

成绩：_____

C++程序设计 A2

课 程 设 计 报 告

课程名称 C++程序设计 A2

课题名称 高校人员信息管理系统

专 业 软件工程

班 级 1720561

学 号 172056114

姓 名 岳立军

指导教师 李燕香

2019 年 1 月 2 日

（一）、课程设计题目：

高校人员信息管理系统

（二）、目的与要求：

1、目的：

- （1）要求学生达到熟练掌握 C++语言的基本知识和技能；
- （2）基本掌握面向对象程序设计的基本思路和方法；
- （3）能够利用所学的基本知识和技能，解决简单的面向对象程序设计问题。

2、基本要求：

- （1）要求利用面向对象的方法以及 C++的编程思想来完成系统的设计；
- （2）要求在设计的过程中，建立清晰的类层次；
- （3）在系统中至少要定义四个类，每个类中要有各自的属性和方法；
- （4）在系统的设计中，至少要用到面向对象的一种机制。

3、创新要求：

在基本要求达到后，可进行创新设计，如根据查找结果进行修改的功能。

（三）、设计方法和基本原理：

功能要求：

（1）基础信息的添加功能

程序能够任意添加高校人员信息、班级信息、课程信息和职务信息。

（2）修改功能

对已添加的记录（高校人员信息、班级信息、课程信息和职务信息）进行修改。

（3）删除功能

对已添加的记录（高校人员信息、班级信息、课程信息和职务信息）进行删除。

如果当前系统中没有相应的记录，则提示“记录为空！”并返回操作；否则输入要删除的编号对该记录进行删除，如果没有找到，则提示相应的记录不存在。

（4）查询功能

①可根据高校人员的班号查询某班的所有人员信息。

②可根据高校人员的学号查询某教师的所有课程信息。

如果未找到，给出相应的提示信息，如果找到，则显示相应的记录信息。

（5）保存功能

将当前系统中各类记录存入文件中。

（6）读取功能

将保存在文件中的信息读入到当前系统中，以供用户使用。

3、问题的解决方案：

根据系统功能要求，可以将问题解决分为以下步骤：

- （1）应用系统分析，建立该系统的功能模块框图以及界面的组织和设计；
- （2）分析系统中的各个实体及它们之间的关系包括属性和行为；
- （3）根据问题描述，设计系统的类层次；
- （4）完成类层次中各个类的描述（包括属性和方法）；
- （5）完成类中各个成员函数的定义；
- （6）完成系统的应用模块；
- （7）功能调试；
- （8）完成系统总结报告以及系统使用说明书

目 录

1. 系统需求分析.....	4
2. 总体设计.....	5
3. 详细设计及实现（附算法流程图和运行结果截图）	8
3.1 Person 类(抽象类)的实现.....	8
.....	8
3.2 教师类的实现.....	8
.....	15
3.3 行政人员类的实现.....	15
3.4 实验员类的实现.....	17
3.5 教师兼行政人员类的实现.....	19
3.6 main()	21
4. 系统调试.....	25
5. 结论.....	26
6. 心得体会.....	26
7. 参考书目与网址.....	26

1. 系统需求分析

1.1 系统设计的目的与意义：

现在的高校人员数量较多，职务也分门别类，因此需要一个高校人员管理系统来对职工的各种信息进行管理。

1.2 系统功能需求：

本系统有五个类：一个基本类——person 类用于存储高校人员的基本信息，四个派生类——Teacher 类、staff 类、Experimenter 类、Tea_Sta 类，用于存储教师的个人信息、行政人员的个人信息、实验员的个人信息、教师兼行政人员的个人信息。

上述的四个大类中，每个大类都附有以下所述功能：

1、基础信息的添加功能

程序能够任意添加高校人员信息、班级信息、课程信息和职务信息。

2、显示功能

程序可以显示所写入的所有高校人员个人信息，直观的表现出来，易于查看。

3、查询功能

①可根据高校人员的编号查询某班的该人员信息。

②可根据高校人员的姓名查询该职工的所有个人信息。

如果未找到，给出相应的提示信息，如果找到，则显示相应的记录信息。

4、修改功能

对已添加的记录（高校人员信息、班级信息、课程信息和职务信息）进行修改。

5、删除功能

对已添加的记录（高校人员信息、班级信息、课程信息和职务信息）进行删除。

如果当前系统中没有相应的记录，则提示“记录为空！”并返回操作；否则输入要删除的编号对该记录进行删除，如果没有找到，则提示相应的记录不存在。

6、统计功能

程序可以统计高校人员的一些基本信息，可分为按性别统计和职务统计。

7、保存功能

将当前系统中各类记录存入文件中。

8、读取功能

将保存在文件中的信息读入到当前系统中，以供用户使用。

0、退出功能

2. 总体设计

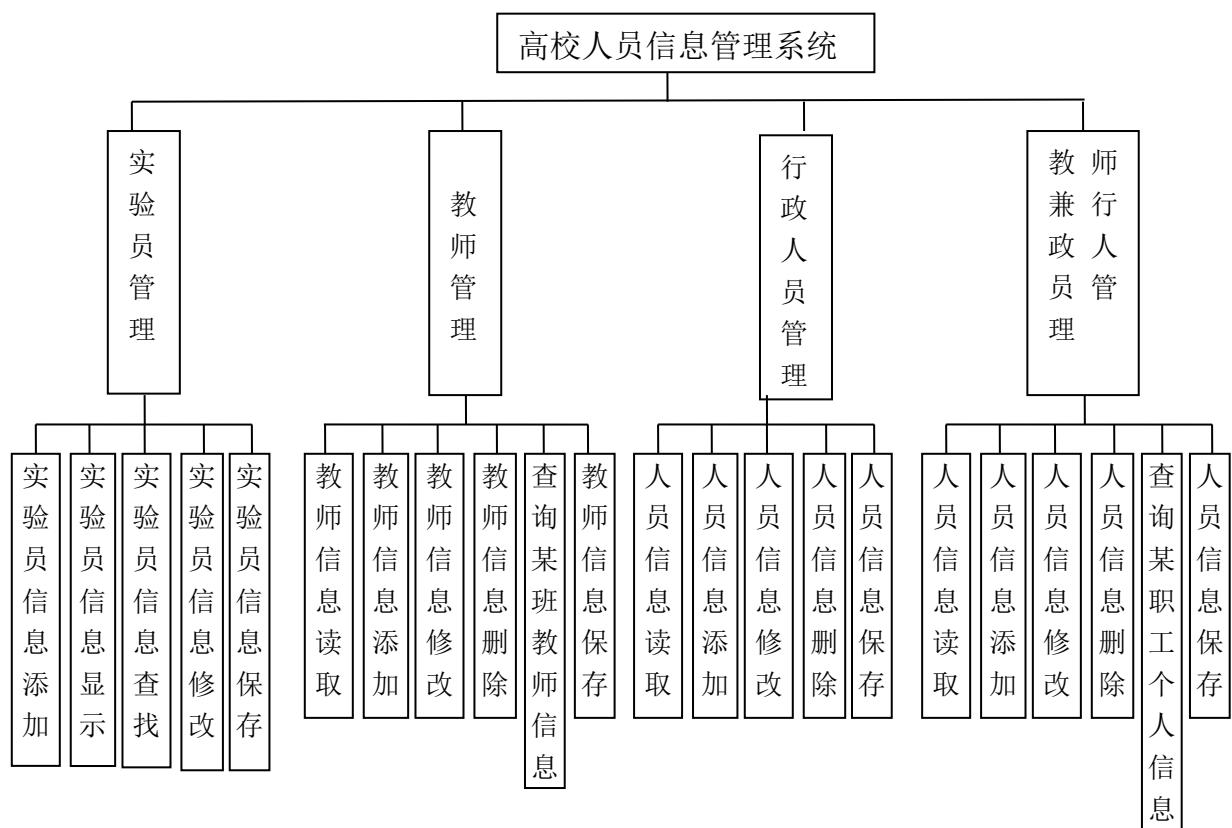
2.1 系统功能分析：

系统通过调用函数来实现信息录入、信息修改、信息删除、信息查询以及数据保存文件等功能。add（）函数用于录入信息；show（）函数用于显示高校人员信息；search（）函数用来查询某班级中该高校人员信息；Edit（）函数用来修改信息；delete（）函数用来删除信息；total（）函数用于统计人员数目情况；save（）函数用于将内存中的信息存入文件内；read（）函数用于将文件中的数据读取到内存中；主函数main（）用于调用这些函数。

2.2 系统功能模块划分与设计：

系统定义了Teacher类、staff类、Experimenter类、Tea_Sta类这四个类，定义了add（）、show（）、search（）、edit（）、delete（）、total（）、save（）、read（）、input（）及output（）等函数来实现系统所需功能。

2.3 系统功能模块图：



附：除基本增删改查功能之外，还有信息显示、信息统计、信息读取功能，空间不足，略以表述。

2.4 类的设计

(1) Person 类（抽象类）的设计：

Person
Person (); （构造函数） virtual void input(); virtual void output(int); Int GetNum() {return num}; StringGetName() {return name};

图 3.1 person 类

(2) 教师类的设计：

Teaher
String depart; String major; String title; Int choicel; Int choice2; Class Tea manage;
void input(); void output();

图 3.2 教师类

(3) 行政人员类的设计：

图 3.3 行政人员类

Staff
String zzmm; String zc; Int choice2; Class staf_Manage;
Staff(); void input(); void output();

(4) 实验员类的设计:

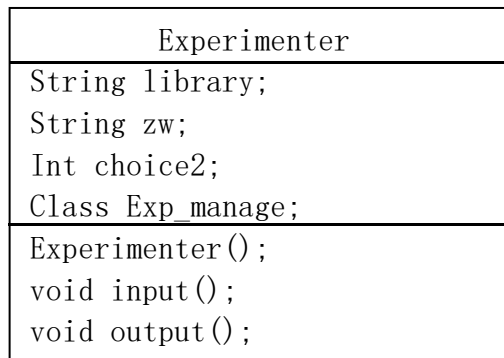


图 3.4 实验员类

(5) 教师兼行政人员类的设计:

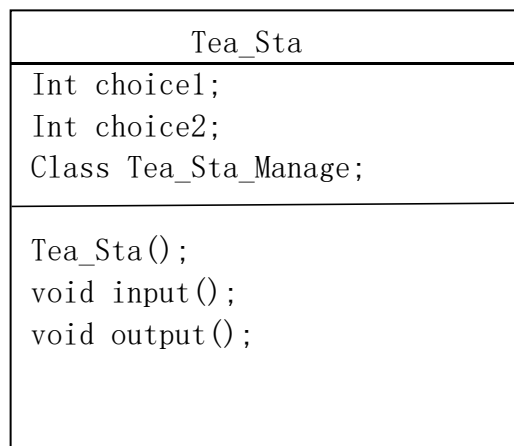
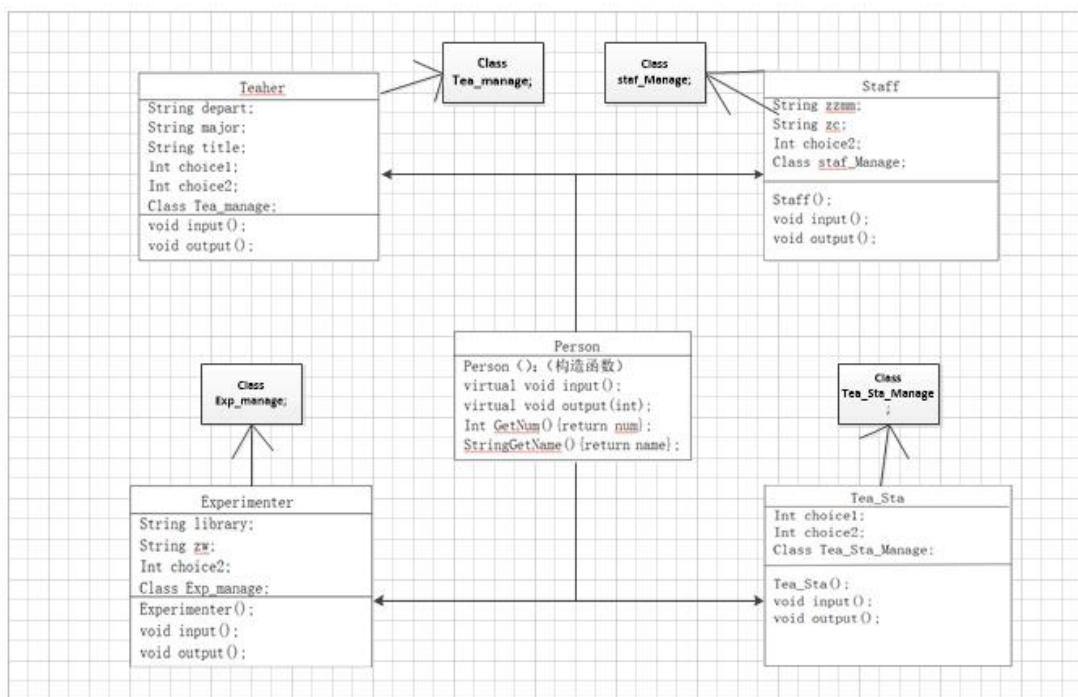


图 3.5 教师兼行政人员类

(6) 各个类的关系描述:



3. 详细设计及实现（附算法流程图和运行结果截图）

3.1 Person 类(抽象类)的实现

```
class Person
{
public:
    int num;
    string name;
    string sex;
    int age;
    Person(int n = 0, string nm = "", string s = "", int a = 0) :num(n), name(nm), sex(s), age(a)
    {};
    virtual void Input() {};
    virtual void output() {};
    int GetNum() { return num; }
    string GetName() { return name; }
};
```

3.2 教师类的实现

Teacher 类:

```
class Teacher :virtual public Person
{
public:
    string depart;
    string major;
    string title;
    int choice1;
    int choice2;
    Teacher(int n = 0, string nm = "", string s = "男", int a = 0, string d = "计算机", string ma
    = "", string t = "") :Person(n, nm, s, a), depart(d), major(ma), title(t) {};
    //输入函数 void Input()
    {
        cout << "请输入编号:" << endl;
        cin >> num;
        cout << "请输入名字:" << endl;
        cin >> name;
        cout << "请输入性别:" << endl;
        cout << "(1.男 2.女)" << endl;
    }
};
```

```

        cout << "请输入你的选择:" << endl;
        cin >> choice2;
        switch (choice2)
        {
            case 1:sex = "男"; break;
            case 2:sex = "女"; break;
            default:cout << "输入错误!" << endl; break;
        }
        cout << "请输入年龄:" << endl;
        cin >> age;
        cout << "请输入所在系别:" << setw(10) << "1. 计算机工程系" << setw(10) << "2. 电子工程系"
        << setw(10) << "3. 经管系" << setw(10) << "4. 外语系" << setw(10) << "5. 环境与安全系" << endl;
        cout << "请输入你的选择:" << endl;
        cin >> choice1;
        switch (choice1)
        {
            case 1:depart = "计算机工程系"; break;
            case 2:depart = "电子工程系"; break;
            case 3:depart = "经管系"; break;
            case 4:depart = "外语系"; break;
            case 5:depart = "环境与安全系"; break;
            default:cout << "输入错误!" << endl; break;
        }
        cout << "请输入专业名称:" << endl;
        cin >> major;
        cout << "请输入职称:" << endl;
        cin >> title;
    }
    //输出函数 void Output()
    {
        cout << setw(10) << num << setw(10) << name << setw(10) << sex << setw(10) << age << setw(10)
        << depart << setw(10) << major << setw(10) << title << endl;
    }
};

```

Tea_Manage 类:

```

class Tea_Manage
{
    Teacher Tea[N];
    int top;
public:
    Tea_Manage()
    {
        top = 0;
    }
}

```

```
        cout << "请输入教师人数:";
        cin >> top;
        cout << "请输入教师信息:" << endl;
        cout << "编号 姓名 性别 年龄 系别 专业 职称" << endl;
        for (int i = 0; i < top; i++)
            cin >> Tea[i].num >> Tea[i].name >> Tea[i].sex >> Tea[i].age >> Tea[i].depart >>
Tea[i].major >> Tea[i].title;
    }
    //成员函数 void Add();
    void Show();
    void Search();
    void Edit();
    void Delete();
    void Total();
    void Save();
    void Read();
};
//增加 void Tea_Manage::Add()
{
    if (top >= N)
    {
        cout << "用户已满" << endl;
        return;
    }
    Teacher t;
    cout << "输入新添加的人员的信息: " << endl;
    t.Input();
    for (int i = 0; i < top; i++)
    {
        if (t.GetNum() == Tea[i].GetNum())
            cout << "该编号的人已存在" << endl;
        break;
    }
    Tea[top] = t;
    top = top + 1;
    cout << "添加成功!" << endl;
}
//显示 void Tea_Manage::Show()
{
    if (top == 0)
    {
        cout << "无数据!" << endl;
        return;
    }
}
```

```
        cout << setw(10) << "编号" << setw(10) << "姓名" << setw(10) << "性别" << setw(10) << "年龄"
<< setw(10) << "系部" << setw(10) << "专业" << setw(10) << "职务" << endl;
        for (int i = 0; i < top; i++)
            Tea[i].Output();
    }
//搜索 void Tea_Manage::Search()
{
    if (top == 0)
    {
        cout << "当前系统中没有存储记录!" << endl;
        return;
    }
    int choice;
    cout << "请选择查询方式: 1. 按编号查找 2. 按姓名查找" << endl;
    cin >> choice;
    switch (choice)
    {
    case 1:
    {
        cout << "请输入要查找的编号:" << endl;
        int num;
        cin >> num;
        for (int i = 0; i < top; i++)
        {
            if (Tea[i].GetNum() == num)
            {
                Tea[i].Output();
                return;
            }
        }
        cout << "查无此人!" << endl;
    } break;
    case 2:
    {
        cout << "请输入要查找的姓名:" << endl;
        string name;
        cin >> name;
        for (int i = 0; i < top; i++)
        {
            if (Tea[i].GetName() == name)
            {
                Tea[i].Output();
            }
        }
    }
}
```

```
        for (int i = 0; i < top; i++)
            if (Tea[i].GetName() == name)
                return;
        cout << "查无此人！" << endl;
    } break;
    default: cout << "无此选项！请重试！" << endl; break;
}

//修改 void Tea_Manage::Edit()
{
    if (top == 0)
    {
        cout << "当前系统没有存储记录！" << endl;
        return;
    }
    int num;
    cout << "请输入要查找的编号：" << endl;
    cin >> num;
    for (int i = 0; i < top; i++)
    {
        if (Tea[i].GetNum() == num)
        {
            cout << "请对这位教师的信息进行修改：" << endl;
            Tea[i].Input();
            cout << "修改成功！" << endl;
        }
    }
    cout << "查无此人！" << endl;
}

//删除 void Tea_Manage::Delete()
{
    if (top == 0)
    {
        cout << "记录为空！" << endl;
        return;
    }
    cout << "请选择删除方式：1. 按编号查找删除 2. 按姓名查找删除" << endl;
    int choice;
    cin >> choice;
    switch (choice)
    {
        case 1:
        {
            cout << "请输入编号：" << endl;
```

```
int num;
cin >> num;
for (int i = 0; i < top; i++)
{
    if (num == Tea[i].num)
    {
        cout << "是否确认删除? 1. 是 2, 否" << endl;
        int choice;
        cin >> choice;
        switch (choice)
        {
            case 1:
            {
                for (int j = i; j < top - 1; j++)
                    Tea[j] = Tea[j + 1];
                cout << "删除成功!" << endl;
                top--;
            }break;
            case 2:return;
            default:cout << "无此项!" << endl; break;
        }return;
    }
}
cout << "无此人!" << endl;
}; break;
case 2:
{
    cout << "请输入姓名: " << endl;
    string name;
    cin >> name;
    for (int i = 0; i < top; i++)
    {
        if (name == Tea[i].name)
        {
            cout << "是否确认删除? 1. 是 2, 否" << endl;
            int choice;
            cin >> choice;
            switch (choice)
            {
                case 1:
                {
                    for (int j = i; j < top - 1; j++)
                        Tea[j] = Tea[j + 1];
                    cout << "删除成功!" << endl;
```

```
        top--;
    }; break;
    case 2: return;
    default: cout << "无此项! " << endl; break;
    } return;
    }
}
cout << "无此人! " << endl;
}; break;
default: cout << "无此项! " << endl; break;
}
}
//统计 void Tea_Manage::Total()
{
    cout << "1. 按人数统计  2. 按性别统计  3. 按系别统计" << endl;
    int choice;
    cout << "请输入你的选择" << endl;
    cin >> choice;
    switch (choice)
    {
    case 1: cout << "教师人数为: " << top << endl; break;
    case 2:
    {
        int a = 0, b = 0;
        for (int i = 0; i < top; i++)
        {
            if (Tea[i].sex == "男") a++;
            if (Tea[i].sex == "女") b++;
        }
        cout << "男教师人数为:" << a << endl;
        cout << "女教师人数为:" << b << endl;
    } break;
    case 3:
    {
        int a1 = 0, a2 = 0, a3 = 0, a4 = 0, a5 = 0;
        for (int i = 0; i < top; i++)
        {
            if (Tea[i].depart == "计算机工程系") a1++;
            if (Tea[i].depart == "电子工程系") a2++;
            if (Tea[i].depart == "外语系") a3++;
            if (Tea[i].depart == "经管系") a4++;
            if (Tea[i].depart == "环境与安全系") a5++;
        }
        cout << "计算机工程系教师人数为: " << a1 << endl;
```

```
        cout << "电子工程系教师人数为: " << a2 << endl;
        cout << "外语系教师人数为: " << a3 << endl;
        cout << "经管系教师人数为: " << a4 << endl;
        cout << "环境与安全系教师人数为: " << a5 << endl;
    }break;
    default:cout << "输入错误! " << endl; break;
}
}
//保存 void Tea_Manage::Save()
{
    ofstream ofile("教师.txt");
    if (!ofile)
    {
        cout << "打开失败! " << endl;
        return;
    }
    cout << "打开成功! " << endl;
    for (int i = 0; i < top; i++)
    {
        ofile << Tea[i].num << " " << Tea[i].name << " " << Tea[i].sex << " " << Tea[i].age << "
" << Tea[i].depart << " " << Tea[i].major << " " << Tea[i].title << endl;
    }
    cout << "写入成功! " << endl;
    ofile.close();
}
//读取 void Tea_Manage::Read()
{
    ifstream ifile;
    ifile.open("教师.txt");
    for (int i = 0; i < top; i++)
    {
        ifile >> Tea[i].num >> Tea[i].name >> Tea[i].sex >> Tea[i].age >> Tea[i].depart >>
Tea[i].major >> Tea[i].title;
        Tea[i].Output();
    }
    ifile.close();
};
```

3.3 行政人员类的实现

Staff 类:

```
class Staff :virtual public Person
{
```



```
public:
    string zzmm;
    string zc;
    int choice2;
    Staff(int n = 0, string nm = "", string s = "男", int a = 0, string z = "", string zc = "") : Person(n,
nm, s, a), zzmm(z), zc(zc) {};
    //输入函数 void Input()
    {
        cout << "请输入编号:" << endl;
        cin >> num;
        cout << "请输入名字:" << endl;
        cin >> name;
        cout << "请输入性别:" << endl;
        cout << "(1. 男 2. 女)" << endl;
        cout << "请输入你的选择:" << endl;
        cin >> choice2;
        switch (choice2)
        {
            case 1: sex = "男"; break;
            case 2: sex = "女"; break;
            default: cout << "输入错误!" << endl; break;
        }
        cout << "请输入年龄:" << endl;
        cin >> age;
        cout << "请输入政治面貌:" << endl;
        cin >> zzmm;
        cout << "请输入职称:" << endl;
        cin >> zc;
    }
    //输出函数 void Output()
    {
        cout << setw(10) << num << setw(10) << name << setw(10) << sex << setw(10) << age << setw(10)
<< zzmm << setw(10) << zc << endl;
    }
};
```

Staf_Manage 类:

```
class Staf_Manage
{
    Staff Tea[N];
    int top;
public:
    Staf_Manage()
    {
```

```

        top = 0;
        cout << "请输入行政人员人数:";
        cin >> top;
        cout << "请输入行政人员信息:" << endl;
        cout << "编号 姓名 性别 年龄 政治面貌 职称" << endl;
        for (int i = 0; i < top; i++)
            cin >> Tea[i].num >> Tea[i].name >> Tea[i].sex >> Tea[i].age >> Tea[i].zzmm >>
            Tea[i].zc;
    }
    //成员函数 void Add();
    void Show();
    void Search();
    void Edit();
    void Delete();
    void Total();
    void Save();
    void Read();
};
//函数定义如上例，此处不一一赘述。

```

3. 实验员类的实现

Experimenter 类:

```

class Experimenter :public Person
{
public:
    string library;
    string zw;
    int choice2;
    Experimenter(int n = 0, string nm = "", string s = "男", int a = 0, string l = "", string z =
    "") :Person(n, nm, s, a), library(l), zw(z) {};
    //输入函数 void Input()
    {
        cout << "请输入编号:" << endl;
        cin >> num;
        cout << "请输入名字:" << endl;
        cin >> name;
        cout << "请输入性别:" << endl;
        cout << "(1. 男 2. 女)" << endl;
        cout << "请输入你的选择:" << endl;
        cin >> choice2;
        switch (choice2)
        {
            case 1:sex = "男"; break;

```

```

        case 2:sex = "女"; break;
        default:cout << "输入错误!" << endl; break;
    }

    cout << "请输入年龄:" << endl;
    cin >> age;
    cout << "请输入实验室:" << endl;
    cin >> library;
    cout << "请输入职务:" << endl;
    cin >> zw;
}

//输出函数 void Output()
{
    cout << setw(10) << "编号" << setw(10) << " 姓名" << setw(10) << " 性别" << setw(10) <<
    "年龄" << setw(10) << "实验室" << setw(10) << "职务" << endl;
    cout << setw(10) << num << setw(10) << name << setw(10) << sex << setw(10) << age << setw(10)
    << library << setw(10) << zw << endl;
}
};

```

Exp_Manage 类:

```

class Exp_Manage
{
    Experimenter Tea[N];
    int top;
public:
    Exp_Manage()
    {
        top = 0;
        cout << "请输入实验员人数:";
        cin >> top;
        cout << "请输入实验员的信息:" << endl;
        cout << "编号 姓名 性别 年龄 实验室 职务" << endl;
        for (int i = 0; i < top; i++)
            cin >> Tea[i].num >> Tea[i].name >> Tea[i].sex >> Tea[i].age >> Tea[i].library >>
            Tea[i].zw;
    }
    //成员函数 void Add();
    void Show();
    void Search();
    void Edit();
    void Delete();
    void Total();
    void Save();
    void Read();
}

```

};//函数定义如上例，此处不一一赘述。

3.5 教师兼行政人员类的实现

Tea_Sta 类:

```
class Tea_Sta :public Teacher, public Staff
{
public:
    int choicel;
    int choice2;
    Tea_Sta(int n = 0, string nm = "", string s = "男", int a = 0, string d = "", string ma = "",
string t = "", string z = "", string zc = "") :Person(n, nm, s, a), Teacher(n, nm, s, a, d, ma, t),
Staff(n, nm, s, a, z, zc) {};
    //输入函数 void Input()
    {
        cout << "请输入编号:" << endl;
        cin >> num;
        cout << "请输入名字:" << endl;
        cin >> name;
        cout << "请输入性别:" << endl;
        cout << "(1.男 2.女)" << endl;
        cout << "请输入你的选择:" << endl;
        cin >> choice2;
        switch (choice2)
        {
            case 1:sex = "男"; break;
            case 2:sex = "女"; break;
            default:cout << "输入错误!" << endl; break;
        }
        cout << "请输入年龄:" << endl;
        cin >> age;
        cout << "请输入所在系别:" << endl;
        cin >> depart;
        cout << setw(10) << "1. 计算机工程系" << setw(10) << "2. 电子工程系" << setw(10) << "3. 经
管系" << setw(10) << "4. 外语系" << setw(10) << "5. 环境与安全系" << endl;
        cout << "请输入你的选择:" << endl;
        cin >> choicel;
        switch (choicel)
        {
            case 1:depart = "计算机工程系"; break;
            case 2:depart = "电子工程系"; break;
            case 3:depart = "经管系"; break;
            case 4:depart = "外语系"; break;
```

```

        case 5:depart = "环境与安全系"; break;
        default:cout << "输入错误!" << endl; break;
    }
    cout << "请输入专业名称:" << endl;
    cin >> major;
    cout << "请输入职称:" << endl;
    cin >> title;
    cout << "请输入政治面貌:" << endl;
    cin >> zzmm;
    cout << "请输入职称:" << endl;
    cin >> zc;
}
//输出函数 void Output()
{
    cout << setw(10) << num << setw(10) << name << setw(10) << sex << setw(10) << age << setw(10)
<< depart << setw(10) << major << setw(10) << title << setw(10) << zzmm << setw(10) << zc << endl;
}
};

```

Tea_Staf_Manage 类:

```

class Tea_Staf_Manage
{
    Tea_Sta Tea[N];
    int top;
public:
    Tea_Staf_Manage()
    {
        top = 0;
        cout << "请输入教师兼行政人员人数:";
        cin >> top;
        cout << "请输入教师兼行政人员信息:" << endl;
        cout << "编号 姓名 性别 年龄 系别 专业 职称 政治面貌 职称" << endl;
        for (int i = 0; i < top; i++)
            cin >> Tea[i].num >> Tea[i].name >> Tea[i].sex >> Tea[i].age >> Tea[i].depart >>
            Tea[i].major >> Tea[i].title >> Tea[i].zzmm >> Tea[i].zc;
    }
    //成员函数 void Add();
    void Show();
    void Search();
    void Edit();
    void Delete();
    void Total();
    void Save();
    void Read();
}

```

};//函数定义如上例，此处不一一赘述。

3.6 main()

//掉用的头文件#include<iostream>

#include<fstream>

#include<iomanip>

#include<string>

const int N = 100;

using namespace std;

//调用的函数

void ExpMenu();

void TeaMenu();

void StafMenu();

void Tea_StafMenu();

//实验员管理子菜单

void ExpMenu()

{

Exp_Manage t;

int choice;

cout << "*****实验员管理*****" << endl;

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

cout << "1.添加 2.显示 3.查找 4.修改 5.删除 6.统计 7.保存 8.读取 0.退出" << endl;

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

cout << "请输入您要操作的选项：" << endl;

cin >> choice;

while (choice != 0)

{

cout << "*****实验员管理*****" <<

endl;

cout << "*****" <<

endl;

cout << "1.添加 2.显示 3.查找 4.修改 5.删除 6.统计 7.保存 8.读取 0.退出" <<

endl;

cout << "*****" <<

endl;

cout << "请输入您要操作的选项：" << endl;

cin >> choice;

switch (choice)

{

case 1:t.Add(); break;

case 2:t.Show(); break;

case 3:t.Search(); break;



```

        case 4:t.Edit(); break;
        case 5:t.Delete(); break;
        case 6:t.Total(); break;
        case 7:t.Save(); break;
        case 8:t.Read(); break;
        case 0:cout << "退出成功" << endl; break;
        default:cout << "没有此项!" << endl; break;
    }
}
}

```

//教师管理子菜单

```
void TeaMenu()
```

```

{
    Tea_Manage t;
    int choice;
    cout << "*****教师管理*****" << endl;
    cout << "*****" << endl;
    cout << "1.添加 2.显示 3.查找 4.修改 5.删除 6.统计 7.保存 8.读取 0.退出" << endl;
    cout << "*****" << endl;
    cout << "请输入您要操作的选项:" << endl;
    cin >> choice;
    while (choice != 0)
    {
        switch (choice)
        {
            case 1:t.Add(); break;
            case 2:t.Show(); break;
            case 3:t.Search(); break;
            case 4:t.Edit(); break;
            case 5:t.Delete(); break;
            case 6:t.Total(); break;
            case 7:t.Save(); break;
            case 8:t.Read(); break;
            case 0:cout << "退出成功" << endl; break;
            default:cout << "没有此项!" << endl; break;
        }
    }
}

```

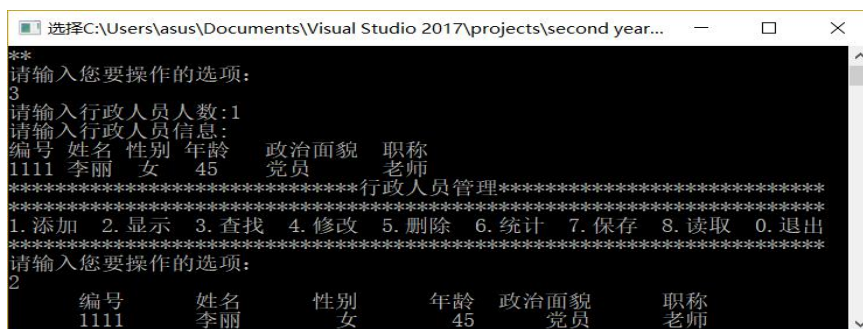
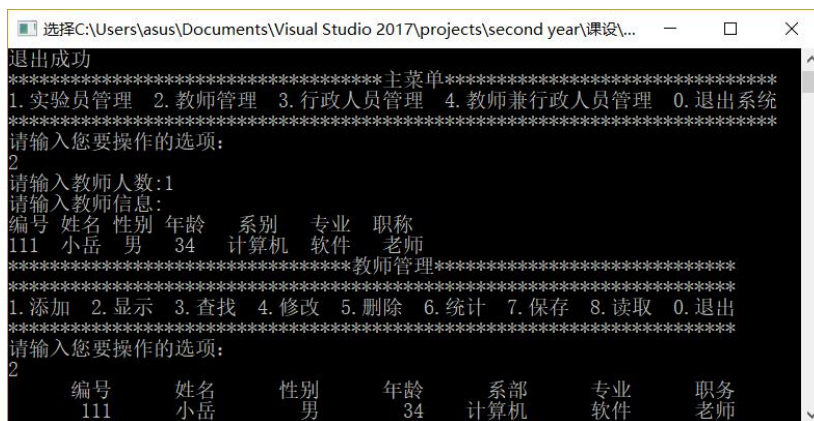
//行政人员管理子菜单

```
void StafMenu()
```

```

{
    Staf_Manage t;
    int choice;

```



```

cout << "*****行政人员管理*****" << endl;
cout << "*****" << endl;
cout << "1. 添加 2. 显示 3. 查找 4. 修改 5. 删除 6. 统计 7. 保存 8. 读取 0. 退出" << endl;
cout << "*****" << endl;
cout << "请输入您要操作的选项：" << endl;
cin >> choice;
while (choice != 0)
{
    switch (choice)
    {
        case 1:t.Add(); break;
        case 2:t.Show(); break;
        case 3:t.Search(); break;
        case 4:t.Edit(); break;
        case 5:t.Delete(); break;
        case 6:t.Total(); break;
        case 7:t.Save(); break;
        case 8:t.Read(); break;
        case 0:cout << "退出成功" << endl; break;
        default:cout << "没有此项!" << endl; break;
    }
}
}

```

//教师兼行政人员管理子菜单

```

void Tea_StafMenu()
{
    Tea_Staf_Manage t;
    int choice;
    cout << "*****教师兼行政人员管理*****" << endl;
    cout << "*****" << endl;
    cout << "1. 添加 2. 显示 3. 查找 4. 修改 5. 删除 6. 统计 7. 保存 8. 读取 0. 退出" << endl;
    cout << "*****" << endl;
    cout << "请输入您要操作的选项：" << endl;
    cin >> choice;
    while (choice != 0)
    {
        switch (choice)
        {
            case 1:t.Add(); break;
            case 2:t.Show(); break;
            case 3:t.Search(); break;
            case 4:t.Edit(); break;
            case 5:t.Delete(); break;
            case 6:t.Total(); break;

```




```

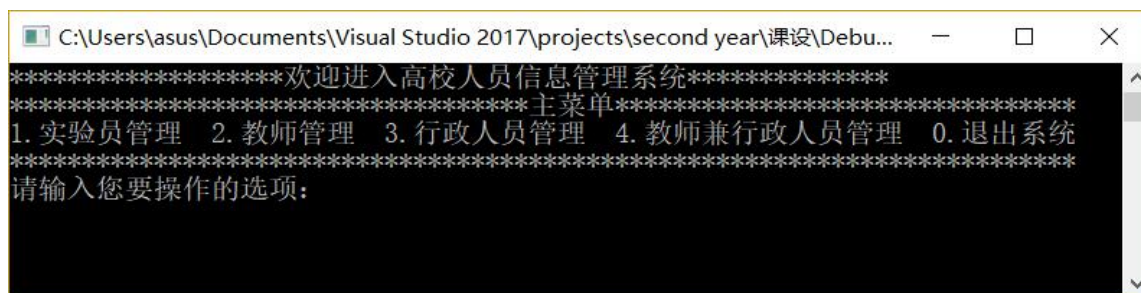
        case 7:t.Save(); break;
        case 8:t.Read(); break;
        case 0:cout << "退出成功" << endl; break;
        default:cout << "没有此项!" << endl; break;
    }
}
}

//高校人员信息管理主菜单
int main()
{
    cout << "*****欢迎进入高校人员信息管理系统*****" << endl;
    int choice;
//显示主菜单
    do {
        cout << "*****主菜单*****" <<
endl;
        cout << "1. 实验员管理  2. 教师管理  3. 行政人员管理  4. 教师兼行政人员管理  0. 退出系统" <<
endl;

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

        cout << "请输入您要操作的选项:" << endl;
        cin >> choice;
        switch (choice)
        {
            case 1: ExpMenu(); break;
            case 2: TeaMenu(); break;
            case 3: StafMenu(); break;
            case 4: Tea_StafMenu(); break;
            case 0:cout << "已退出系统!" << endl; break;
            default:cout << "没有此项!" << endl; break;
        }
    } while (choice != 0);
    cout << "欢迎下次使用!" << endl;
    return 0;
}

```



4.1 在调试过程中，出现了下面一些错误：

```

选择C:\Users\asus\Documents\Visual Studio 2017\projects\second year\课设D...
*****欢迎进入高校人员信息管理系统*****
*****主菜单*****
1. 实验员管理 2. 教师管理 3. 行政人员管理 4. 教师兼行政人员管理 0. 退出系统
*****
请输入您要操作的选项:
2
请输入教师人数:1
请输入教师信息:
编号 姓名 性别 年龄 系别 专业 职称
111 小明 男 21 计算机 软件 学生
*****教师管理*****
*****
1. 添加 2. 显示 3. 查找 4. 修改 5. 删除 6. 统计 7. 保存 8. 读取 0. 退出
*****
请输入您要操作的选项:
2
    编号      姓名      性别      年龄      系部      专业      职务
    111      小明      男        21       计算机  软件     学生
    编号      姓名      性别      年龄      系部      专业      职务
    111      小明      男        21       计算机  专       职学
    编号      姓名      性别      年龄      系部      业       生务
    111      小明      男        21       计算机  软       学
    编号      姓名      性别      年龄      系部      件       生务
    111      小明      男        21       计算机  专       学
    编号      姓名      性别      年龄      系部      业       生务
    111      小明      男        21       计算机  软       学
    编号      姓名      性别      年龄      系部      件       生务
    111      小明      男        21       计算机  专       学
    编号      姓名      性别      年龄      系部      业       生务
    111      小明      男        21       计算机  软       学
    编号      姓名      性别      年龄      系部      件       生务
    111      小明      男        21       计算机  专       学
    编号      姓名      性别      年龄      系部      业       生务
    111      小明      男        21       计算机  软       学
    编号      姓名      性别      年龄      系部      件       生务
    111      小明      男        21       计算机  专       学

```

诸如上错误很多，此处略举例一二，不做赘述。

1、对于死循环的调试，则是因为 switch 循环缺少选择语句造成的，只需要给 switch 循环加上控制按钮即可：附图：

```
while (choice != 0)
{
    cout << "*****实验员管理*****" << endl;
    cout << "*****" << endl;
    cout << "1.添加 2.显示 3.查找 4.修改 5.删除 6.统计 7.保存 8.读取 0.退出" << endl;
    cout << "*****" << endl;
    cout << "请输入您要操作的选项: " << endl;
    cin >> choice;
    switch (choice)
    {
```

25

5. 结论

程序总共设计了五个类，满足了课题对类的要求，在设计的过程中，建立了较为清晰的类层次，也利用了面向对象的方法以及 C++ 的编程思想来完成系统的设计，做到了用文件存储高校人员信息、从文件中读取高校人员信息、修改信息、查询信息、删除信息显示文件中存储的信息等要求。而且界面简洁，容易操作。

其次由于没有分开写类的定义及定义类的函数，在一个 .cpp 文件中编写，程序太长，不够简练，导致出现错误很难发现和修改，这是一个不足之处。

望课设不足之处还请老师谅解，希望得到老师指正！

6. 心得体会

此次我所选实验课题为——高校人员信息管理系统。对于该系统，我将其分为五个大类：一个基类及四个派生类。之后四个派生类又分别有一个新的类 `_Manage`，这两个类使用组合方式连接起来，用于演示主菜单中的子菜单。然后进行组合，即为我编写代码的思路。

通过这次课设的经历，我懂得了 c++ 面向对象的一些基本原则，也更加熟练地运用各个类以及函数的使用，受益颇多，以前不清晰、有误记的地方也更加清楚的知道该怎样处理。

哈哈，此次课设就到这里了，不足之处还望老师海涵啊，同时也希望得到老师的指正！

7. 参考书目与网址

- 1、菜鸟教程——（附网址：<https://www.runoob.com/>）；
- 2、c++面向对象程序设计——谭浩强 编著
- 3、c++类的封装