華南理工大學

SOUTH CHINA UNIVERSITY OF TECHNOLOGY

C++程序设计基础

课程设计报告

设计题目：职工档案管理系统设计

学　　号 202330372051

班　　级 信息工程中法菁英班

姓　　名 温传志

完成方式：（单选）	自我评分：（单选）
○　独立完成	○　优秀
■　参考他人工作的基础上完成	■　良好
○　在他人的帮助下完成	○　中等
○　完整拷贝他人成果	○　及格
	○　不及格

2023 年 5 月 22 日

设计任务：C++语言课程设计任务书 4

题目：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。
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。

分步实施：
1、初步完成总体设计，搭好框架，确定人机对话的界面，确定函数个数。
2、建立一个文件，将每位职工的信息写入文件中并能显示于屏幕上。
3、完成上述查找（编号，姓名，性别等）、修改等功能。
4、进一步要求，修改记录设置密码操作。

要求：1、用 C++语言实现程序设计；
2、利用结构体（或类）数组实现职工信息的数据结构设计；
3、系统的各个功能模块要求用函数的形式实现；
4、界面友好（良好的人机交互），程序加必要的注释。

一、总体设计

1．程序的总体设计

构建两个类，员工类和系统类，构建相应成员函数实现系统的添加员工、查找员工和列出员工的功能，同时运用文件保存员工数据。

2. 数据结构

员工类：{变量：ID、姓名、性别、电话}，存放于头文件

系统类：{变量：员工数量、员工数组、判断文件。

基础函数：构造函数、析构函数

员工函数：添加函数、查找函数、列出函数

功能函数：初始化函数、菜单函数、退出函数

文件函数：保存函数、判断员工函数、获取员工数量函数

}存放于头文件

主函数：{

定义一个系统类 ES;

定义一个整型变量 choice;

While（true）

{

Switch（choice）

{Case 0：

退出系统

Case 1：

添加员工；

Case 2：

列出员工；

Case 3：

查找员工}

}

}

3. 函数原型声明的说明（功能，参数，返回值）

```
EmployeeSystem();//构造函数
void ShowMenu();//显示菜单
void ExitSystem();//退出系统
void AddEmp();//添加员工
void SearchEmp();//查找员工
void ListEmp();//列出员工
void Save();//保存文件
void InitEmp();//初始化函数
int GetNum();//获取员工数量
```

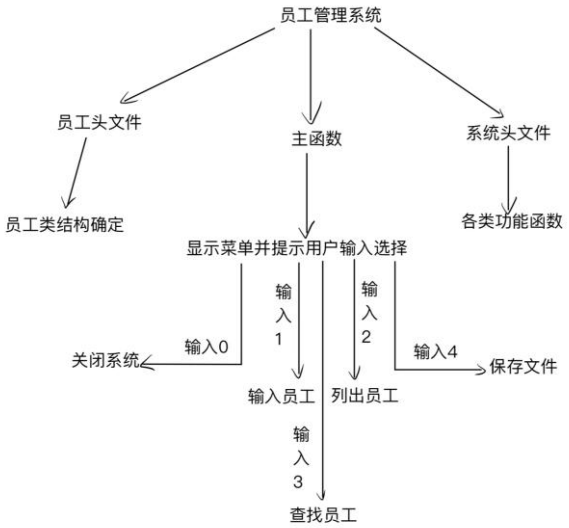
输出参数：员工总数量
返回值：整型

```
long Exist(long id);//判断员工存在性函数
```

输入参数：员工ID
输出参数：对应数组下标
返回值：长整型

```
~EmployeeSystem();//析构函数
```

4. 流程图



二、详细设计

1、员工头文件设计

```
#pragma once
#include<iostream>
#include<string>
using namespace std;
class Employee {
public:
    long Id;
    string Name;
    char Gender ; //用字符型表示以省空间，F或M
    long Phone ;//员工信息
    Employee(long id, string name, char gender, long phone);//构造函数
    void ShowInfo();//显示员工信息
};//定义员工类
```

2、员工系统头文件设计

```
#pragma once
#include<iostream>
#include<string>
#include"Employee.h"
#include<fstream>
#define  File "EmpFile.txt"
using namespace std;
class EmployeeSystem
{
public:
    int EmpNum;//总员工数量
    Employee**EmpArray;//员工数组
    bool Empty;//判断文件是否为空
    EmployeeSystem();//构造函数
    void ShowMenu();//显示菜单
    void ExitSystem();//退出系统
    void AddEmp();//添加员工
    void SearchEmp();//查找员工
    void ListEmp();//列出员工
    void Save();//保存文件
```

```
    void InitEmp();//初始化函数
    int GetNum();//获取员工数量
    long Exist(long id);//判断员工存在性函数
    ~EmployeeSystem();//析构函数
};//构建系统类
```

3、员工系统 cpp 文件设计

```
#include"EmployeeSystem.h"
#include"Employee.h"
#include<iostream>
#include<string>
EmployeeSystem::EmployeeSystem()
{
    ifstream ifs;
    ifs.open(File, ios::in);
    if (!ifs.is_open() )
    {
        cout << "No Document" << endl;
        this->EmpArray = NULL;
        this->EmpNum = 0;
        this->Empty = true;
        ifs.close();
        return;
    }//文件不存在
    char ch;
    ifs >> ch;
    if (ifs.eof())
    {
        cout<< "File is Empty" << endl;
        this->EmpNum = 0;
        this->Empty = true;
        this->EmpArray = NULL;
        ifs.close();
        return;
    }//文件不存在
    int num = this->GetNum();
    this->EmpNum = num;//导入员工数量
    this->EmpArray = new Employee * [this->EmpNum];//开辟空间
    this->InitEmp();
}
void EmployeeSystem::InitEmp()
{
    ifstream ifs;
    ifs.open(File, ios::in);
```

```
    long id;
    string name;
    char gender;
    long phone;
    int index = 0;
    while (ifs >> id && ifs >> name && ifs >> gender && ifs >> phone)
    {
        Employee* emp = new Employee(id,name,gender,phone);
        this->EmpArray[index] = emp;
        index++;
    }//逐一对员工元素赋值
    ifs.close();
}//初始化员工人数及数组
long EmployeeSystem::Exist(long id)
{
    int index = -1;
    for (int i = 0; i < this->EmpNum; i++)
    {
        if (this->EmpArray[i]->Id == id)
        {
            index = i;
            break;
        }
    }
    return index;
}//判断编号是否存在于系统中，并返回在数组中的位置
void EmployeeSystem::ShowMenu()
{
    cout <<"Welcome to Employee Administering System!" << endl;
    cout << "    "<<"Please Press Numbers to Continue"<<"        " <<
endl;
    cout << endl;
    cout <<"    " <<"0, " <<"Exit Employee Administering System       "
<< endl;
    cout << endl;
    cout << "    " << "1, " << "Add New Empolyee       " << endl;
    cout << endl;
    cout << "    " << "2, " << "Show All Employees        " << endl;
    cout << endl;
    cout << "    " << "3, " << "Search Empolyee       " << endl;
    cout << endl;
    cout << "    " << "4, " << "Save Files       " << endl;
}//显示菜单函数
void EmployeeSystem::ExitSystem()
```

```
{
    cout << "Shuting Down..." << endl;
    system("pause");
    exit(0);
}//关闭系统函数
void EmployeeSystem::AddEmp()
{
    cout << "Please Input the Number of Employees You Need to Add" << endl;
    int AddNum = 0;//输入添加员工数量
    cin >> AddNum;
    if (AddNum > 0)
    {
        int NewSize = this->EmpNum + AddNum;//新员工总数
        Employee** NewSpace = new Employee * [NewSize];//开辟对应大小的
空间
        if (this->EmpArray != NULL) //判断数组情况
        {
            for (int i = 0; i < EmpNum; i++)
            {
                NewSpace[i] = EmpArray[i];
            }
        }//输入现有员工数组
        for (int i = 0; i < AddNum; i++)
        {
            long id;  string  name; char gender; long phone;
            cout << "Please Input the Num." << i + 1 << "Employee's Id You
Want to Add" << endl;
            cin >> id;
            cout << "Please Input the Num." << i + 1 << "Employee's Name
You Want to Add" << endl;
            cin >> name;
            cout << "Please Input the Num." << i + 1 << "Employee's Gender
You Want to Add" << endl;
            cin >> gender;
            cout << "Please Input the Num." << i + 1 << "Employee's Phone
You Want to Add" << endl;
            cin >> phone;
            Employee* ep = new Employee(id, name, gender, phone);
            NewSpace[EmpNum + i] = ep;
        }////添加新员工信息并将新员工添加入员工数组
        this->EmpArray = NewSpace;//更改新空间指向
        this->EmpNum = NewSize;//更新员工总人数
        cout << AddNum << "New Employees Have Been Added" << endl;
        this->Empty = false;
```

```
        this->Save();//保存到文件中
    }
    else {
        cout << "ERROR" << endl;
        system("cls");
        }//输入数值不合理时清屏
}//添加员工函数
void EmployeeSystem::Save()
{
    ofstream ofs;
    ofs.open(File, ios::out);//打开并写入文件
    for (int i = 0; i < EmpNum; i++)
    {
        ofs <<this->EmpArray[i]->Id << " "
            << this->EmpArray[i]->Name << " "
            << this->EmpArray[i]->Gender << " "
            <<this->EmpArray[i]->Phone << endl;
    }
    ofs.close();//关闭文件
}//保存函数
void EmployeeSystem::SearchEmp()
{
    if (this->Empty)
    {
        cout << "No Results" << endl;
    }
    else
    {
        cout << "Please Input the Id of the Employee You Want to Search"
<< endl;
        long x = 0;
        cin >> x;
        long index = Exist(x);//判断员工是否存在
        if (index != -1)
        {
            cout << "the Employee You Want to Search is" << endl;
            this->EmpArray[index]->ShowInfo();//显示员工
        }
        else
        {
            cout << "No Results" << endl;
        }
    }
}//查找员工函数
```

```
void EmployeeSystem::ListEmp()
{
    if (this->Empty)
    {
        cout << "No Results" << endl;
    }
    else
    {
        for (int i = 0; i < this->EmpNum; i++)
        {
            this->EmpArray[i]->ShowInfo();
        }
    }
    system("pause");
    system("cls");
}//列出员工函数
int EmployeeSystem::GetNum()
{
    ifstream ifs;
    ifs.open(File, ios::in);
    int id;
    string name;
    char gender;
    long phone;
    int num = 0;
    while (ifs >> id && ifs >> name && ifs >> gender && ifs >> phone)
    {
        num++;
    }
    ifs.close();
    return num;
}//获取员工数量函数
EmployeeSystem::~EmployeeSystem()
{
}//析构函数
```

4、员工 cpp 文件设计

```
#include"Employee.h"
Employee::Employee(long id,string name,char gender,long phone)
{
    this->Id = id;
    this->Name = name;
    this->Gender = gender;
    this->Phone = phone;
```

```
}//初始化员工信息
void Employee::ShowInfo()
{
    cout << "Id:          " << this->Id << "\t";
    cout << "Name:    " << this->Name << "\t";
    cout << "Gender: " << this->Gender << "\t";
    cout << "Phone:    " << this->Phone  << "\t" << endl;
};//显示员工信息函数
```

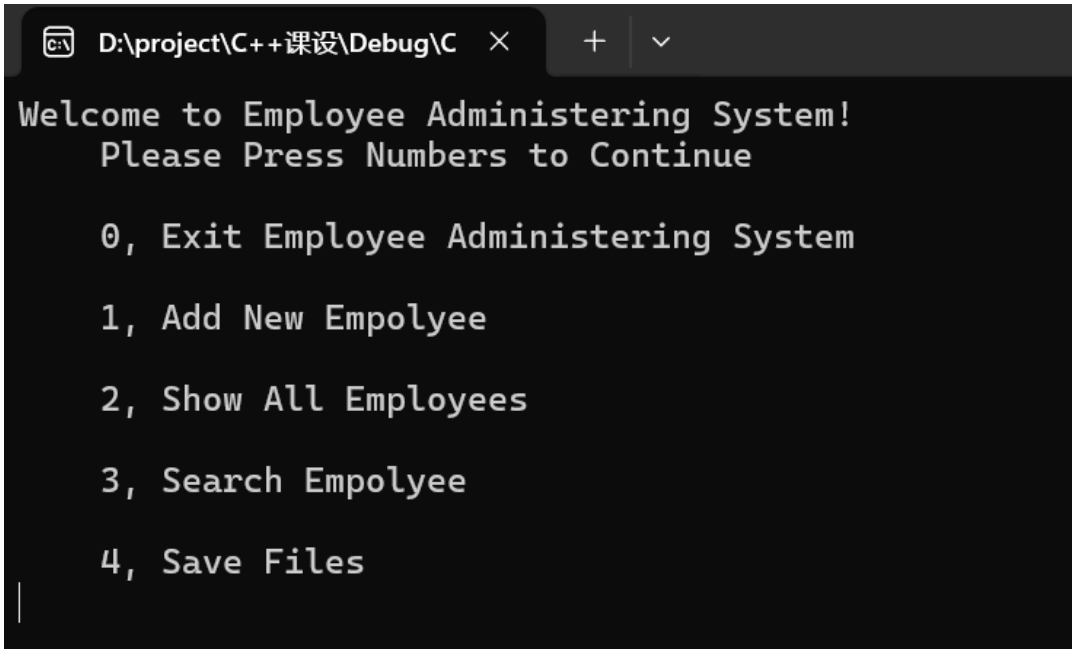
5、主函数设计

```
#include<iostream>
#include"EmployeeSystem.h"
#include"Employee.h"
int main()
{
    EmployeeSystem es;//建立新系统
    int choice=0;//记录管理员的选择
    while (true)
    {
        es.ShowMenu();
        cin >> choice;
        switch (choice)
        {
        case 0://退出系统
            es.ExitSystem();
            break;
        case 1://添加员工
            es.AddEmp();
            break;
        case 2://列出所有员工
            es.ListEmp();
            break;
        case 3://查找员工
            es.SearchEmp();
            break;
        case 4:
           cout << "File Has Been Saved " << endl;
            es.Save();
            break;//保存当前信息
        default:
            system("cls");//输入无效数字后清屏，输入字母或其他键自动关闭
            break;
        }
    }
```

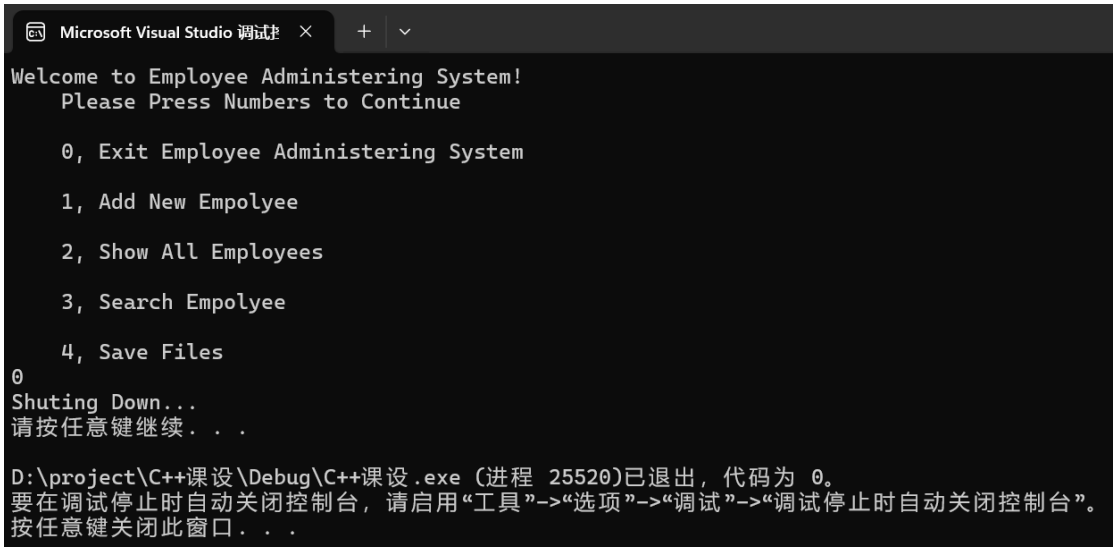
```
    system("pause");
}
```

三、测试及调试

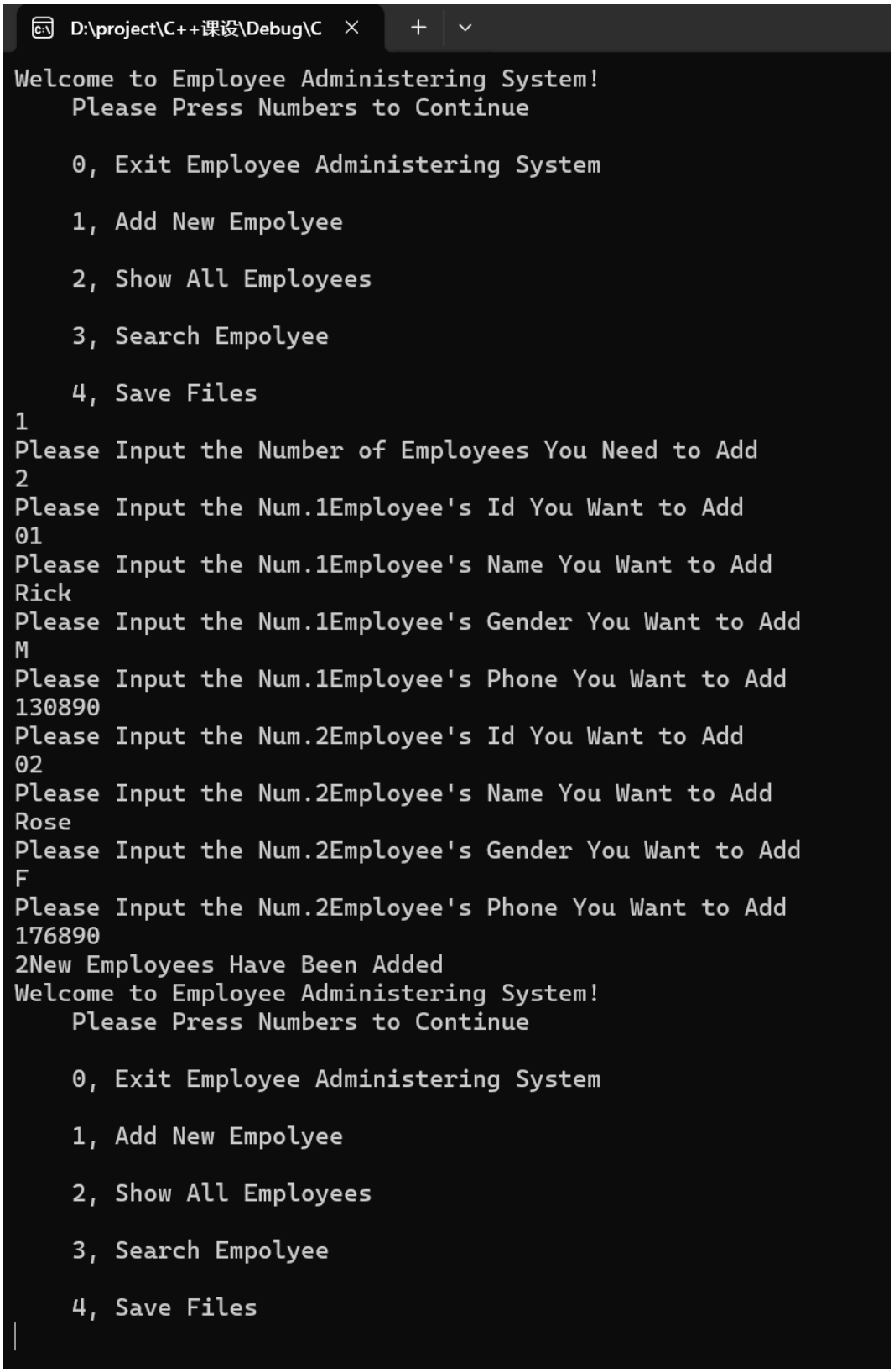
1、启动界面浏览



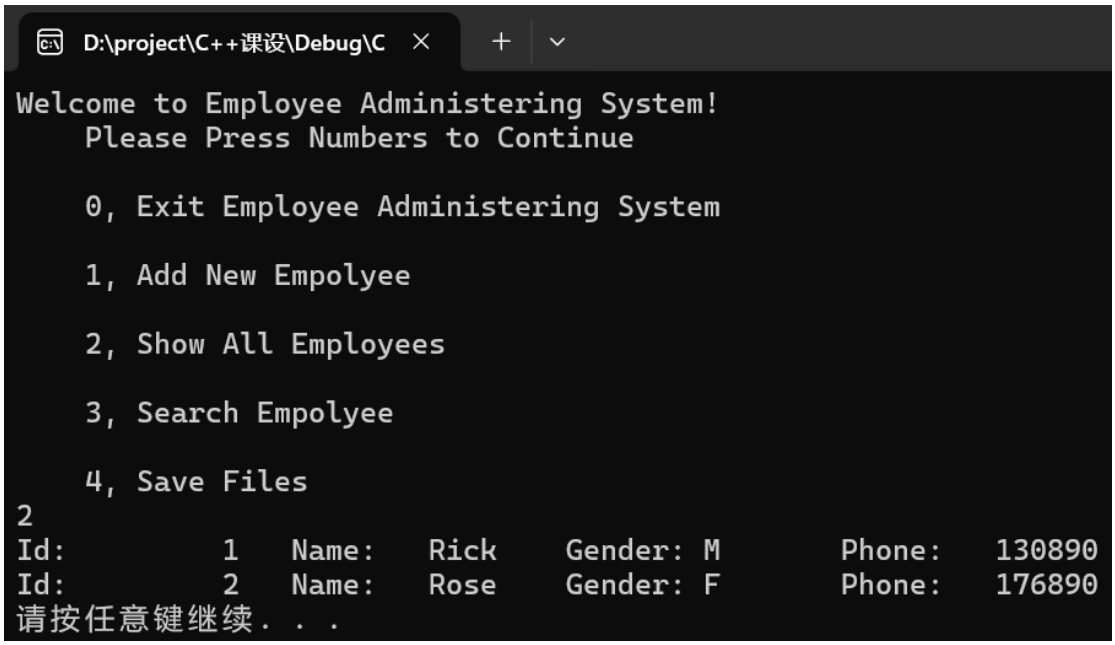
2、退出测试


Microsoft Visual Studio 调试控
Welcome to Employee Administering System!
Please Press Numbers to Continue
0, Exit Employee Administering System
1, Add New Empolyee
2, Show All Employees
3, Search Empolyee
4, Save Files
0
Shuting Down...
请按任意键继续. . .
D:\project\C++课设\Debug\C++课设.exe (进程 25520)已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口. . .

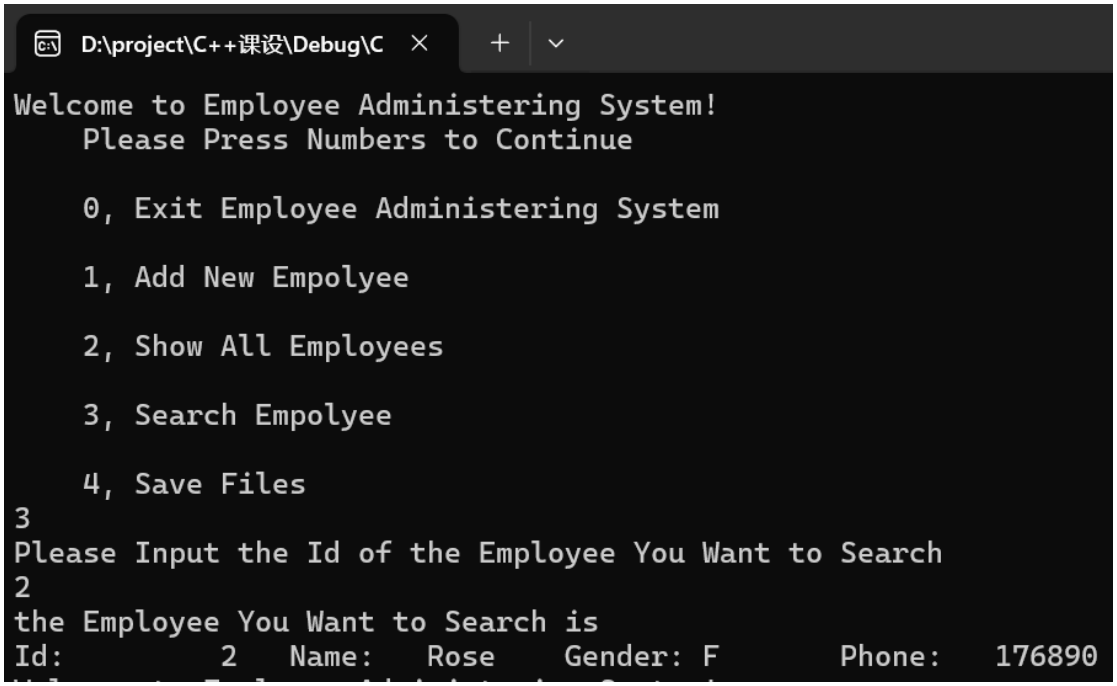
3、添加员工测试


D:\project\C++课设\Debug\C
Welcome to Employee Administering System!
Please Press Numbers to Continue
0, Exit Employee Administering System
1, Add New Empolyee
2, Show All Employees
3, Search Empolyee
4, Save Files
1
Please Input the Number of Employees You Need to Add
2
Please Input the Num.1Employee's Id You Want to Add
01
Please Input the Num.1Employee's Name You Want to Add
Rick
Please Input the Num.1Employee's Gender You Want to Add
M
Please Input the Num.1Employee's Phone You Want to Add
130890
Please Input the Num.2Employee's Id You Want to Add
02
Please Input the Num.2Employee's Name You Want to Add
Rose
Please Input the Num.2Employee's Gender You Want to Add
F
Please Input the Num.2Employee's Phone You Want to Add
176890
2New Employees Have Been Added
Welcome to Employee Administering System!
Please Press Numbers to Continue
0, Exit Employee Administering System
1, Add New Empolyee
2, Show All Employees
3, Search Empolyee
4, Save Files

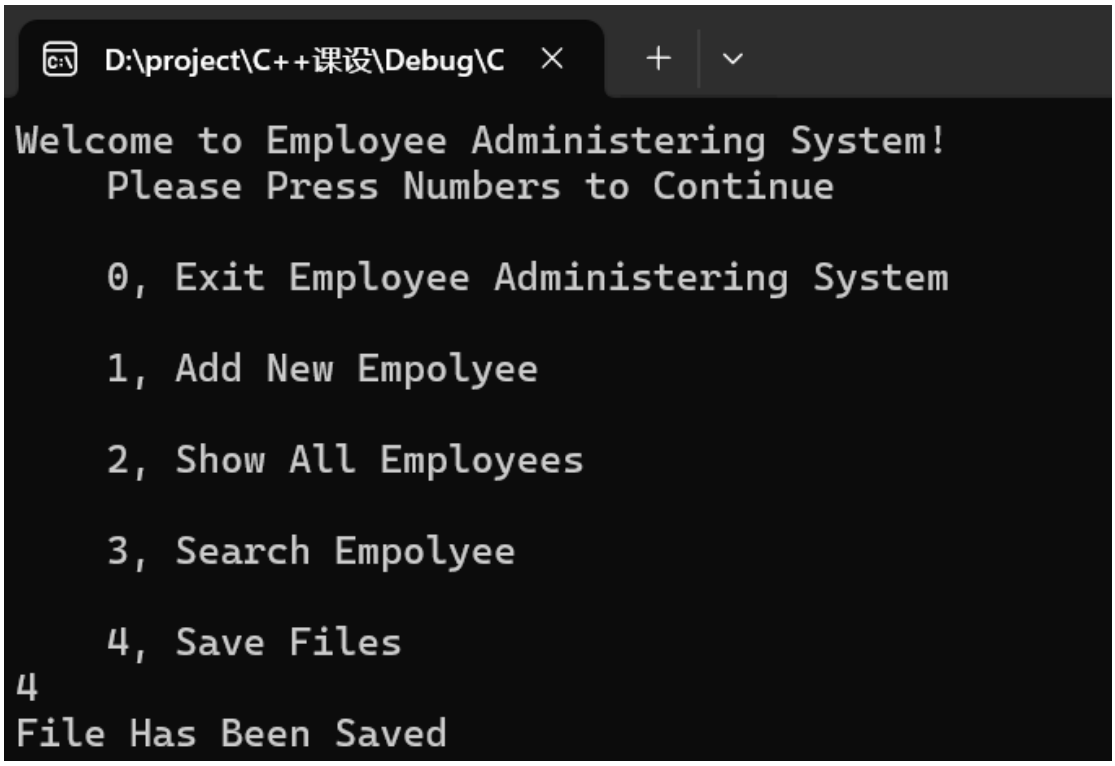
4、浏览员工测试



5、查找员工测试



6、保存文件测试



7、实际文件测试

四、小结（收获和建议）

通过本次课程设计，我的编程代码能力得到了提高，对 C++代码有了更加深刻的认识。通过一次次对课程设计代码的调试，我体会到了软件开发人员的艰辛，开发软件编程是一项庞大的工程，需要很有

耐心，静下心来认认真真，勤勤恳恳地专注做一个项目。总体来说，此次课程设计是具有教育意义和现实意义的，对当今的学习以及未来的职业发展都很有帮助。