

考试报名系统

1.项目简介

考试报名工作给各高校报名工作带来了新的挑战，给教务管理部门增加了很大的工作量。本项目是对考试报名管理的简单模拟，用控制台选项的选择方式完成下列功能：输入考生信息；输出考生信息；查询考生信息；添加考生信息；修改考生信息；删除考生信息。

2.核心代码及功能介绍

- 定义类

- Stu(储存一位考生的信息单元)

```
class Stu{
public:
    Stu();
    ~Stu(){};
public:
    string number;           //考生学号
    string name;             //考生姓名
    string sex;              //性别
    int age;                 //年龄
    string catagery;         //职业
    Stu* next;               //指向下一个考生
    Stu* former;            //指向前一个考生
};
```

- System(整个考生信息系统，包含所有需要功能)

```
class System{
public:
    System();
    ~System();
    void Display();           //显示所有考生信息
    void Insert();           //插入考生信息
    void Find();             //查找考生是否存在
    void Modify();           //修改考生信息
    Stu* FindByOrder(int Pos); //按照考生顺序寻找考生
    Stu* FindByNum(string stuNumber); //按照考生学号寻找考生
    void Del();              //删除考生
    void getInfo(Stu* obj);   //获得考生信息
private:
    int stuNum;
    Stu* Head;
};
```

- 考生信息的建立

这里我们将读入考生信息抽象成getInfo函数从而达到减少代码，使项目操作清晰的目的，同时加入的简单的错误处理机制。

```
System::System() {
    cout<<"首先请建立考生信息系统"<<endl;
    while(1){
        cout<<"请输入考生人数: ";
        cin>>stuNum;
        getchar();
        if(stuNum<0){
            cout<<"student number must > 0"<<endl;
            continue;
        }
        else if(stuNum>0){
            cout<<"请一次输入考生的考号, 姓名, 性别, 年龄及报考类别! "<<endl;
            Stu *ptr1,*ptr2;
            ptr1=new Stu;
            Head=ptr1;
            getInfo(ptr1); //从stdin读入一位考生信息
            for(int i=0;i<stuNum-1;i++){ //读入剩下stuNum-1位考生信息
                ptr2=new Stu;
                getInfo(ptr2);
            }
        }
    }
}
```

```

        ptr1->next=ptr2;
        ptr2->former=ptr1;
        ptr1=ptr2;
    }
}
break;
}
Display();
}

```

- 考生信息的插入

对于这个部分需要区分三种情况，插入的考生的位置在首位，中间和末尾，从而采用三种不同的插入操作

```

void System::Insert() {
    int Pos;
    while(1){
        cout<<"请输入你要插入的位置: ";
        cin>>Pos;
        getchar();
        if(Pos<=0||Pos>stuNum+1){           //如果插入的位置不存在就报错重新输入
            cerr<<"can not insert at"<<Pos<<endl;
            continue;
        }
        else{
            cout<<"请依次输入要插入的考生的考号，性别，年龄及报考类别:";
            auto ptr1=new Stu;
            getInfo(ptr1);
            if(Pos==1){                       //插入的位置在首位
                if(stuNum){
                    Head->former=ptr1;
                    ptr1->next=Head;
                    Head=ptr1;
                }
                else{
                    Head=ptr1;
                }
            }
            else if(Pos==stuNum+1){          //插入的位置在末尾
                auto position=FindByOrder(stuNum);
                position->next=ptr1;
                ptr1->former=position;
            }
            else{                             //插入的位置在中间
                auto position=FindByOrder(Pos);
                auto temp=position->former;
                temp->next=ptr1;
                ptr1->former=temp;
                position->former=ptr1;
                ptr1->next=position;
            }
            stuNum++;
        }
        break;
    }
    Display();
    return;
}

```

- 考生信息的删除

这里要考虑如果为找到该位考生要做特殊处理，其他删除的时候也要分两种情况，一种为在首位，另一种为不在首位。

```

void System::Del() {
    string stuNumber;
    cout<<"请输入要删除的考生的考号: ";
    cin>>stuNumber;
    auto position=FindByNum(stuNumber);    //寻找该位置的考生
    if(position){                          //如果找到该位考生就执行删除操作
        cout<<"你要删除的考生信息是: ";
        cout<<position->number<<" " <<position->name<<" "
            <<position->sex<<" " <<position->age<<" "
            <<position->catagery<<endl;
        if(position==Head){                //如果删除的考生在首位
            auto posNext=position->next;
            posNext->former=NULL;
            Head=posNext;
            delete position;
        }
        else{                             //如果删除的考生不在首位
            auto posFormer=position->former;
            auto posNext=position->next;
            if(posNext){
                posFormer->next=posNext;
                posNext->former=posFormer;
            }
        }
    }
}

```

```

        }
        else{
            posFormer->next=NULL;
        }
    }
    stuNum--;
}
else{
    cerr<<"can not find this student"<<endl;
}
Display(); //显示删除后所有考生信息
return;
}

```

- 考生信息的修改

```

void System::Modify() {
    string stuNumber;
    cout<<"请输入你要修改的考生考号:";
    cin>>stuNumber;
    getchar();
    auto position=FindByNum(stuNumber);
    if(position){
        cout<<"请依次输入要插入的考生的考号, 性别, 年龄及报考类别:";
        getInfo(position);
    }
    else{
        cerr<<"can not find this student"<<endl;
    }
    return;
}

```

- 考生信息的查找

```

void System::Find(){
    string stuNumber;
    cout<<"请输入要查找的考生考号"<<endl;
    cin>>stuNumber;
    auto position=FindByNum(stuNumber);
    if(position){
        cout<<"考号    姓名    性别    年龄    报考类别"<<endl;
        cout<<position->number<<"    "<<position->name<<"    "
            <<position->sex<<"    "<<position->age<<"    "
            <<position->category<<endl;
    }
    else{
        cerr<<"can not find this student"<<endl;
    }
    return;
}

```

- 两个抽象出来的函数 findByNum & findByOrder

由于在整个程序中需要用到大量的按照考生考号和考生位置查找, 所以我特地抽象出两个查找函数, findByNum按照考生考号的查找, 返回指向该位考生的指针, 如果没有找到就返回NULL, findByOrder按照考生在整个储存链表里的位置查找, 返回指向指向该位置考生的指针, 如果没有找到返回NULL

```

Stu* System::FindByOrder(int Pos) {
    if(Pos>stuNum||Pos<=0){ //当找不到时返回NULL
        return NULL;
    }
    auto ptr=Head;
    Pos--;
    while(Pos--){ //找到第pos位考生并返回指向
        ptr=ptr->next; //该位考生的指针
    }
    return ptr;
}
Stu* System::FindByNum(string stuNumber) {
    auto ptr=Head;
    int count=stuNum;
    while(count--){
        //cout<<stuNumber<<endl;
        if(ptr->number==stuNumber){
            return ptr; //如果找到该位考生返回指向他的指针
        }
        else{
            ptr=ptr->next;
        }
    }
    return NULL; //如果没有找到则返回NULL
}

```

- 考生信息的统计
在这里默认展示所有的考生信息

```
void System::Display(){
    cout<<"考号    姓名    性别    年龄    报考类别"<<endl;
    auto ptr=Head;
    for(int i=0;i<stuNum;i++){
        cout<<ptr->number<<"    "
            <<ptr->sex<<"    "
            <<ptr->age<<"    "
            <<ptr->catagery<<endl;
        ptr=ptr->next;
    }
}
```

3.项目代码演示

下列操作均为顺序执行

- 项目进入界面

```
/Users/kirito/CLionProjects/untitled/cmake-build-debug/untitled
首先请建立考生信息系统
请输入考生人数: 3
请一次输入考生的考号, 姓名, 性别, 年龄及报考类别!
1 stu1 女 20 软件设计师
2 stu2 男 21 软件开发师
3 stu3 男 19 软件爱好者
考号    姓名    性别    年龄    报考类别
1      stu1    女      20      软件设计师
2      stu2    男      21      软件开发师
3      stu3    男      19      软件爱好者
```

- 插入操作显示(包括报错处理)

```
请选择您要进行的操作 (1为插入, 2为删除, 3为查找, 4为修改, 5为统计, 0为取消操作)
请选择您要的操作: 1
请输入你要插入的位置: 5
can not insert at5
请输入你要插入的位置: 4
请依次输入要插入的考生的考号, 性别, 年龄及报考类别:
4 stu4 女 24 软件开发师
考号    姓名    性别    年龄    报考类别
1      stu1    女      20      软件设计师
2      stu2    男      21      软件开发师
3      stu3    男      19      软件爱好者
4      stu4    女      24      软件开发师
```

- 删除操作显示(包括报错处理)

```
请选择您要进行的操作 (1为插入, 2为删除, 3为查找, 4为修改, 5为统计, 0为取消操作)
请选择您要的操作: 2
请输入要删除的考生的考号: -1
can not find this student
考号    姓名    性别    年龄    报考类别
1      stu1    女      20      软件设计师
2      stu2    男      21      软件开发师
3      stu3    男      19      软件爱好者
4      stu4    女      24      软件开发师
请选择您要进行的操作 (1为插入, 2为删除, 3为查找, 4为修改, 5为统计, 0为取消操作)
请选择您要的操作: 2
请输入要删除的考生的考号: 2
你要删除的考生信息是: 2      stu2    男      21      软件开发师
考号    姓名    性别    年龄    报考类别
1      stu1    女      20      软件设计师
3      stu3    男      19      软件爱好者
4      stu4    女      24      软件开发师
```

- 查找操作显示(包括错误处理)

```
请选择您要进行的操作 (1为插入, 2为删除, 3为查找, 4为修改, 5为统计, 0为取消操作)
```

请选择您要的操作: 3
请输入要查找的考生考号
32
can not find this student
请选择您要进行的操作 (1为插入, 2为删除, 3为查找, 4为修改, 5为统计, 0为取消操作)
请选择您要的操作: 3
请输入要查找的考生考号
4
考号 姓名 性别 年龄 报考类别
4 stu4 女 24 软件开发师

- 修改操作显示

请选择您要进行的操作 (1为插入, 2为删除, 3为查找, 4为修改, 5为统计, 0为取消操作)
请选择您要的操作: 4
请输入您要修改的考生考号: 1
请依次输入要插入的考生的考号, 性别, 年龄及报考类别: 5 stu5 男 23 学生
考号 姓名 性别 年龄 报考类别
5 stu5 男 23 学生
3 stu3 男 19 软件爱好者
4 stu4 女 24 软件开发师

- 统计操作显示

请选择您要进行的操作 (1为插入, 2为删除, 3为查找, 4为修改, 5为统计, 0为取消操作)
请选择您要的操作: 5
考号 姓名 性别 年龄 报考类别
5 stu5 男 23 学生
3 stu3 男 19 软件爱好者
4 stu4 女 24 软件开发师