|  |  |  |  |
| --- | --- | --- | --- |
| **学号** | 2106412030 | **成绩** |  |





**课程设计说明书**

**设计名称 C++课程设计**

**设计题目 职工管理系统**

**设计时间 2022.6.6—2022.6.10**

**学 院 信息与控制工程学院**

**专 业 计算机科学与技术**

**班 级 2102**

**姓 名 潘钊拓**

**指导教师 宋晓宇**

**2022 年 6 月 6 日**

C++课程设计说明书

**目录**

1. 设计任务说明

2. 开发工具

3. 设计思路

4. 系统效果图（每个图要有文字说明，可附多图。）

5. 源代码

6. 参考文献

7. 设计体会

**一、设计任务说明**

**职工档案管理系统设计**

**功能：职工档案管理系统设计，每个职工是一条记录，包括编号、姓名、性别、出生年**

**月、所在部门、职称、工资级别、电话等。**

**系统可实现以下功能：**

**1、输入功能：输入每一位职工记录，将其信息存入文件中，要求有文件读写操作；**

**2、显示功能：完成全部职工记录的显示；**

**3、查找功能：完成按编号或姓名查找职工的相关记录，并显示；**

**4、信息删除与修改——输入编号，修改或删除该职工的信息。**

**二、开发工具**

Microsoft Visual Studio 2022

**三、设计思路**

**1、菜单界面**

**使用cout和cin输入输出函数设计出主界面交互菜单，通过switch case语句判断输入的选项值，执行相对应得子函数。**

**用户推出菜单选项之后，需要使用C++语言文件操作，保存已经录入的全部员工信息和工资。程序下次启动时可以直接查询上次保存的数据**

**2、添加员工信息**

**设计为单独的子函数供主函数调用。**

**定义一个结构体类型，包括多个成员变量（编号，姓名，性别，年龄，工资等），并用该类型定义一个全局的结构体数组，供外部录入员工信息时存储。**

**定义一个全局变量累存储当前员工的个数。**

**添加员工信息时，需要从外部读入员工的编号，姓名，性别，等信息，员工数量加一，并根据当前员工个数将其保存在结构体数组对应序号的元素中。**

**在录入员工信息时可以选择录入人数来实现一次录入多名人员信息。**

**3、显示全部员工的信息**

**通过对当前结构体数组进行遍历，并显示输出**

**4、查找单个员工**

**根据输入需要查找员工的编号或姓名，遍历当前结构体数组，判断是否含有一致编号(姓名)的元素，若有则显示输出对应的员工信息，否则显示无此人。**

**若有多名员工姓名相同 则所有该姓名的员工的信息均会被显示出来。**

**5、删除单个员工**

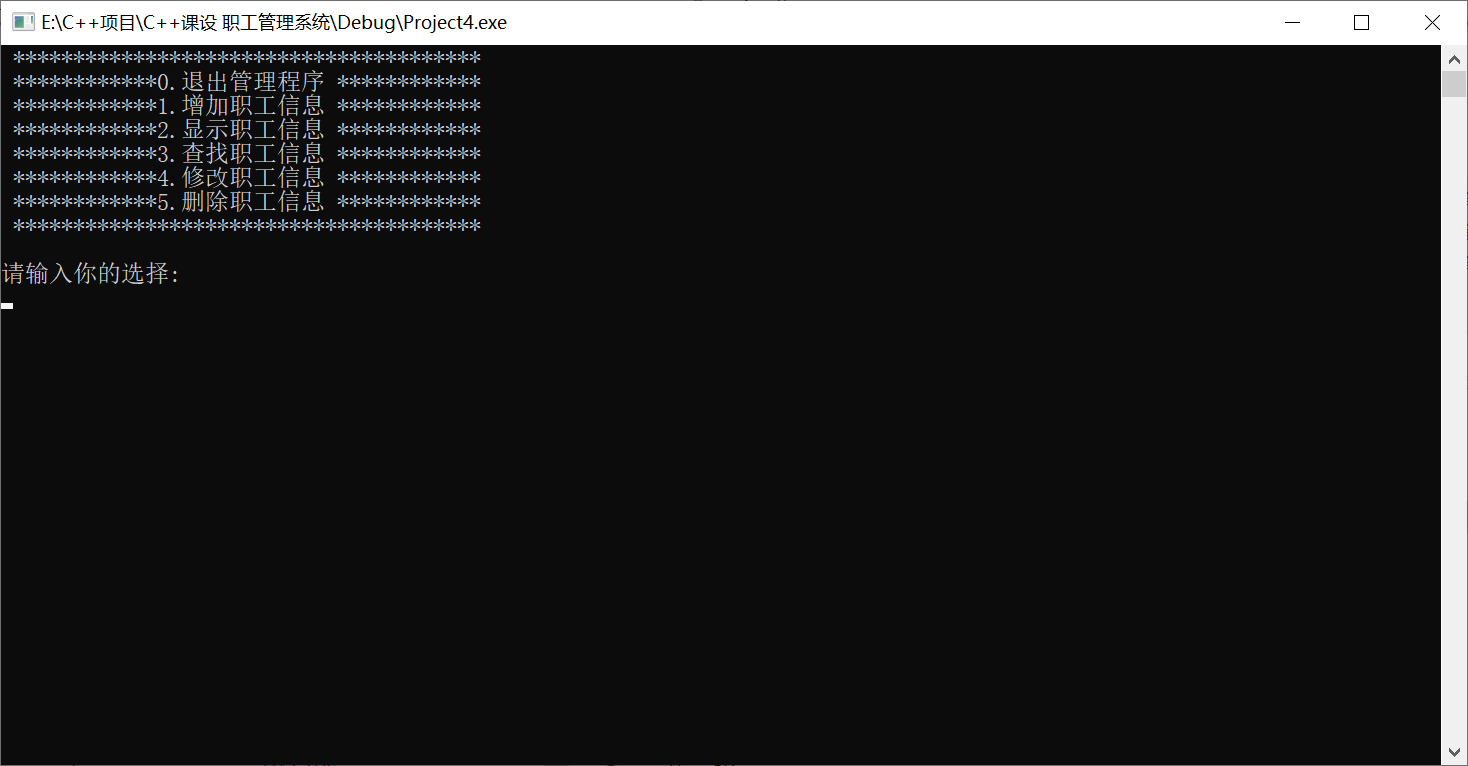
**输入需要删除员工的编号，遍历当前结构体数组，判断是否含有一致编号的元素，若有则从该元素之后依次将下一个元素赋值到上一个元素，直至最后一个赋值完成，最后当前员工个数变量减1。**

**6、修改员工信息**

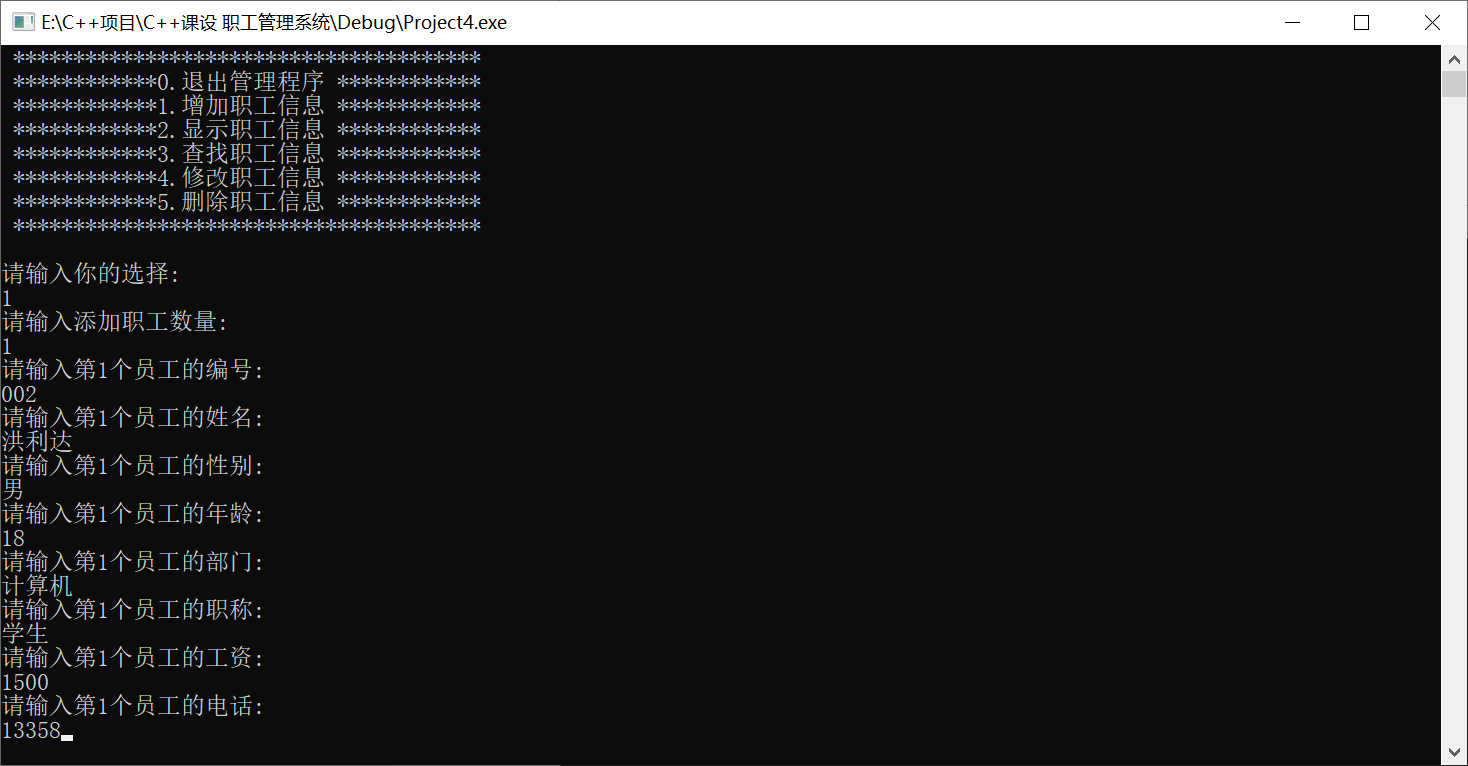
**输入需要修改学员工的编号，遍历当前结构体数组，判断是否含有一致编号的元素，若有则依次从外部读入用户所修改的值，保存到对应成员变量中。**

**四、系统效果图**

**1、运行界面**



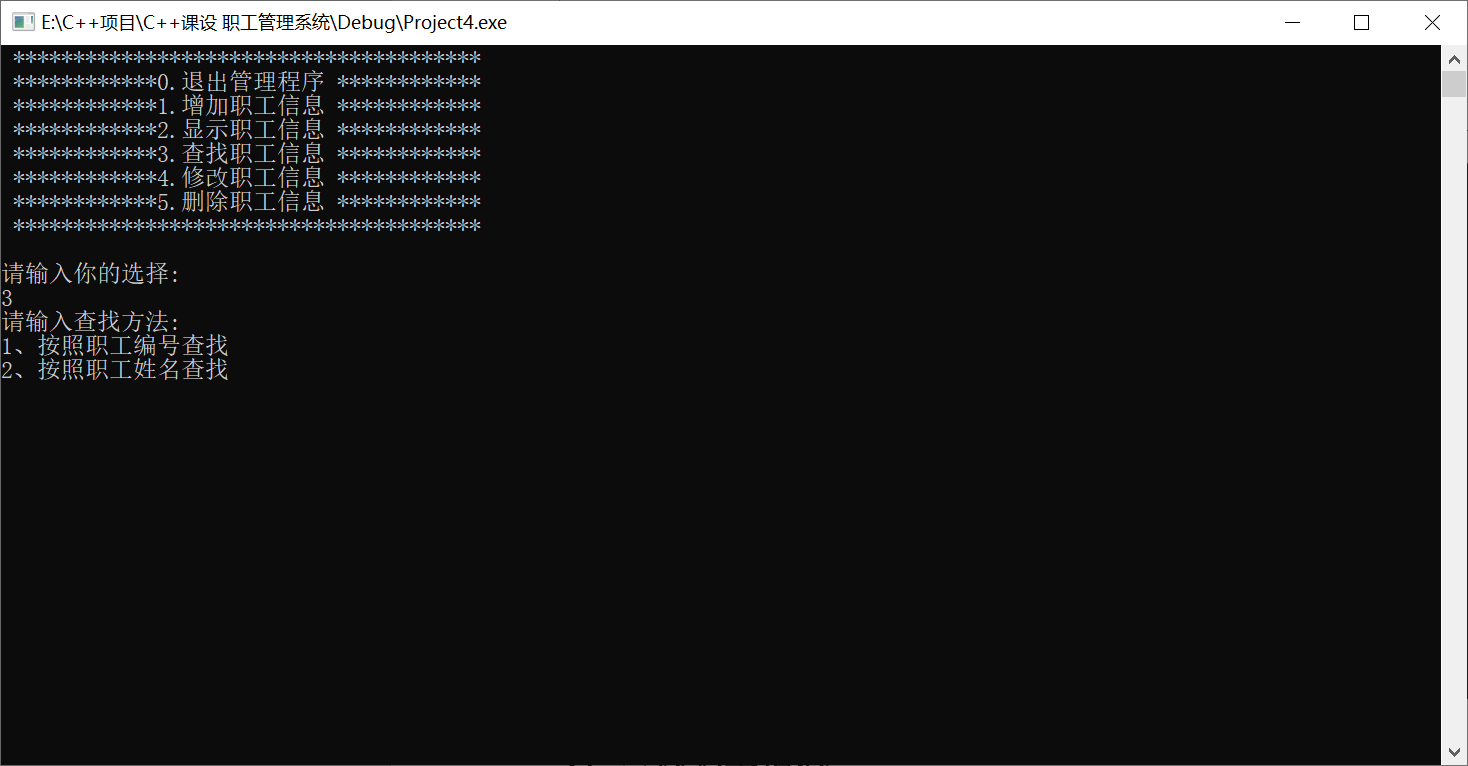
**2、增加员工信息界面**



**3、显示职工信息界面**

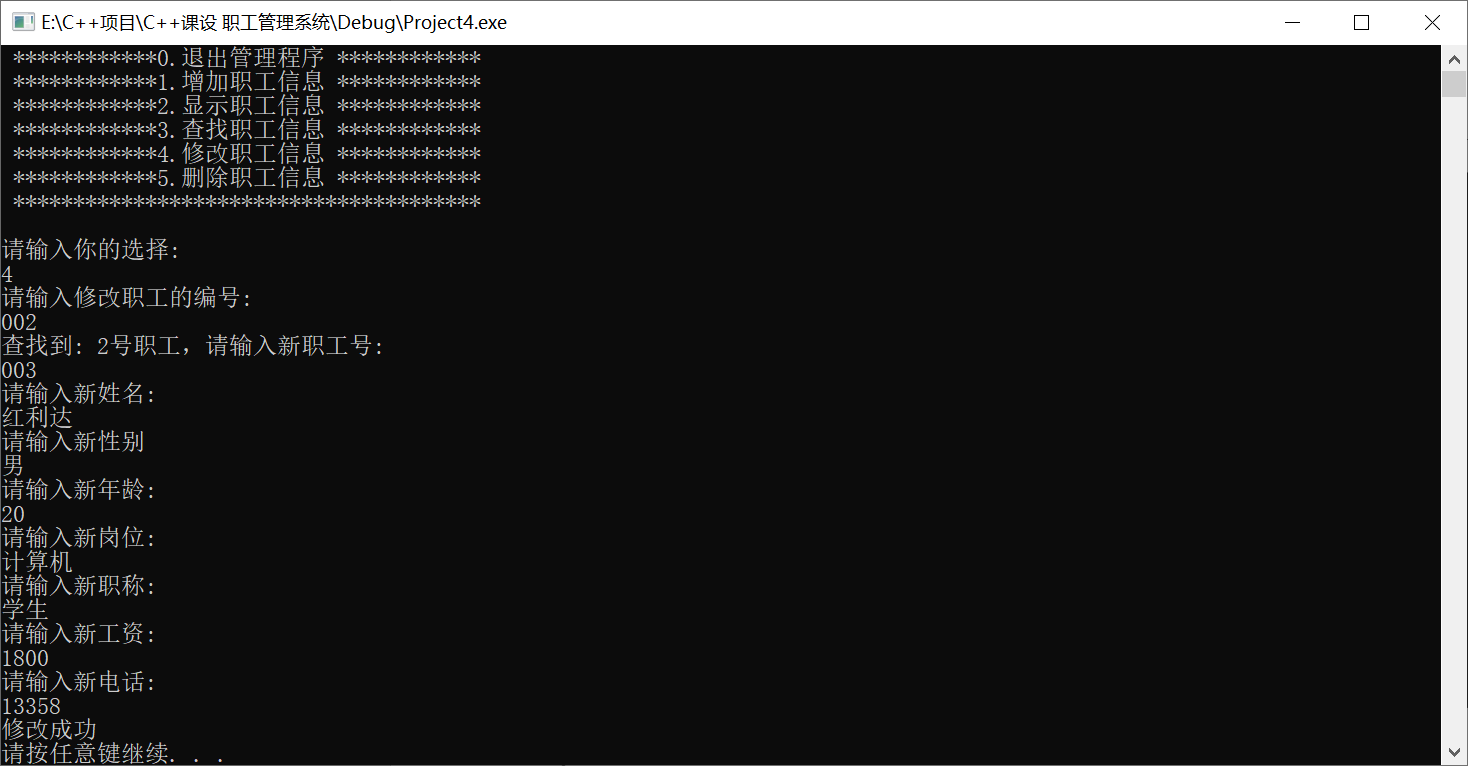


**4、查找职工信息**

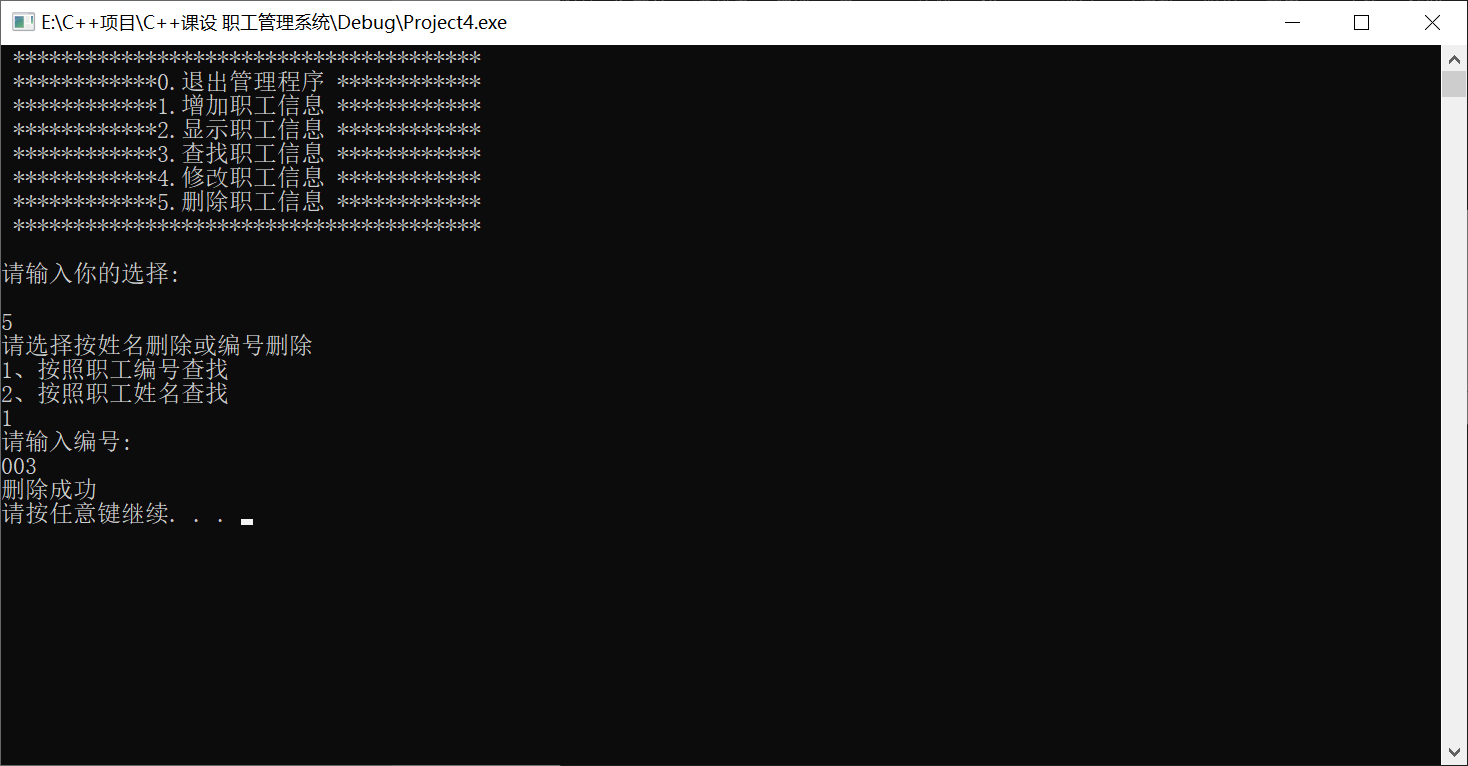




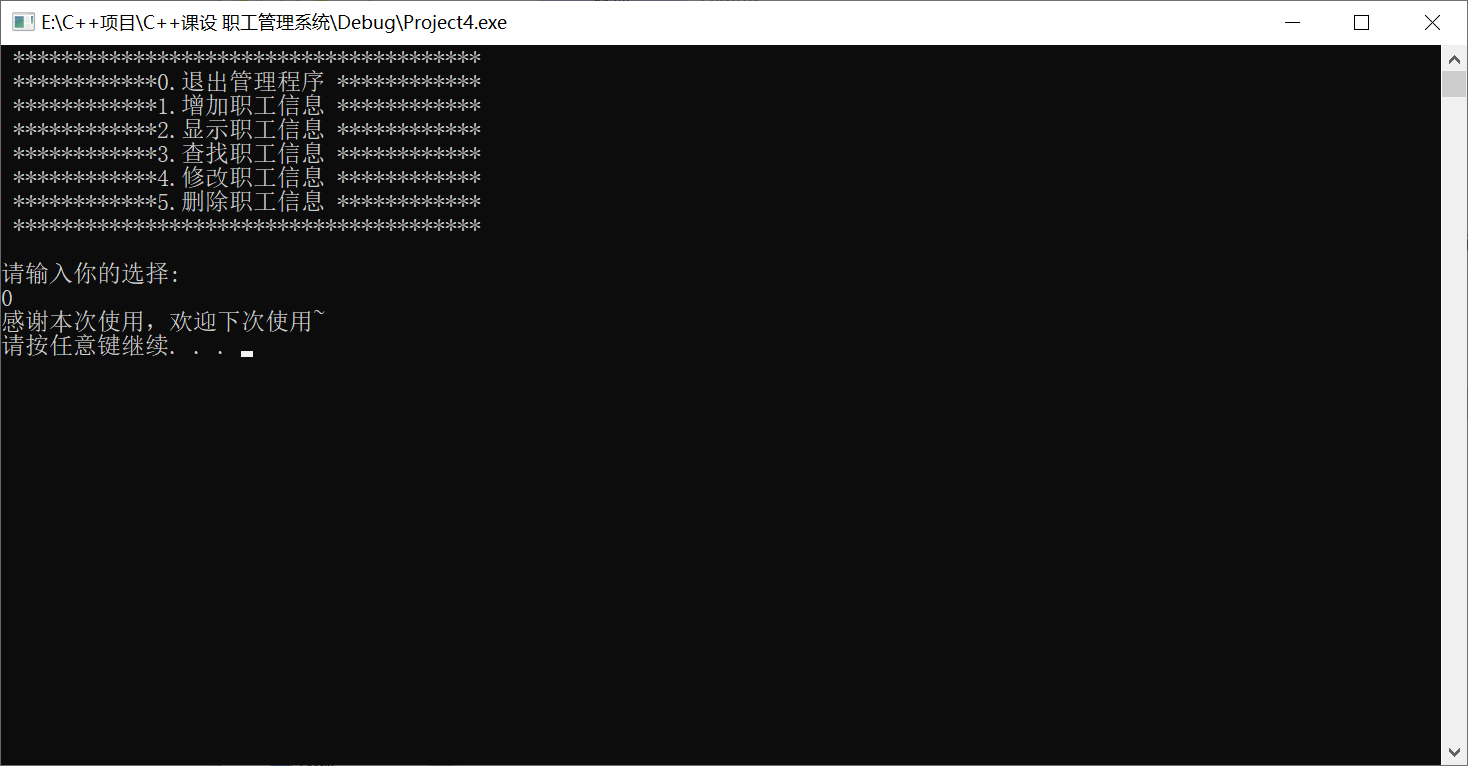
**4、修改职工信息界面**



**5、删除界面**



**6、退出界面**



**五、源代码**

**WorkerManager.h**

#pragma once//防止头文件重复包含

#include<iostream>

#include<fstream>

using namespace std;

#include"Worker.h"

#define FILENAME "empFile.txt"

class WorkerManager

{

public:

WorkerManager();

//菜单系统

void Menu();

//退出系统

void Exit();

//记录职工人数

int Number;

//职工数组指针

worker \*\* m\_Emparray;

//添加职工

void Add\_Emp();

//保存至文件中

void save();

//判断文件是否为空

bool m\_FileIsEmpty;

//统计文件中人数

int get\_EmpNum();

//初始化员工

void init\_Emp();

//显示职工

void show\_Emp();

//删除职工

void Del\_Emp();

//判断职工是否存在

int IsExist1(int id);

int IsExist2(string name);

//修改职工

void mod\_emp();

//查找职工

void find\_emp();

//析构函数

~WorkerManager();

};

**WorkerManager.cpp**

#include"WorkerManager.h"

using namespace std;

WorkerManager::WorkerManager()

{

//文件未创建

ifstream ifs;

ifs.open(FILENAME, ios::in);

if (!ifs.is\_open())

{

this->Number = 0;

this->m\_Emparray = NULL;

//初始化文件是否为空

this->m\_FileIsEmpty = true;

ifs.close();

return;

}

//文件存在 数据为空

char ch;

ifs >> ch;

if (ifs.eof())

{

//文件为空

this->Number = 0;

this->m\_Emparray = NULL;

//初始化文件是否为空

this->m\_FileIsEmpty = true;

ifs.close();

return;

}

//文件存在，内容也存在

int num = this->get\_EmpNum();

this->Number = num;

//将文件中数据存入到数组中

this->m\_Emparray = new worker \* [this->Number];

this->init\_Emp();

}

void WorkerManager:: Menu()

{

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<< endl;

cout << " \*\*\*\*\*\*\*\*\*\*\*\*0.退出管理程序 \*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\*\*\*\*\*\*\*\*\*\*\*1.增加职工信息 \*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\*\*\*\*\*\*\*\*\*\*\*2.显示职工信息 \*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\*\*\*\*\*\*\*\*\*\*\*3.查找职工信息 \*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\*\*\*\*\*\*\*\*\*\*\*4.修改职工信息 \*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\*\*\*\*\*\*\*\*\*\*\*5.删除职工信息 \*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << endl;

}

void WorkerManager::Exit()

{

cout << "感谢本次使用，欢迎下次使用~" << endl;

system("pause");

exit(0);//退出程序

}

void WorkerManager::Add\_Emp()

{

cout << "请输入添加职工数量: " << endl;

int addNum = 0;

cin >> addNum;

if (addNum > 0)

{

int newSize = this-> Number + addNum;//新空间人数

//开辟新空间

worker \*\* newSpace= new worker\* [newSize];

//旧空间拷贝到新空间

if (this->m\_Emparray != NULL)

{

for (int i = 0; i < this->Number; i++)

{

newSpace[i] = this->m\_Emparray[i];

}

}

//批量添加新数据

for (int i = 0; i < addNum; i++)

{

int id;

string name;

string xingbie;

int age;

string add;

string zhicheng;

int gongzi;

int dianhua;

cout << "请输入第" << i + 1 << "个员工的编号: " << endl;

cin >> id;

cout << "请输入第" << i + 1 << "个员工的姓名: " << endl;

cin >> name;

cout << "请输入第" << i + 1 << "个员工的性别: " << endl;

cin >> xingbie;

cout << "请输入第" << i + 1 << "个员工的年龄: " << endl;

cin >> age;

cout << "请输入第" << i + 1 << "个员工的部门: " << endl;

cin >> add;

cout << "请输入第" << i + 1 << "个员工的职称: " << endl;

cin >> zhicheng;

cout << "请输入第" << i + 1 << "个员工的工资: " << endl;

cin >> gongzi;

cout << "请输入第" << i + 1 << "个员工的电话: " << endl;

cin >> dianhua;

worker\* Str = NULL;

Str = new worker(id, name, xingbie, age, add, zhicheng, gongzi, dianhua);

//将创建员工保存到数组中

newSpace[Number + i] = Str;

}

//释放旧空间

delete[]this->m\_Emparray;

//更改空间指向

this->m\_Emparray = newSpace;

//更新新的职工人数

this->Number = newSize;

cout << "添加成功" << endl;

//职工文件内不为空标志

this->m\_FileIsEmpty = false;

//保存数据到文件中

this->save();

}

else

{

cout << "输入数据有误" << endl;

}

system("pause");

system("cls");

}

void WorkerManager:: save()

{

ofstream ofs;

ofs.open(FILENAME, ios::out);//写文件

for (int i = 0; i < this->Number; i++)

{

ofs << this->m\_Emparray[i]->m\_id << " "

<< this->m\_Emparray[i]->m\_name << " "

<< this->m\_Emparray[i]->m\_xingbie << " "

<< this->m\_Emparray[i]->m\_age << " "

<< this->m\_Emparray[i]->m\_add << " "

<< this->m\_Emparray[i]->m\_zhicheng << " "

<< this->m\_Emparray[i]->m\_gongzi << " "

<< this->m\_Emparray[i]->m\_dianhua << endl;

}

ofs.close();

}

int WorkerManager::get\_EmpNum()

{

ifstream ifs;

ifs.open(FILENAME, ios::in);//读文件

int id;

string name;

string xingbie;

int age;

string add;

string zhicheng;

int gongzi;

int dianhua;

int num = 0;

while (ifs >> id && ifs >> name && ifs >> xingbie && ifs >> age &&

ifs >> add && ifs >> zhicheng && ifs >> gongzi && ifs >> dianhua)

{

//统计人数

num++;

}

return num;

}

void WorkerManager::init\_Emp()

{

ifstream ifs;

ifs.open(FILENAME, ios::in);

int id;

string name;

string xingbie;

int age;

string add;

string zhicheng;

int gongzi;

int dianhua;

int temp = 0;

while (ifs >> id && ifs >> name && ifs >> xingbie && ifs >> age &&

ifs >> add && ifs >> zhicheng && ifs >> gongzi && ifs >> dianhua)

{

worker\* emp = NULL;

emp = new worker(id, name, xingbie, age, add, zhicheng, gongzi, dianhua);

this->m\_Emparray[temp] = emp;

temp++;

}

//关闭文件

ifs.close();

}

void WorkerManager::show\_Emp()

{

//判断文件是否为空

if (m\_FileIsEmpty)

{

cout << "文件不存在或为空" << endl;

}

else

{

for (int i=0; i < Number; i++)

{

this->m\_Emparray[i]->showInfo();

}

}

system("pause");

system("cls");

}

//删除职工

void WorkerManager::Del\_Emp()

{

if (this->m\_FileIsEmpty)

{

cout << "文件不存在或者为空" << endl;

}

else

{

cout << "请选择按姓名删除或编号删除" << endl;

cout << "1、按照职工编号查找 " << endl;

cout << "2、按照职工姓名查找 " << endl;

int index=0;

int temp=0;

cin >> index;

//按编号查找

if (index == 1)

{

int id;

cout << "请输入编号: " << endl;

cin >> id;

temp = this->IsExist1(id);

}

//按姓名查找

if (index == 2)

{

int name;

cout << "请输入姓名: " << endl;

cin >> name;

temp = this->IsExist1(name);

}

if (temp != -1)

{

for (int i = temp; i < Number - 1; i++)

{

this->m\_Emparray[i] = this->m\_Emparray[i + 1];

}

this->Number--;

//同步更新到文件中

this->save();

cout << "删除成功" << endl;

}

else

{

cout << "删除失败 未找到该职工!" << endl;

}

}

system("pause");

system("cls");

}

//判断职工是否存在

int WorkerManager::IsExist1(int id)

{

int temp = -1;

for (int i = 0; i < Number; i++)

{

if (this->m\_Emparray[i]->m\_id == id)

{

temp = i;

break;

}

}

return temp;

}

int WorkerManager::IsExist2(string name)

{

int temp = -1;

for (int i = 0; i < Number; i++)

{

if (this->m\_Emparray[i]->m\_name == name)

{

temp = i;

break;

}

}

return temp;

}

void WorkerManager::mod\_emp()

{

if (this->m\_FileIsEmpty)

{

cout << "文件不存在或者为空" << endl;

}

else

{

cout << "请输入修改职工的编号: " << endl;

int id;

cin >> id;

int temp = this->IsExist1(id);

if (temp != -1)

{

delete this->m\_Emparray[temp];

int newid=0;

string newname="";

string xingbie="";

int age=0;

string add="";

string zhicheng="";

int gongzi=0;

int dianhua=0;

cout << "查找到: " << id << "号职工，请输入新职工号: " << endl;

cin >> newid;

cout << "请输入新姓名: " << endl;

cin >> newname;

cout << "请输入新性别 " << endl;

cin >> xingbie;

cout << "请输入新年龄: " << endl;

cin >> age;

cout << "请输入新岗位: " << endl;

cin >> add;

cout << "请输入新职称: " << endl;

cin >> zhicheng;

cout << "请输入新工资: " << endl;

cin >> gongzi;

cout << "请输入新电话: " << endl;

cin >> dianhua;

worker\* emp = NULL;

emp = new worker(newid, newname, xingbie, age, add, zhicheng, gongzi, dianhua);

this->m\_Emparray[temp] = emp;

cout <<"修改成功" << endl;

this->save();

}

else

{

cout << "未查找到该员工,修改失败" << endl;

}

}

system("pause");

system("cls");

}

void WorkerManager::find\_emp()

{

if (this->m\_FileIsEmpty)

{

cout << "文件不存在或者为空" << endl;

}

else

{

cout << "请输入查找方法: " << endl;

cout << "1、按照职工编号查找 " << endl;

cout << "2、按照职工姓名查找 " << endl;

int findmod = 0;

cin >> findmod;

if (findmod == 1)

{

//按照编号查找

int id;

cout << "请输入查找的职工编号: " << endl;

cin >> id;

int temp=IsExist1(id);

if (temp != -1)

{

cout << "查找到职工 该职工信息如下: " << endl;

this->m\_Emparray[temp]->showInfo();

}

else

{

cout << "查无此人！" << endl;

}

}

else if (findmod == 2)

{

//按照姓名查找

string name;

cout << "请输入查找的姓名: " << endl;

cin >> name;

//是否找到的标志

bool flag = false;

for (int i = 0; i < Number; i++)

{

if (this->m\_Emparray[i]->m\_name == name)

{

cout << "查找成功，职工编号为" << this->m\_Emparray[i]->m\_id

<< "的员工信息如下: " << endl;

flag = true;

this->m\_Emparray[i]->showInfo();

}

}

if (flag == false)

{

cout << "查无此人！" << endl;

}

}

else

{

cout << "输入选项有误." << endl;

}

}

system("pause");

system("cls");

}

WorkerManager::~WorkerManager()

{

if (this->m\_Emparray != NULL)

{

delete[] this->m\_Emparray;

this->m\_Emparray = NULL;

}

}

**Worker.h**

#pragma once

#include<iostream>

#include<string>

using namespace std;

//职工类

class worker

{

public:

worker(int id,string name,string xingbie,int age,string add,string zhicheng,int gongzi,int dianhua)

{

m\_id = id;

m\_name = name;

m\_xingbie = xingbie;

m\_age = age;

m\_add = add;

m\_zhicheng = zhicheng;

m\_gongzi = gongzi;

m\_dianhua = dianhua;

}

//职工编号

int m\_id;

//职工姓名

string m\_name;

//性别

string m\_xingbie;

//出生年月(年龄)

int m\_age;

//所在部门

string m\_add;

//职称

string m\_zhicheng;

//工资级别

int m\_gongzi;

//电话号码

int m\_dianhua;

void showInfo()

{

cout << "编号: " << this->m\_id

<< "\t姓名： " << this->m\_name

<< "\t性别： " << this->m\_xingbie

<< "\t年龄： " << this->m\_age

<< "\t部门： " << this->m\_add

<< "\t职称： " << this->m\_zhicheng

<< "\t工资： " << this->m\_gongzi

<< "\t电话： " << this->m\_dianhua << endl;

}

};

职工管理系统.cpp

#include<iostream>

using namespace std;

#include"WorkerManager.h"

#include"Worker.h"

int main()

{

WorkerManager wm;

int num = 0;//记录用户的选择

while (true)//死循环 不返回0不退出

{

wm.Menu();

cout << "请输入你的选择: " << endl;

cin >> num;

switch (num)

{

case 0://退出系统

wm.Exit();

break;

case 1://增加职工信息

wm.Add\_Emp();

break;

case 2://显示职工信息

wm.show\_Emp();

break;

case 3://查找职工信息

wm.find\_emp();

break;

case 4://修改职工信息

wm.mod\_emp();

break;

case 5://删除职工信息

wm.Del\_Emp();

default:

system("cls");

break;

}

}

system("pause");

return 0;

}

六、参考文献

C++面向对象程序设计教程(第4版)

(陈维兴 清华大学出版社)

七、设计体会

短学期的课程设计是一次非常珍贵的机会，这是一次可以让我们所学的理论与实际相结合的机会。我们选择了成绩统计系统这一题目。通过这次比较完整的一个程序的设计，我走出了纯理论的学习，从一种全新的角度去学习。并且在小团队实现的过程中，除去知识的大量更新，我学到了很多东西。

就个人而言，自己看书学习的经验，以及从网上以及其他各种途径获得信息和知识的经验都让我受益匪浅。理论与实际相结合的设计，锻炼了我综合运用所学的基础知识，解决实际问题的能力，同时也提高我查阅文献资料、对程序整体的把握等其他能力水平。而且通过对整体的掌控，对局部的取舍，以及对细节的斟酌处理，都使我的能力得到了锻炼我的各方面经验都得到了极大的丰富。

提高是有限的但提高也是全面的。正是这一次设计让我积累了许多实际经验。也必然会让我在未来的工作学习中表现出更高的耐力、理解力、实践力。顺利如期的完成本次课程设计给了我很大的信心，但是也为我指出很多不足的地方。学习其实就是一个不断完善的过程，正视自己的不足之处。在以后的工作和学习中不断的弥补这些不足之处，在以后的生活中也要保持同样的态度，不断地完善自己。

为期一周的 C++课程设计就要这样画上一个句号了。从这不长不短的一周中，我获得了知识，学到了研究的坚持与韧性，这一周不仅仅是交出了一份作业，还对自己有了新的认识。

潘钊拓

计算机2102

2106410230