    */
    Staff::Staff(char * name1,char * no1,char * dep1,int money1,char * work1)
```

```
    {
        strcpy(name,name1);
        strcpy(no,no1);
        strcpy(department,dep1);
        money=money1;
        strcpy(work,work1);
    }
/*****************************************************************************/
    /*
    函数名称：Staff::~Staff()
    功能描述：类 Staff 的析构函数
    修改记录：
    */
    Staff::~Staff()
    {
    }
/*****************************************************************************/
    /*
    函数名称：Company()
    功能描述：类 Company 的构造函数
    修改记录：
    */
    Company::Company()
    {
        count=0;
        for(int i=0;i<30;i++)
        {
            add[i]=0;
        }
    }
/*****************************************************************************/
    /*
    函数名称：Company::~Company()
    功能描述：类 Company 的析构函数
    修改记录：
    */
    Company::~Company()
    {delete[] Sta;}
/*****************************************************************************/
    /*
    函数名称：AddStaff(char * name,char * no,char * dep,int money,char * work)
    功能描述：增加职工信息
    修改记录：
    */
```

```
bool Company::AddStaff(char * name,char * no,char * dep,int money,char * work)
{
    Staff * ptr;
    ptr=new Staff(name,no,dep,money,work);
    for(int i=0;i<count;i++)
    {
        Sta=(Staff * )add[i];
        if(strcmp(Sta->no,no)==0)
        {
            cout<<"这个职工已经存在,无法加入!"<<endl;
            return false;
        }
    }
    if(count<30 && i==count)
    {
        add[count]=(int)ptr;
        count++;
        cout<<"职工添加成功!"<<endl;
        return true;
    }
    else
        return false;
}
```

此函数的效果可以参考图 3.3。

```
/*****************************************************************************/
    /*
    函数名称:DispAll()
    功能描述:显示所有的职工信息
    修改记录:
    */
    void Company::DispAll()
    {
        cout<<"姓名"<<"\t 职工号"<<"\t 部门"<<"\t 工资"<<"\t 职位"<<endl;
        for(int i=0;i<count;i++)
        {
            Sta=(Staff * )add[i];
            cout<<Sta->name <<'\t'<<Sta->no <<'\t'<<Sta->department <<'\t'<<
Sta->money <<'\t'<<Sta->work <<endl;
        }
    }
```

此函数的效果可以参考图 3.4。

```
/*****************************************************************************/
    /*
    函数名称：DeleteStaff(char * no)
    功能描述：删除某个职工的信息
    修改记录：
    */
    bool Company::DeleteStaff(char * no)
    {
        for(int i=0;i<count;i++)
        {
            Sta=(Staff * )add[i];
            if(strcmp(Sta->no,no)==0)        //找到要删除的结点
            {
                int j=i;
                for(j;j<count;j++)
                {
                    add[j]=add[j+1];
                }
                cout<<"这个职工的信息已经删除!"<<endl;
                count--;
                return true;
            }
        }
        if(i==count)
        {
            cout<<"这个职工没有找到,无法删除!"<<endl;
            return false;
        }
        return false;
    }
/*****************************************************************************/
    /*
    函数名称：FindStaff(char * no)
    功能描述：查找某个职工的信息
    修改记录：
    */
    bool Company::FindStaff(char * no)
    {
        for(int i=0;i<count;i++)
        {
            Sta=(Staff * )add[i];
            if(strcmp(Sta->no,no)==0)
            {
                cout<<"职工信息找到!"<<endl;
```

```
                cout<<"姓名"<<"\t职工号"<<"\t部门"<<"\t工资"<<"\t职位"<<endl;
                cout<<Sta->name  <<'\t'<<Sta->no  <<'\t'<<Sta->department <<
'\t'<<Sta->money <<'\t'<<Sta->work <<endl;
                return true;
            }
        }
        if(i==count)
        {
            cout<<"职工信息没有找到!"<<endl;
            return false;
        }
        return false;
    }
```

此函数的效果可以参考图 3.5。

```
/*****************************************************************************/
    /*
    函数名称:main()
    功能描述:主函数
    修改记录:
    */
    void main()
    {
        Company shiyou;
        int sel;
        sel=1;
        char name[10];
        char no[5];
        char department[10];
        int money;
        char work[10];
        cout<<endl<<endl;
        cout<<"        ------------------------------------"<<endl;
        cout<<"        *                                   *"<<endl;
        cout<<"            欢迎进入公司职工信息管理程序     "<<endl;
        cout<<"        *                                   *"<<endl;
        cout<<"        ------------------------------------"<<endl;
        while(sel)
        {

            cout<<endl<<endl;
            cout<<"请选择您的操作:"<<endl<<endl;
            cout<<"                    1.增加职工信息;"<<endl;
            cout<<"                    2.查找职工信息;"<<endl;
```