严禁抄袭

仅供参考

Blog:Zhangshier.vip

目录

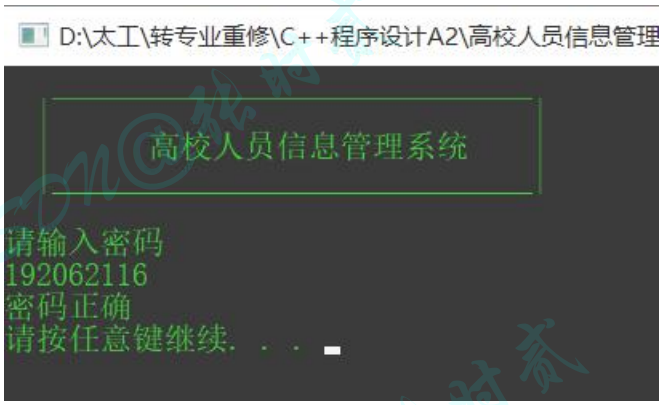
一、项目描述.....2
1.1 登录密码.....2
1.2 菜单.....2
1.3 保存与读取.....3
1.4 添加功能.....3
1.5 查询功能.....4
1.6 显示功能.....4
1.7 编辑功能.....5
1.8 删除功能.....5
1.9 统计功能.....6
二、项目设计.....6
2.1 继承关系说明.....6
2.2 全局函数说明.....7
三、项目实现.....8
3.1Person 类.....8
3.2Teacher 类.....8
3.3 Experiment 类.....16
3.3 Politician 类.....23
3.4 Tea_Po 类.....30
3.5 教师菜单.....37
3.6 实验人员菜单.....39
3.7 行政人员类菜单.....40
3.8 教师兼行政人员菜单.....41
四、项目总结.....46
五、参考文献.....46

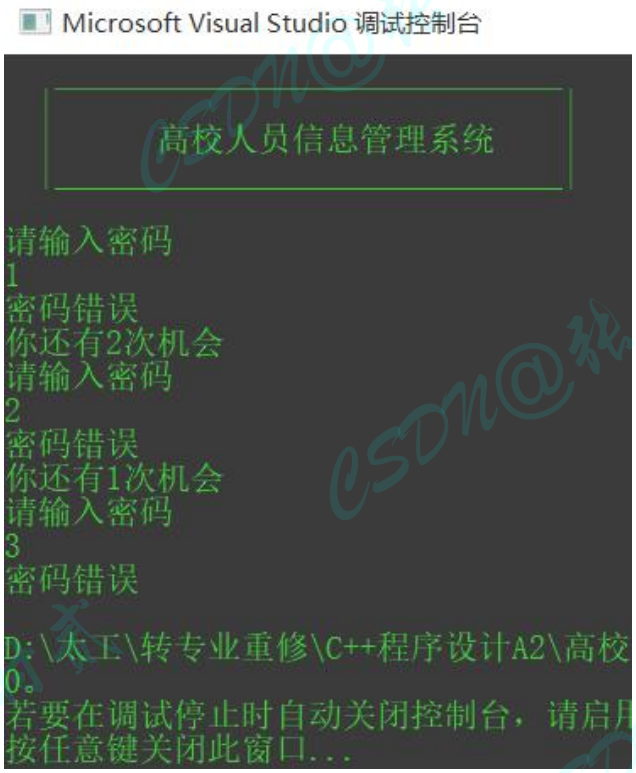
一、 项目描述

（描述项目的一段话，附运行效果截屏）

1.1 登录密码

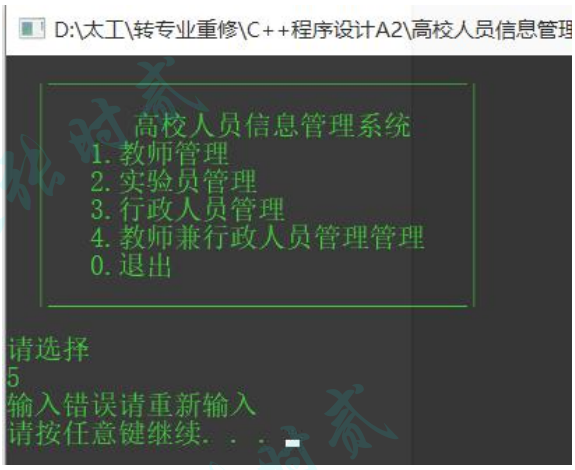
进入需要输入管理密码，如果错误超过三次，退出程序





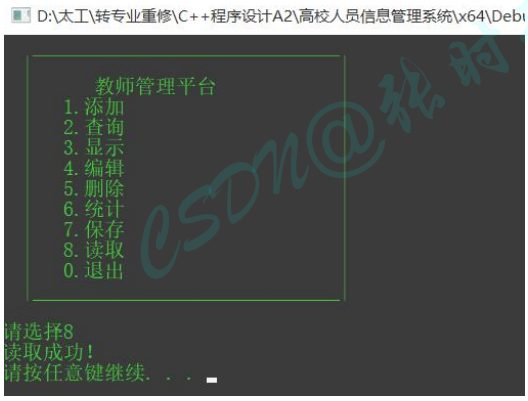
1.2 菜单

在任何界面如果输入菜单以外的选项，返回错误信息



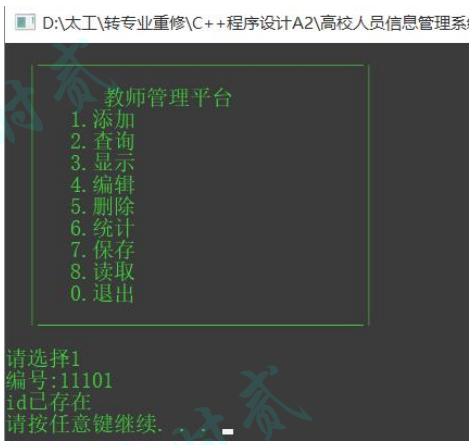
1.3 保存与读取

以教师管理为例，可以保存或读取二进制文件



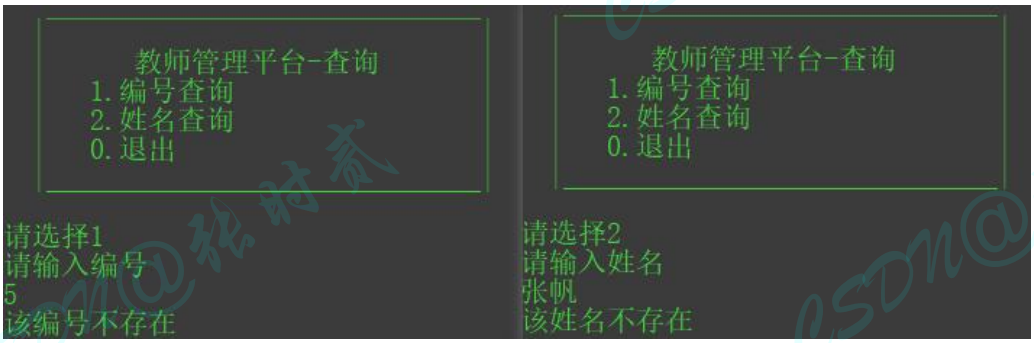
1.4 添加功能

如果 id 已存在，则返回提示信息，如果不存在则添加其余信息



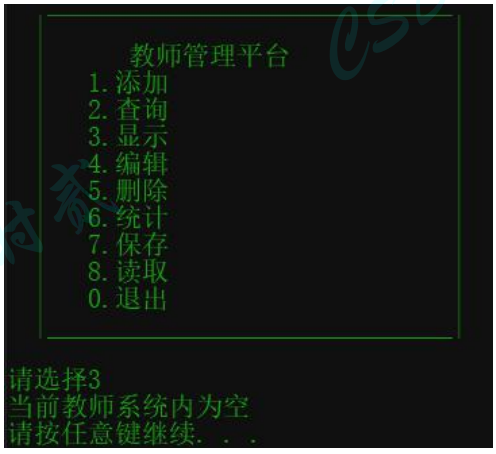
1.5 查询功能

可以利用编号或姓名查询，如果不存在则返回提示信息，如果存在则显示输出对应信息



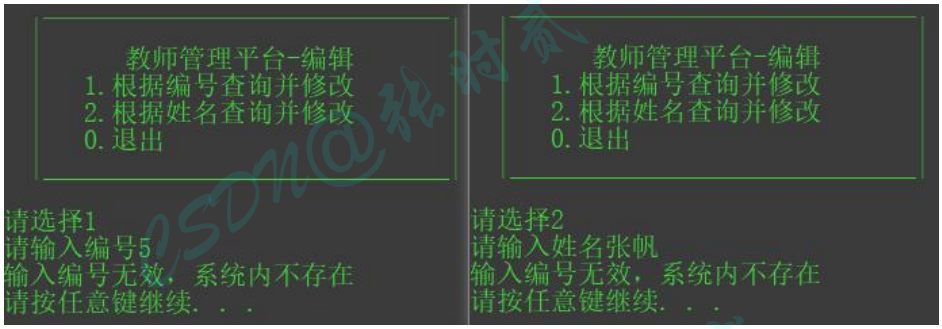
1.6 显示功能

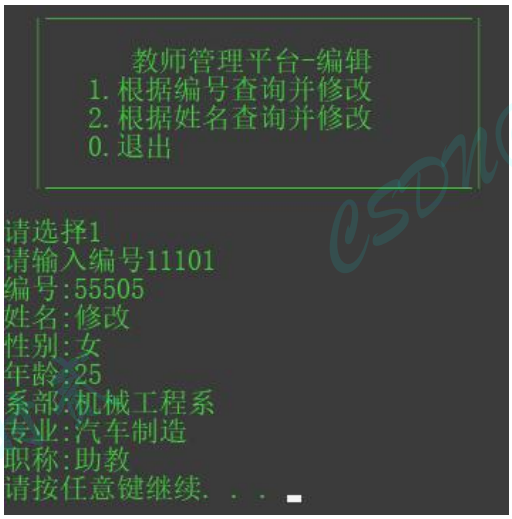
显示该人员中所有人员信息，如果管理平台内为空则返回提示信息



1.7 编辑功能

可以利用编号或姓名编辑，并先检索是否有该信息，如果有则进入修改页面，如果没有返回提示信息

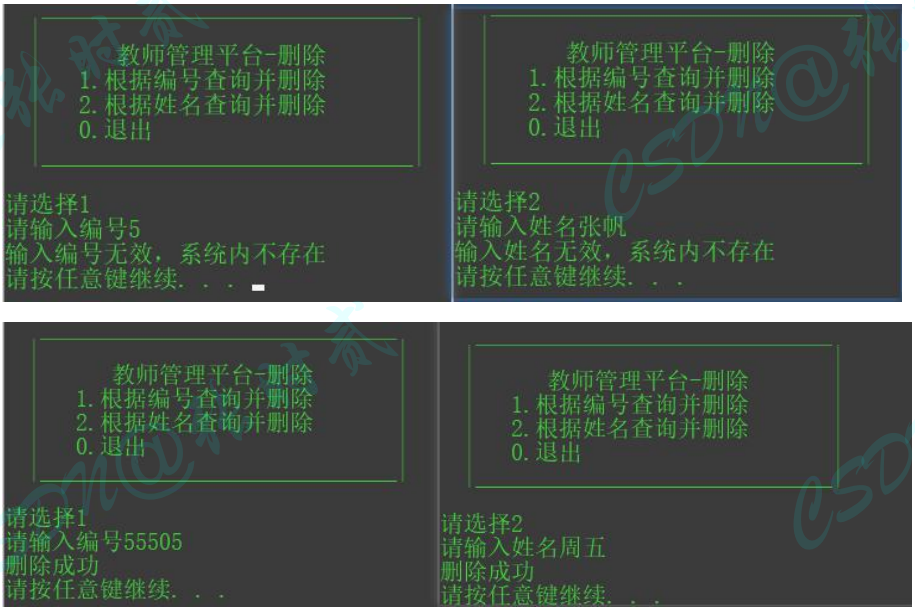




1.8 删除功能

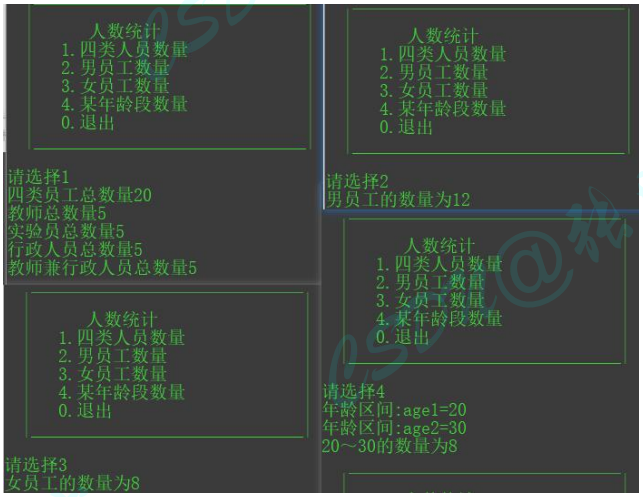
可以利用编号或姓名删除，并先检索是否有该信息，如果有则返回“删

除成功”信息，如果没有则返回提示信息



1.9 统计功能

可根据四类人员的数量、男员工数量、女员工数量、年龄段分别统计



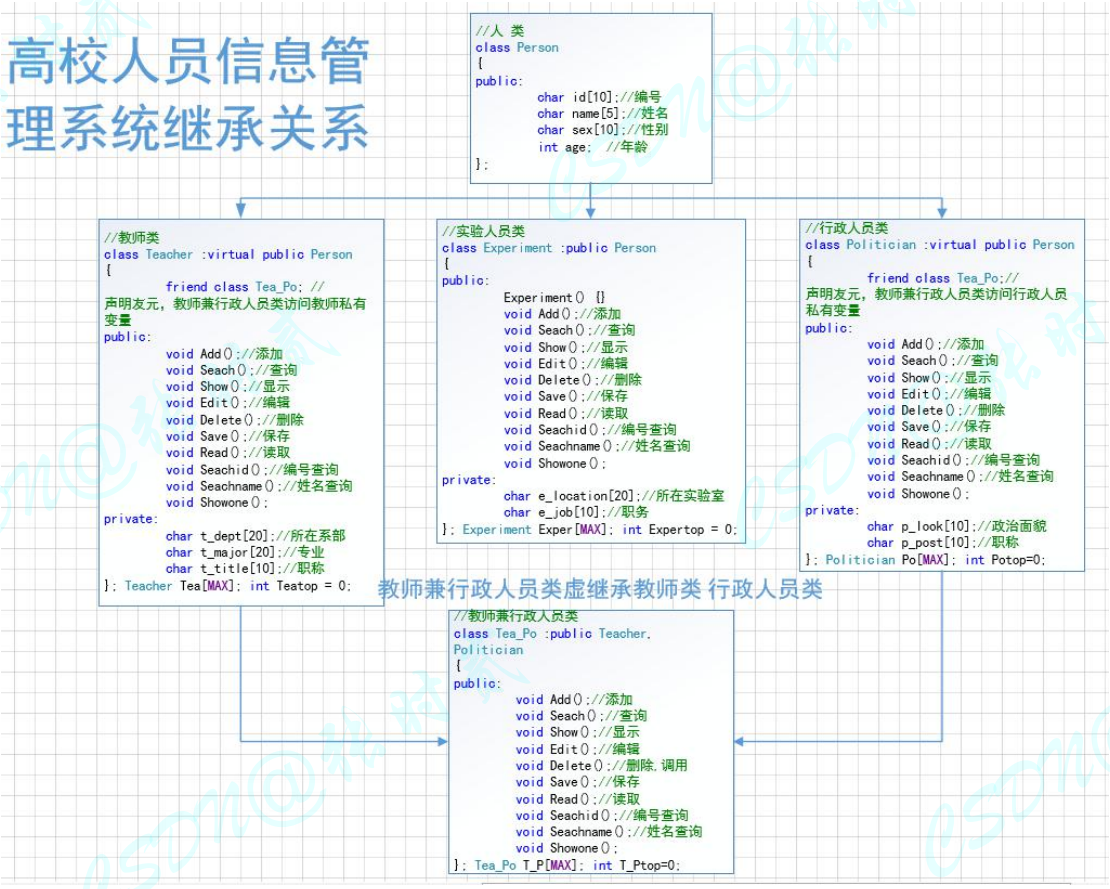
二、 项目设计

（原理、算法和模型）

2.1 继承关系说明

使用菱形继承，教师类、实验人员类、行政人员类、分别继承人类，

教师兼行政人员类虚继承教师类和行政人员类



2.2 全局函数说明

分别为每一个人员类编写单独的菜单

三、 项目实现

（关键代码）

3.1Person 类

```
//人 类
class Person
{
public:
    char id[10];  //编号
    char name[5];//姓名
    char sex[10];//性别
    int age;  //年龄
};
```

3.2Teacher 类

```
//教师类
class Teacher :virtual public Person
{
```

```
    friend class Tea_Po; //声明友元，教师兼行政人员类访问教师私有变量
public:
    void Add();//添加
    void Seach();//查询
    void Show();//显示
    void Edit();//编辑
    void Delete();//删除
    void Save();//保存
    void Read();//读取
    void Seachid();//编号查询
    void Seachname();//姓名查询
    void Showone();
private:
    char t_dept[20];//所在系部
    char t_major[20];//专业
    char t_title[10];//职称
}; Teacher Tea[MAX]; int Teatop = 0;

void Teacher::Add()
{
    Teacher t_temp;//临时变量
    if (Teatop < MAX)
    {
        cout << "编号:"; cin >> t_temp.id;
        for (int i = 0; i < Teatop; i++)
        {
            if (!strcmp(Tea[i].id, t_temp.id))
            {
                cout << "id 已存在" << endl;
                return;
            }
        }
        cout << "姓名:"; cin >> t_temp.name;
        cout << "性别:"; cin >> t_temp.sex;
        cout << "年龄:"; cin >> t_temp.age;
        cout << "系部:"; cin >> t_temp.t_dept;
        cout << "专业:"; cin >> t_temp.t_major;
        cout << "职称:"; cin >> t_temp.t_title;
        Tea[Teatop] = t_temp;
        Teatop++;
    }
    else
    {
        cout << "当前教师已招满" << endl;
```

```
    }
}
void Teacher::Seach()
{
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |_____________________________________|" << endl;
        cout << "  |                                     |" << endl;
        cout << "  |       教师管理平台-查询             |" << endl;
        cout << "  |    1.编号查询                       |" << endl;
        cout << "  |    2.姓名查询                       |" << endl;
        cout << "  |    0.退出                           |" << endl;
        cout << "  |_____________________________________|" << endl << endl;
        cout << "请选择"; cin >> c;
        switch (c)
        {
        case 1:
            Seachid();
            break;
        case 2:
            Seachname();
            break;
        case 0:
            return;
            break;
        default:
            cout << "输入错误请重新输入" << endl;
            break;
        }
    }
}
void Teacher::Show()
{
    if (Teatop == 0)
    {
        cout << "当前教师系统内为空" << endl;
    }
    else
    {
        for (int i = 0; i < Teatop; i++)
        {
            Tea[i].Showone();
        }
```

```
    }
}
void Teacher::Edit()
{
    Teacher t_temp;
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |————————————————————————————————|" << endl;
        cout << "  |          教师管理平台-编辑          |" << endl;
        cout << "  |    1.根据编号查询并修改             |" << endl;
        cout << "  |    2.根据姓名查询并修改             |" << endl;
        cout << "  |    0.退出                          |" << endl;
        cout << "  |————————————————————————————————|" << endl << endl;
        cout << "请选择"; cin >> c;
        if (c == 1)
        {
            char sid[10];
            int i;
            cout << "请输入编号"; cin >> sid;
            for (i = 0; i < Teatop; i++)
            {
                if (!strcmp(Tea[i].id, sid))
                    break;
            }
            if (i == Teatop)
                cout << "输入编号无效，系统内不存在" << endl;
            else
            {
                cout << "编号:"; cin >> t_temp.id;
                cout << "姓名:"; cin >> t_temp.name;
                cout << "性别:"; cin >> t_temp.sex;
                cout << "年龄:"; cin >> t_temp.age;
                cout << "系部:"; cin >> t_temp.t_dept;
                cout << "专业:"; cin >> t_temp.t_major;
                cout << "职称:"; cin >> t_temp.t_title;
                Tea[i] = t_temp;
            }
            break;
        }
        else if (c == 2)
        {
            char sname[5];
```

```
            int i;
            cout << "请输入姓名"; cin >> sname;
            for (i = 0; i < Teatop; i++)
            {
                if (!strcmp(Tea[i].name, sname))
                    break;
            }
            if (i == Teatop)
                cout << "输入编号无效，系统内不存在" << endl;
            else
            {
                cout << "编号:"; cin >> t_temp.id;
                cout << "姓名:"; cin >> t_temp.name;
                cout << "性别:"; cin >> t_temp.sex;
                cout << "年龄:"; cin >> t_temp.age;
                cout << "系部:"; cin >> t_temp.t_dept;
                cout << "专业:"; cin >> t_temp.t_major;
                cout << "职称:"; cin >> t_temp.t_title;
                Tea[i] = t_temp;
            }
            break;
        }
        else if (c == 0)
        {
            break;
        }
        else
        {
            cout << "输入错误请重新选择" << endl;
        }
    }
}
void Teacher::Delete()
{
    if (Teatop == 0)
    {
        cout << "当前记录为空" << endl;
        return;
    }
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |------------------------------------|" << endl;
```

```
cout << "  |       教师管理平台-删除          |" << endl;
cout << "  |   1.根据编号查询并删除           |" << endl;
cout << "  |   2.根据姓名查询并删除           |" << endl;
cout << "  |   0.退出                         |" << endl;
cout << "  |__________________________________|" << endl << endl;
cout << "请选择"; cin >> c;
if (c == 1)
{
    char sid[10];
    int i;
    cout << "请输入编号"; cin >> sid;
    for (i = 0; i < Teatop; i++)
    {
        if (!strcmp(Tea[i].id, sid))
            break;
    }
    if (i == Teatop)
        cout << "输入编号无效，系统内不存在" << endl;
    else
    {
        for (i; i < Teatop; i++)
        {
            Tea[i] = Tea[i + 1];
        }
        cout << "删除成功" << endl;
        Teatop--;
    }
    break;
}
else if (c == 2)
{
    char sname[5];
    int i;
    cout << "请输入姓名"; cin >> sname;
    for (i = 0; i < Teatop; i++)
    {
        if (!strcmp(Tea[i].name, sname))
            break;
    }
    if (i == Teatop)
        cout << "输入姓名无效，系统内不存在" << endl;
    else
    {
        for (i; i < Teatop; i++)
```

```
                {
                    Tea[i] = Tea[i + 1];
                }
                cout << "删除成功" << endl;
                Teatop--;
            }
            break;
        }
        else if (c == 0)
        {
            break;
        }
        else
        {
            cout << "输入错误请重新选择" << endl;
        }
    }
}

void Teacher::Save()
{
    int i;
    ofstream outfile, outfile1;
    outfile1.open("Teatop.dat", ios::out); //写文件 Teatop.dat
    outfile1 << Teatop;
    outfile.open("Tea_data.dat", ios::binary);
    if (!outfile)
    {
        cerr << "open error!" << endl; return;
    }
    for (i = 0; i < Teatop; i++)
        outfile.write((char *)&Tea[i], sizeof(Tea[i]));
    outfile.close();
    cout << "保存成功！" << endl;
}
void Teacher::Read()
{
    int i;
    ifstream infile, infile1;
    infile1.open("Teatop.dat", ios::in);
    infile1 >> Teatop;
    infile.open("Tea_data.dat", ios::binary);
    if (!infile)
    {
```

```
        cerr << "open error!" << endl; return;
    }
    for (i = 0; i < Teatop; i++)
        infile.read((char *)&Tea[i], sizeof(Tea[i]));
    infile.close();
    cout << "读取成功！" << endl;
}
void Teacher::Seachid()
{
    char sid[10];
    cout << "请输入编号" << endl; cin >> sid;
    int i;
    for (i = 0; i < Teatop; i++)
    {
        if (!strcmp(Tea[i].id, sid))
        {
            Tea[i].Showone();
            break;
        }
    }
    if (i == Teatop)
    {
        cout << "该编号不存在" << endl;
    }
}
void Teacher::Seachname()
{
    char sname[5];
    cout << "请输入姓名" << endl; cin >> sname;
    int i;
    for (i = 0; i < Teatop; i++)
    {
        if (!strcmp(Tea[i].name, sname))
        {
            Tea[i].Showone();
            break;
        }
    }
    if (i == Teatop)
    {
        cout << "该姓名不存在" << endl;
    }
}
void Teacher::Showone()
```

```
{
    cout << "编号\t" << id << "\t 姓名\t" << name << "\t 性别\t" << sex
<< "\t 年龄\t" << age << "\t 系部\t" << t_dept << "\t 专业\t" << t_major
<< "\t 职称\t" << t_title << endl;
}
```

3.3 Experiment 类

```
//实验人员类
class Experiment :public Person
{
public:
    Experiment() {}
    void Add();//添加
    void Seach();//查询
    void Show();//显示
    void Edit();//编辑
    void Delete();//删除
    void Save();//保存
    void Read();//读取
    void Seachid();//编号查询
    void Seachname();//姓名查询
    void Showone();
private:
    char e_location[20];//所在实验室
    char e_job[10];//职务
}; Experiment Exper[MAX]; int Expertop = 0;

void Experiment::Add()
{
    Experiment e_temp;//临时变量
    if (Expertop < MAX)
    {
        cout << "编号:"; cin >> e_temp.id;
        for (int i = 0; i < Expertop; i++)
        {
            if (!strcmp(Exper[i].id, e_temp.id))
            {
                cout << "id 已存在" << endl;
                return;
            }
        }
        cout << "姓名:"; cin >> e_temp.name;
        cout << "性别:"; cin >> e_temp.sex;
```

```
        cout << "年龄:"; cin >> e_temp.age;
        cout << "实验室:"; cin >> e_temp.e_location;
        cout << "职务:"; cin >> e_temp.e_job;
        Exper[Expertop] = e_temp;
        Expertop++;
    }
    else
    {
        cout << "当前实验人员已招满" << endl;
    }
}
void Experiment::Seach()
{
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |_____________________________|" << endl;
        cout << "  |  实验人员管理平台-查询      |" << endl;
        cout << "  |   1.编号查询                |" << endl;
        cout << "  |   2.姓名查询                |" << endl;
        cout << "  |   0.退出                    |" << endl;
        cout << "  |_____________________________|" << endl << endl;
        cout << "请选择"; cin >> c;
        switch (c)
        {
        case 1:
            Seachid();
            break;
        case 2:
            Seachname();
            break;
        case 0:
            return;
            break;
        default:
            cout << "输入错误请重新输入" << endl;
            break;
        }
    }
}
void Experiment::Show()
{
    if (Expertop == 0)
```

```
    {
        cout << "当前实验人员系统内为空" << endl;
    }
    else
    {
        for (int i = 0; i < Expertop; i++)
        {
            Exper[i].Showone();
        }
    }
}
void Experiment::Edit()
{
    Experiment e_temp;
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |————————————————————————————|" << endl;
        cout << "  |  实验人员管理平台-编辑     |" << endl;
        cout << "  |   1.根据编号查询并修改     |" << endl;
        cout << "  |   2.根据姓名查询并修改     |" << endl;
        cout << "  |   0.退出                   |" << endl;
        cout << "  |————————————————————————————|" << endl << endl;
        cout << "请选择"; cin >> c;
        if (c == 1)
        {
            char sid[10];
            int i;
            cout << "请输入编号"; cin >> sid;
            for (i = 0; i < Expertop; i++)
            {
                if (!strcmp(Exper[i].id, sid))
                    break;
            }
            if (i == Expertop)
                cout << "输入编号无效，系统内不存在" << endl;
            else
            {
                cout << "编号:"; cin >> e_temp.id;
                cout << "姓名:"; cin >> e_temp.name;
                cout << "性别:"; cin >> e_temp.sex;
                cout << "年龄:"; cin >> e_temp.age;
                cout << "实验室:"; cin >> e_temp.e_location;
```

```
            cout << "职务:"; cin >> e_temp.e_job;
            Exper[i] = e_temp;
        }
        break;
    }
    else if (c == 2)
    {
        char sname[5];
        int i;
        cout << "请输入姓名"; cin >> sname;
        for (i = 0; i < Expertop; i++)
        {
            if (!strcmp(Exper[i].name, sname))
                break;
        }
        if (i == Expertop)
            cout << "输入姓名无效，系统内不存在" << endl;
        else
        {
            cout << "编号:"; cin >> e_temp.id;
            cout << "姓名:"; cin >> e_temp.name;
            cout << "性别:"; cin >> e_temp.sex;
            cout << "年龄:"; cin >> e_temp.age;
            cout << "实验室:"; cin >> e_temp.e_location;
            cout << "职务:"; cin >> e_temp.e_job;
            Exper[i] = e_temp;
        }
        break;
    }
    else if (c == 0)
    {
        break;
    }
    else
    {
        cout << "输入错误请重新选择" << endl;
    }
  }
}
void Experiment::Delete()
{
    if (Expertop == 0)
    {
        cout << "当前记录为空" << endl;
```

```
        return;
    }
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |————————————————————————————————|" << endl;
        cout << "  |   实验人员管理平台-删除        |" << endl;
        cout << "  |   1.根据编号查询并删除         |" << endl;
        cout << "  |   2.根据姓名查询并删除         |" << endl;
        cout << "  |   0.退出                       |" << endl;
        cout << "  |————————————————————————————————|" << endl << endl;
        cout << "请选择"; cin >> c;
        if (c == 1)
        {
            char sid[10];
            int i;
            cout << "请输入编号"; cin >> sid;
            for (i = 0; i < Expertop; i++)
            {
                if (!strcmp(Exper[i].id, sid))
                    break;
            }
            if (i == Expertop)
                cout << "输入编号无效，系统内不存在" << endl;
            else
            {
                for (i; i < Expertop; i++)
                {
                    Exper[i] = Exper[i + 1];
                }
                cout << "删除成功" << endl;
                Expertop--;
            }
            break;
        }
        else if (c == 2)
        {
            char sname[5];
            int i;
            cout << "请输入姓名"; cin >> sname;
            for (i = 0; i < Expertop; i++)
            {
                if (!strcmp(Exper[i].name, sname))
```

```
                break;
            }
            if (i == Expertop)
                cout << "输入姓名无效，系统内不存在" << endl;
            else
            {
                for (i; i < Expertop; i++)
                {
                    Exper[i] = Exper[i + 1];
                }
                cout << "删除成功" << endl;
                Expertop--;
            }
            break;
        }
        else if (c == 0)
        {
            break;
        }
        else
        {
            cout << "输入错误请重新选择" << endl;
        }
    }
}

void Experiment::Save()
{
    int i;
    ofstream outfile, outfile1;
    outfile1.open("Expertop.dat", ios::out);  //写文件 Expertop.dat
    outfile1 << Expertop;
    outfile.open("Exper_data.dat", ios::binary);
    if (!outfile)
    {
        cerr << "open error!" << endl; return;
    }
    for (i = 0; i < Expertop; i++)
        outfile.write((char *)&Exper[i], sizeof(Exper[i]));
    outfile.close();
    cout << "保存成功！" << endl;
}
void Experiment::Read()
{
```

```
    int i;
    ifstream infile, infile1;
    infile1.open("Expertop.dat", ios::in);
    infile1 >> Expertop;
    infile.open("Exper_data.dat", ios::binary);
    if (!infile)
    {
        cerr << "open error!" << endl; return;
    }
    for (i = 0; i < Expertop; i++)
        infile.read((char *)&Exper[i], sizeof(Exper[i]));
    infile.close();
    cout << "读取成功！" << endl;
}
void Experiment::Seachid()
{
    char sid[10];
    cout << "请输入编号" << endl; cin >> sid;
    int i;
    for (i = 0; i < Expertop; i++)
    {
        if (!strcmp(Exper[i].id, sid))
        {
            Exper[i].Showone();
            break;
        }
    }
    if (i == Expertop)
    {
        cout << "该编号不存在" << endl;
    }
}
void Experiment::Seachname()
{
    char sname[5];
    cout << "请输入姓名" << endl; cin >> sname;
    int i;
    for (i = 0; i < Expertop; i++)
    {
        if (!strcmp(Exper[i].name, sname))
        {
            Exper[i].Showone();
            break;
        }
```

```
    }
    if (i == Expertop)
    {
        cout << "该姓名不存在" << endl;
    }
}
void Experiment::Showone()
{
    cout << "编号\t" << id << "\t 姓名\t" << name << "\t 性别\t" << sex
<< "\t 年龄\t" << age << "\t 实验室\t" << e_location << "\t 职务\t" << e_job
<< endl;
}
```

3.3 Politician 类

```
//行政人员类
class Politician :virtual public Person
{
    friend class Tea_Po;//声明友元，教师兼行政人员类访问行政人员私有变
量
public:
    void Add();//添加
    void Seach();//查询
    void Show();//显示
    void Edit();//编辑
    void Delete();//删除
    void Save();//保存
    void Read();//读取
    void Seachid();//编号查询
    void Seachname();//姓名查询
    void Showone();
private:
    char p_look[10];//政治面貌
    char p_post[10];//职称
}; Politician Po[MAX]; int Potop = 0;

void Politician::Add()
{
    Politician p_temp;//临时变量
    if (Potop < MAX)
    {
        cout << "编号:"; cin >> p_temp.id;
        for (int i = 0; i < Potop; i++)
        {
```

```
            if (!strcmp(Po[i].id, p_temp.id))
            {
                cout << "id 已存在" << endl;
                return;
            }
        }
        cout << "姓名:"; cin >> p_temp.name;
        cout << "性别:"; cin >> p_temp.sex;
        cout << "年龄:"; cin >> p_temp.age;
        cout << "政治面貌:"; cin >> p_temp.p_look;
        cout << "职称:"; cin >> p_temp.p_post;
        Po[Potop] = p_temp;
        Potop++;
    }
    else
    {
        cout << "当前行政人员已招满" << endl;
    }
}
void Politician::Seach()
{
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |————————————————————————————|" << endl;
        cout << "  |  行政人员管理平台-查询     |" << endl;
        cout << "  |   1.编号查询               |" << endl;
        cout << "  |   2.姓名查询               |" << endl;
        cout << "  |   0.退出                   |" << endl;
        cout << "  |————————————————————————————|" << endl << endl;
        cout << "请选择"; cin >> c;
        switch (c)
        {
        case 1:
            Seachid();
            break;
        case 2:
            Seachname();
            break;
        case 0:
            return;
            break;
        default:
```

```
            cout << "输入错误请重新输入" << endl;
            break;
        }
    }
}
void Politician::Show()
{
    if (Potop == 0)
    {
        cout << "当前行政人员系统内为空" << endl;
    }
    else
    {
        for (int i = 0; i < Potop; i++)
        {
            Po[i].Showone();
        }
    }
}
void Politician::Edit()
{
    Politician p_temp;
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |————————————————————————————|" << endl;
        cout << "  |  行政人员管理平台-编辑     |" << endl;
        cout << "  |   1.根据编号查询并修改     |" << endl;
        cout << "  |   2.根据姓名查询并修改     |" << endl;
        cout << "  |   0.退出                   |" << endl;
        cout << "  |————————————————————————————|" << endl << endl;
        cout << "请选择"; cin >> c;
        if (c == 1)
        {
            char sid[10];
            int i;
            cout << "请输入编号"; cin >> sid;
            for (i = 0; i < Potop; i++)
            {
                if (!strcmp(Po[i].id, sid))
                    break;
            }
            if (i == Potop)
```

```
            cout << "输入编号无效，系统内不存在" << endl;
        else
        {
            cout << "编号:"; cin >> p_temp.id;
            cout << "姓名:"; cin >> p_temp.name;
            cout << "性别:"; cin >> p_temp.sex;
            cout << "年龄:"; cin >> p_temp.age;
            cout << "政治面貌:"; cin >> p_temp.p_look;
            cout << "职称:"; cin >> p_temp.p_post;
            Po[i] = p_temp;
        }
        break;
    }
    else if (c == 2)
    {
        char sname[5];
        int i;
        cout << "请输入姓名"; cin >> sname;
        for (i = 0; i < Potop; i++)
        {
            if (!strcmp(Po[i].name, sname))
                break;
        }
        if (i == Potop)
            cout << "输入编号无效，系统内不存在" << endl;
        else
        {
            cout << "编号:"; cin >> p_temp.id;
            cout << "姓名:"; cin >> p_temp.name;
            cout << "性别:"; cin >> p_temp.sex;
            cout << "年龄:"; cin >> p_temp.age;
            cout << "政治面貌:"; cin >> p_temp.p_look;
            cout << "职称:"; cin >> p_temp.p_post;
            Po[i] = p_temp;
        }
        break;
    }
    else if (c == 0)
    {
        break;
    }
    else
    {
        cout << "输入错误请重新选择" << endl;
```

```
        }
    }
}
void Politician::Delete()
{
    if (Potop == 0)
    {
        cout << "当前记录为空" << endl;
        return;
    }
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |————————————————————————————|" << endl;
        cout << "  |                            |" << endl;
        cout << "  |   行政人员管理平台-删除    |" << endl;
        cout << "  |   1.根据编号查询并删除     |" << endl;
        cout << "  |   2.根据姓名查询并删除     |" << endl;
        cout << "  |   0.退出                   |" << endl;
        cout << "  |————————————————————————————|" << endl << endl;
        cout << "请选择"; cin >> c;
        if (c == 1)
        {
            char sid[10];
            int i;
            cout << "请输入编号"; cin >> sid;
            for (i = 0; i < Potop; i++)
            {
                if (!strcmp(Po[i].id, sid))
                    break;
            }
            if (i == Potop)
                cout << "输入编号无效，系统内不存在" << endl;
            else
            {
                for (i; i < Potop; i++)
                {
                    Po[i] = Po[i + 1];
                }
                cout << "删除成功" << endl;
                Potop--;
            }
            break;
        }
```

```
        else if (c == 2)
        {
            char sname[5];
            int i;
            cout << "请输入姓名"; cin >> sname;
            for (i = 0; i < Potop; i++)
            {
                if (!strcmp(Po[i].name, sname))
                    break;
            }
            if (i == Potop)
                cout << "输入姓名无效，系统内不存在" << endl;
            else
            {
                for (i; i < Potop; i++)
                {
                    Po[i] = Po[i + 1];
                }
                cout << "删除成功" << endl;
                Potop--;
            }
            break;
        }
        else if (c == 0)
        {
            break;
        }
        else
        {
            cout << "输入错误请重新选择" << endl;
        }
    }
}

void Politician::Save()
{
    int i;
    ofstream outfile, outfile1;
    outfile1.open("Potop.dat", ios::out); //写文件 Potop.dat
    outfile1 << Potop;
    outfile.open("Po_data.dat", ios::binary);
    if (!outfile)
    {
        cerr << "open error!" << endl; return;
```

```
    }
    for (i = 0; i < Potop; i++)
        outfile.write((char *)&Po[i], sizeof(Po[i]));
    outfile.close();
    cout << "保存成功！" << endl;
}
void Politician::Read()
{
    int i;
    ifstream infile, infile1;
    infile1.open("Potop.dat", ios::in);
    infile1 >> Potop;
    infile.open("Po_data.dat", ios::binary);
    if (!infile)
    {
        cerr << "open error!" << endl; return;
    }
    for (i = 0; i < Potop; i++)
        infile.read((char *)&Po[i], sizeof(Po[i]));
    infile.close();
    cout << "读取成功！" << endl;
}
void Politician::Seachid()
{
    char sid[10];
    cout << "请输入编号" << endl; cin >> sid;
    int i;
    for (i = 0; i < Potop; i++)
    {
        if (!strcmp(Po[i].id, sid))
        {
            Po[i].Showone();
            break;
        }
    }
    if (i == Potop)
    {
        cout << "该编号不存在" << endl;
    }
}
void Politician::Seachname()
{
    char sname[5];
    cout << "请输入姓名" << endl; cin >> sname;
```

```
    int i;
    for (i = 0; i < Potop; i++)
    {
        if (!strcmp(Po[i].name, sname))
        {
            Po[i].Showone();
            break;
        }
    }
    if (i == Potop)
    {
        cout << "该姓名不存在" << endl;
    }
}
void Politician::Showone()
{
    cout << "编号\t" << id << "\t 姓名\t" << name << "\t 性别\t" << sex
<< "\t 年龄\t" << age << "\t 政治面貌\t" << p_look << "\t 职称\t" << p_post
<< endl;
}
```

3.4 Tea_Po 类

```
//教师兼行政人员类
class Tea_Po :public Teacher, Politician
{
public:
    void Add();//添加
    void Seach();//查询
    void Show();//显示
    void Edit();//编辑
    void Delete();//删除,调用
    void Save();//保存
    void Read();//读取
    void Seachid();//编号查询
    void Seachname();//姓名查询
    void Showone();
}; Tea_Po T_P[MAX]; int T_Ptop = 0;


void Tea_Po::Add()
{
    Tea_Po tp_temp;//临时变量
```

```
    if (T_Ptop < MAX)
    {
        cout << "编号:"; cin >> tp_temp.id;
        for (int i = 0; i < T_Ptop; i++)
        {
            if (!strcmp(T_P[i].id, tp_temp.id))
            {
                cout << "id 已存在" << endl;
                return;
            }
        }
        cout << "姓名:"; cin >> tp_temp.name;
        cout << "性别:"; cin >> tp_temp.sex;
        cout << "年龄:"; cin >> tp_temp.age;
        cout << "系部:"; cin >> tp_temp.t_dept;
        cout << "专业:"; cin >> tp_temp.t_major;
        cout << "政治面貌:"; cin >> tp_temp.p_look;
        cout << "职称:"; cin >> tp_temp.p_post;
        T_P[T_Ptop] = tp_temp;
        T_Ptop++;
    }
    else
    {
        cout << "当前教师兼行政人员已招满" << endl;
    }
}
void Tea_Po::Seach()
{
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |-------------------------------|" << endl;
        cout << "  |  教师兼行政人员管理平台-查询      |" << endl;
        cout << "  |   1.编号查询                  |" << endl;
        cout << "  |   2.姓名查询                  |" << endl;
        cout << "  |   0.退出                      |" << endl;
        cout << "  |-------------------------------|" << endl << endl;
        cout << "请选择"; cin >> c;
        switch (c)
        {
        case 1:
            Seachid();
            break;
```

```
        case 2:
            Seachname();
            break;
        case 0:
            return;
            break;
        default:
            cout << "输入错误请重新输入" << endl;
            break;
        }
    }
}
void Tea_Po::Show()
{
    if (T_Ptop == 0)
    {
        cout << "当前教师兼行政人员系统内为空" << endl;
    }
    else
    {
        for (int i = 0; i < T_Ptop; i++)
        {
            T_P[i].Showone();
        }
    }
}
void Tea_Po::Edit()
{
    Tea_Po tp_temp;
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |_____________________________________|" << endl;
        cout << "  |  教师兼行政人员管理平台-编辑        |" << endl;
        cout << "  |    1.根据编号查询并修改             |" << endl;
        cout << "  |    2.根据姓名查询并修改             |" << endl;
        cout << "  |    0.退出                           |" << endl;
        cout << "  |_____________________________________|" << endl << endl;
        cout << "请选择"; cin >> c;
        if (c == 1)
        {
            char sid[10];
            int i;
```

```
            cout << "请输入编号"; cin >> sid;
            for (i = 0; i < T_Ptop; i++)
            {
                if (!strcmp(T_P[i].id, sid))
                    break;
            }
            if (i == T_Ptop)
                cout << "输入编号无效，系统内不存在" << endl;
            else
            {
                cout << "编号:"; cin >> tp_temp.id;
                cout << "姓名:"; cin >> tp_temp.name;
                cout << "性别:"; cin >> tp_temp.sex;
                cout << "年龄:"; cin >> tp_temp.age;
                cout << "系部:"; cin >> tp_temp.t_dept;
                cout << "专业"; cin >> tp_temp.t_major;
                cout << "政治面貌:"; cin >> tp_temp.p_look;
                cout << "职称:"; cin >> tp_temp.p_post;
                T_P[i] = tp_temp;
            }
            break;
        }
        else if (c == 2)
        {
            char sname[5];
            int i;
            cout << "请输入姓名"; cin >> sname;
            for (i = 0; i < T_Ptop; i++)
            {
                if (!strcmp(T_P[i].name, sname))
                    break;
            }
            if (i == T_Ptop)
                cout << "输入姓名无效，系统内不存在" << endl;
            else
            {
                cout << "编号:"; cin >> tp_temp.id;
                cout << "姓名:"; cin >> tp_temp.name;
                cout << "性别:"; cin >> tp_temp.sex;
                cout << "年龄:"; cin >> tp_temp.age;
                cout << "系部:"; cin >> tp_temp.t_dept;
                cout << "专业"; cin >> tp_temp.t_major;
                cout << "政治面貌:"; cin >> tp_temp.p_look;
                cout << "职称:"; cin >> tp_temp.p_post;
```

```
                T_P[i] = tp_temp;
            }
            break;
        }
        else if (c == 0)
        {
            break;
        }
        else
        {
            cout << "输入错误请重新选择" << endl;
        }
    }
}
void Tea_Po::Delete()
{
    if (T_Ptop == 0)
    {
        cout << "当前记录为空" << endl;
        return;
    }
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |————————————————————————————|" << endl;
        cout << "  |   教师兼行政人员管理平台-删除|" << endl;
        cout << "  |   1.根据编号查询并删除       |" << endl;
        cout << "  |   2.根据姓名查询并删除       |" << endl;
        cout << "  |   0.退出                     |" << endl;
        cout << "  |————————————————————————————|" << endl << endl;
        cout << "请选择"; cin >> c;
        if (c == 1)
        {
            char sid[10];
            int i;
            cout << "请输入编号"; cin >> sid;
            for (i = 0; i < T_Ptop; i++)
            {
                if (!strcmp(T_P[i].id, sid))
                    break;
            }
            if (i == T_Ptop)
                cout << "输入编号无效，系统内不存在" << endl;
```

```
            else
            {
                for (i; i < T_Ptop; i++)
                {
                    T_P[i] = T_P[i + 1];
                }
                cout << "删除成功" << endl;
                T_Ptop--;
            }
            break;
        }
        else if (c == 2)
        {
            char sname[5];
            int i;
            cout << "请输入姓名"; cin >> sname;
            for (i = 0; i < T_Ptop; i++)
            {
                if (!strcmp(T_P[i].name, sname))
                    break;
            }
            if (i == T_Ptop)
                cout << "输入姓名无效，系统内不存在" << endl;
            else
            {
                for (i; i < T_Ptop; i++)
                {
                    T_P[i] = T_P[i + 1];
                }
                cout << "删除成功" << endl;
                T_Ptop--;
            }
            break;
        }
        else if (c == 0)
        {
            break;
        }
        else
        {
            cout << "输入错误请重新选择" << endl;
        }
    }
}
```

```
void Tea_Po::Save()
{
    int i;
    ofstream outfile, outfile1;
    outfile1.open("T_Ptop.dat", ios::out); //写文件 T_Ptop.dat
    outfile1 << T_Ptop;
    outfile.open("T_P_data.dat", ios::binary);
    if (!outfile)
    {
        cerr << "open error!" << endl; return;
    }
    for (i = 0; i < T_Ptop; i++)
        outfile.write((char *)&T_P[i], sizeof(T_P[i]));
    outfile.close();
    cout << "保存成功！" << endl;
}
void Tea_Po::Read()
{
    int i;
    ifstream infile, infile1;
    infile1.open("T_Ptop.dat", ios::in);
    infile1 >> T_Ptop;
    infile.open("T_P_data.dat", ios::binary);
    if (!infile)
    {
        cerr << "open error!" << endl; return;
    }
    for (i = 0; i < T_Ptop; i++)
        infile.read((char *)&T_P[i], sizeof(T_P[i]));
    infile.close();
    cout << "读取成功！" << endl;
}
void Tea_Po::Seachid()
{
    char sid[10];
    cout << "请输入编号" << endl; cin >> sid;
    int i;
    for (i = 0; i < T_Ptop; i++)
    {
        if (!strcmp(T_P[i].id, sid))
        {
            T_P[i].Showone();
            break;
```

```
        }
    }
    if (i == T_Ptop)
    {
        cout << "该编号不存在" << endl;
    }
}
void Tea_Po::Seachname()
{
    char sname[5];
    cout << "请输入姓名" << endl; cin >> sname;
    int i;
    for (i = 0; i < T_Ptop; i++)
    {
        if (!strcmp(T_P[i].name, sname))
        {
            T_P[i].Showone();
            break;
        }
    }
    if (i == T_Ptop)
    {
        cout << "该姓名不存在" << endl;
    }
}
void Tea_Po::Showone()
{
    cout << "编号\t" << id << "\t 姓名\t" << name << "\t 性别\t" << sex
<< "\t 年龄\t" << age << "\t 系部\t" << t_dept << "\t 专业" << t_major <<
"\t 政治面貌\t" << p_look << "\t 职称\t" << p_look << endl;
}
```

3.5 教师菜单

```
void Tea_menu(Teacher tea)
{
    while (true)
    {
        system("pause");
        system("cls");//清除主菜单
        int c;
        cout << endl;
        cout << "  |————————————————————————————————|" << endl;
```

```
cout << "  |       教师管理平台                |" << endl;
cout << "  |   1.添加                          |" << endl;
cout << "  |   2.查询                          |" << endl;
cout << "  |   3.显示                          |" << endl;
cout << "  |   4.编辑                          |" << endl;
cout << "  |   5.删除                          |" << endl;
cout << "  |   6.统计                          |" << endl;
cout << "  |   7.保存                          |" << endl;
cout << "  |   8.读取                          |" << endl;
cout << "  |   0.退出                          |" << endl;
cout << "  |___________________________________|" << endl << endl;
cout << "请选择"; cin >> c;
switch (c)
{
case 1:
    tea.Add();
    break;
case 2:
    tea.Seach();
    break;
case 3:
    tea.Show();
    break;
case 4:
    tea.Edit();
    break;
case 5:
    tea.Delete();
    break;
case 6:
    Statistics();
    break;
case 7:
    tea.Save();
    break;
case 8:
    tea.Read();
    break;
case 0:
    break;
default:
    cout << "输入错误请重新输入" << endl;
    break;
}
```

```
        if (c == 0)
            break;
    }
}
```

3.6 实验人员菜单

```
void Exper_menu(Experiment exper)
{
    while (true)
    {
        system("pause");
        system("cls");//清除主菜单
        int c;
        cout << endl;
        cout << "  |————————————————————————————————|" << endl;
        cout << "  |          实验人员管理平台          |" << endl;
        cout << "  |    1.添加                          |" << endl;
        cout << "  |    2.查询                          |" << endl;
        cout << "  |    3.显示                          |" << endl;
        cout << "  |    4.编辑                          |" << endl;
        cout << "  |    5.删除                          |" << endl;
        cout << "  |    6.统计                          |" << endl;
        cout << "  |    7.保存                          |" << endl;
        cout << "  |    8.读取                          |" << endl;
        cout << "  |    0.退出                          |" << endl;
        cout << "  |————————————————————————————————|" << endl << endl;
        cout << "请选择"; cin >> c;
        switch (c)
        {
        case 1:
            exper.Add();
            break;
        case 2:
            exper.Seach();
            break;
        case 3:
            exper.Show();
            break;
        case 4:
            exper.Edit();
            break;
        case 5:
```

```
            exper.Delete();
            break;
        case 6:
            Statistics();
            break;
        case 7:
            exper.Save();
            break;
        case 8:
            exper.Read();
            break;
        case 0:
            break;
        default:
            cout << "输入错误请重新输入" << endl;
            break;
        }
        if (c == 0)
            break;
    }
}
```

3.7 行政人员类菜单

```
void P_menu(Politician po)
{
    while (true)
    {
        system("pause");
        system("cls");//清除主菜单
        int c;
        cout << endl;
        cout << "  |-------------------------------------|" << endl;
        cout << "  |          行政人员管理平台           |" << endl;
        cout << "  |    1.添加                           |" << endl;
        cout << "  |    2.查询                           |" << endl;
        cout << "  |    3.显示                           |" << endl;
        cout << "  |    4.编辑                           |" << endl;
        cout << "  |    5.删除                           |" << endl;
        cout << "  |    6.统计                           |" << endl;
        cout << "  |    7.保存                           |" << endl;
        cout << "  |    8.读取                           |" << endl;
        cout << "  |    0.退出                           |" << endl;
```

```
        cout << "  |_____|" << endl << endl;
        cout << "请选择"; cin >> c;
        switch (c)
        {
        case 1:
            po.Add();
            break;
        case 2:
            po.Seach();
            break;
        case 3:
            po.Show();
            break;
        case 4:
            po.Edit();
            break;
        case 5:
            po.Delete();
            break;
        case 6:
            Statistics();
            break;
        case 7:
            po.Save();
            break;
        case 8:
            po.Read();
            break;
        case 0:
            break;
        default:
            cout << "输入错误请重新输入" << endl;
            break;
        }
        if (c == 0)
            break;
    }
}
```

3.8 教师兼行政人员菜单

```
void TeaPo_menu(Tea_Po t_p)
```

```
{
    while (true)
    {
        system("pause");
        system("cls");//清除主菜单
        int c;
        cout << endl;
        cout << "  |————————————————————————————————|" << endl;
        cout << "  |                                |" << endl;
        cout << "  |  教师兼行政人员人员管理平台    |" << endl;
        cout << "  |    1.添加                      |" << endl;
        cout << "  |    2.查询                      |" << endl;
        cout << "  |    3.显示                      |" << endl;
        cout << "  |    4.编辑                      |" << endl;
        cout << "  |    5.删除                      |" << endl;
        cout << "  |    6.统计                      |" << endl;
        cout << "  |    7.保存                      |" << endl;
        cout << "  |    8.读取                      |" << endl;
        cout << "  |    0.退出                      |" << endl;
        cout << "  |————————————————————————————————|" << endl << endl;
        cout << "请选择"; cin >> c;
        switch (c)
        {
        case 1:
            t_p.Add();
            break;
        case 2:
            t_p.Seach();
            break;
        case 3:
            t_p.Show();
            break;
        case 4:
            t_p.Edit();
            break;
        case 5:
            t_p.Delete();
            break;
        case 6:
            Statistics();
            break;
        case 7:
            t_p.Save();
            break;
        case 8:
```

```
            t_p.Read();
            break;
        case 0:
            break;
        default:
            cout << "输入错误请重新输入" << endl;
            break;
        }
        if (c == 0)
            break;
    }
}



void Statistics()
{
    while (true)
    {
        int c;
        cout << endl;
        cout << "  |_____________________________|" << endl;
        cout << "  |        人数统计             |" << endl;
        cout << "  |    1.四类人员数量           |" << endl;
        cout << "  |    2.男员工数量             |" << endl;
        cout << "  |    3.女员工数量             |" << endl;
        cout << "  |    4.某年龄段数量           |" << endl;
        cout << "  |    0.退出                   |" << endl;
        cout << "  |_____________________________|" << endl << endl;
        cout << "请选择"; cin >> c;
        switch (c)
        {
        case 1:
            Statisticstotal();
            break;
        case 2:
            Statisticman();
            break;
        case 3:
            Statisticwomen();
            break;
        case 4:
            Statisticage();
            break;
```

```
        case 0:
            return;
            break;
        default:
            cout << "输入错误请重新输入" << endl;
            break;
        }
    }
}
//四类员工数量
void Statisticstotal()
{
    cout << "四类员工总数量" << Teatop + Expertop + Potop + T_Ptop << endl;
    cout << "教师总数量" << Teatop << endl;
    cout << "实验员总数量" << Expertop << endl;
    cout << "行政人员总数量" << Potop << endl;
    cout << "教师兼行政人员总数量" << T_Ptop << endl;
}
//男员工数量
void Statisticman()
{
    int man = 0;
    for (int i = 0; i < Teatop; i++)
    {
        if (!strcmp(Tea[i].sex, "男"))
            man++;
    }
    for (int i = 0; i < Expertop; i++)
    {
        if (!strcmp(Exper[i].sex, "男"))
            man++;
    }
    for (int i = 0; i < Potop; i++)
    {
        if (!strcmp(Po[i].sex, "男"))
            man++;
    }
    for (int i = 0; i < T_Ptop; i++)
    {
        if (!strcmp(T_P[i].sex, "男"))
            man++;
    }
    cout << "男员工的数量为" << man << endl;
}
```

```
//女员工数量
void Statisticwomen()
{
    int women = 0;
    for (int i = 0; i < Teatop; i++)
    {
        if (!strcmp(Tea[i].sex, "女"))
            women++;
    }
    for (int i = 0; i < Expertop; i++)
    {
        if (!strcmp(Exper[i].sex, "女"))
            women++;
    }
    for (int i = 0; i < Potop; i++)
    {
        if (!strcmp(Po[i].sex, "女"))
            women++;
    }
    for (int i = 0; i < T_Ptop; i++)
    {
        if (!strcmp(T_P[i].sex, "女"))
            women++;
    }
    cout << "女员工的数量为" << women << endl;
}
//年龄段统计
void Statisticage()
{
    int age1, age2;
    cout << "年龄区间:age1="; cin >> age1;
    cout << "年龄区间:age2="; cin >> age2;
    int num = 0;
    for (int i = 0; i < Teatop; i++)
    {
        if (Tea[i].age > age1&& Tea[i].age < age2)
            num++;
    }
    for (int i = 0; i < Expertop; i++)
    {
        if (Exper[i].age > age1&&Exper[i].age < age2)
            num++;
    }
    for (int i = 0; i < Potop; i++)
```

```
    {
        if (Po[i].age > age1&&Po[i].age < age2)
            num++;
    }
    for (int i = 0; i < T_Ptop; i++)
    {
        if (T_P[i].age > age1&&T_P[i].age < age2)
            num++;
    }
    cout << age1 << "~" << age2 << "的数量为" << num << endl;
}
```

四、 项目总结

（写项目中应用到的知识点）

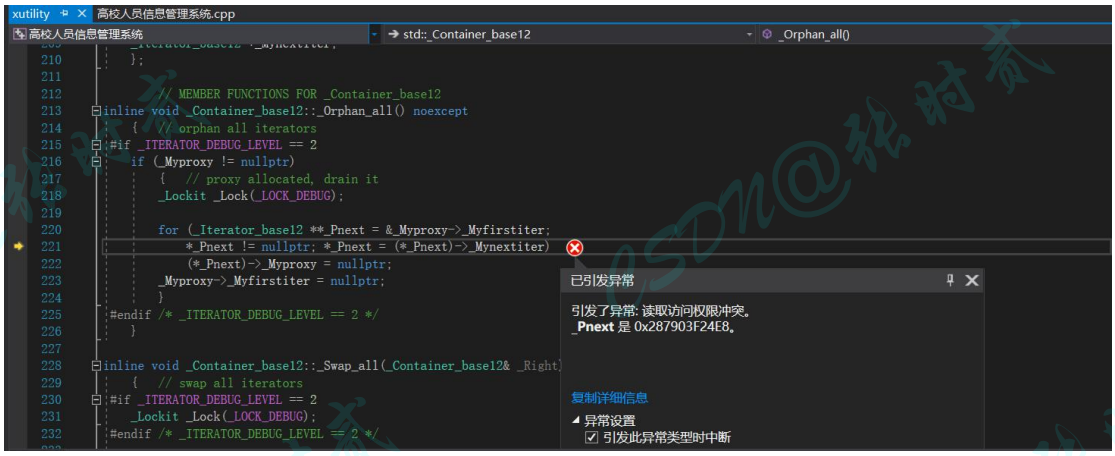
本次课设主要利用了C++中面向对象设计中类的设计和类的继承，处理好各个类之间的父子孙关系，以及利用虚继承解决菱形继承导致的参数不明确问题

五、 参考文献

（至少有一篇除教材之外的参考文献）

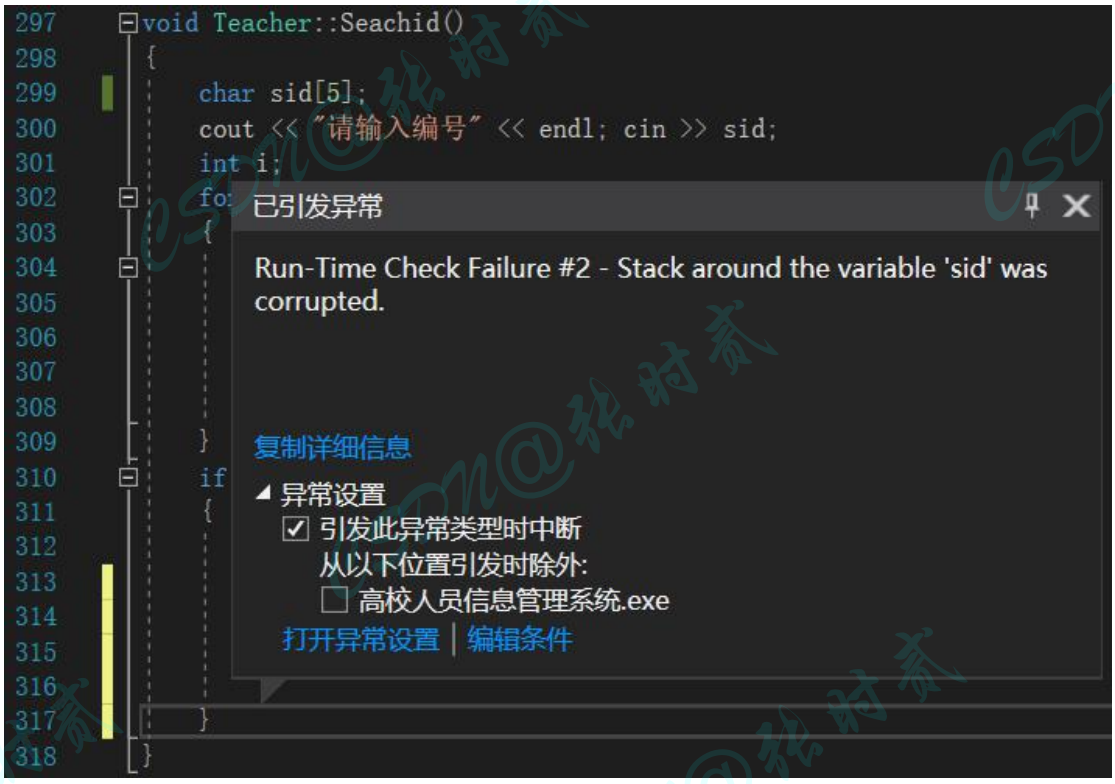
在每次读取完文件退出主程序时，出现 xutility 错误，因为 string 对象维护的是一段内存，而读入二进制文件保存的是内存地址，在下一次读取时这段内存地址发生变化，所以出现错误。解决办法，将四个人员类中的变量定义为 char 类型的数组

C++类对象进行文件读取后出现的 xutility 错误



解决 string 类型读写问题后，代码没有报错但是一旦运行检索类的功能，出现变量堆栈损坏，数组中输入的字符超过了定义的长度，所以各个 char 类型长度要合理定义

Run-Time Check Failure #2 - Stack around the variable 'XXX' was corrupted.错误的解决方法



注意：项目模板格式不得随意改写，制作报告完成后更新目录。