

Assignment_1 考试报名系统

一、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。

二、构建设想：

本题目实际上是一个相对简单的链表题目，为了建立系统我应该定义适合的链表节点，然后要注意到链表的插入删除所需要注意的选项就可以实现系统的大部分功能了。最后注意一些异常情况的拍错，提高程序的健壮性，优化代码的可读性。

三、程序设计：

1. 程序的类与结构组织

因为要使用链表，所以首先要做的是对于链表的节点进行定义，如下代码块就是对链表节点结构的定义,其代表了整个一行学生的数据：

```
struct student
{
    int      id;          //考号
    int      age;         //年龄
    string    name;       //姓名
    string    gender;     //性别
    string    career;     //职业
    student*  next;       //指向下一个节点
};
```

接下来的是考试系统管理的类，这个类里包含了链表的初始化函数与对链表的插入，删除，查询，以及打印的操作，这是整个程序的核心类：

```

class StudentData
{
private:
    student* _head;        //表头
    student* _tail;        //表尾

public:
    StudentData();          //构造函数,初始化数据库
    void find();            //查找学生数据
    void run();             //系统运行的主函数
    void modify();          //修改学生数据
    void print() const;     //打印学生数据
    void insertData();      //插入考生数据
    void deleteData();      //删除考生数据
};

```

2.程序类中函数的实现

(1) 构造函数 StudentData() :

构造函数的功能主要是构建报名系统的数据库,当考试系统建立的时候,构造函数的功能是引导输入数据库中最基本的信息,然后对于表头跟表尾初始化,建立链表,如下是具体实现的代码:

```

//初始化数据库
StudentData::StudentData()
{
    _head = nullptr;
    cout << "首先建立考生信息系统" << endl << "请输入考生人数:";
    int studentNum;
    cin >> studentNum;
    cout << "请依次输入考生考号, 姓名, 性别, 年龄及报考类别!" << endl;
    _head = new student;
    cin >> _head->id >> _head->name >> _head->gender >> _head->age >> _head->career;
    this->_tail = _head;
    for(int i = 1; i < studentNum; i++)
    {
        _tail->next = new student;
        _tail = _tail->next;
        cin >> _tail->id >> _tail->name >> _tail->gender >> _tail->age >> _tail->career;
    }
    _tail->next = nullptr;
    print();
}

```

(2) 查找函数 find() :

查找函数的主要思路是把链表从头到尾遍历一遍，并且对各个节点进行比对如果找到匹配的信息就将其打印下来，如果找不到的话就提示返回上一层,如下是具体的代码实现:

```
//查找个人信息
void StudentData::find()
{
    cout << "请输入要查找的考号:" << endl;
    int id;
    cin >> id;
    auto p = _head;

    while(p != nullptr)
    {
        if(p->id == id)
        {
            cout << "考号\t姓名\t性别\t年龄\t报考类别" << endl;
            cout << p->id << "\t" << p->name << "\t"
                << p->gender << "\t" << p->age << "\t"
                << p->career << endl;
            break;
        }
        p = p-> next;
    }
    if( p == nullptr)
    {
        cout << "找不到学生信息，请重新输入" << endl;
    }
}
```

(3) 运行主函数 run() :

运行主函数主要的原理就是组织一个while的主循环，在用户不输入退出的情况下保持程序的一直运行，如果退出的话就提示学生已经成功退出，如下是具体的实现代码:

```

//选择的主程序
void StudentData::run()
{
    while(1)
    {
        int command;
        cout <<"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）" << endl;
        cin >> command;
        switch (command)
        {
            case 1:
                insertData();
                print();
                break;
            case 2:
                deleteData();
                print();
                break;
            case 3:
                find();
                break;
            case 4:
                modify();
                break;
            case 5:
                print();
                break;
            case 0:
                cout << "\n程序已经成功退出!\n " << endl;
                return ;
        }
    }
}

```

(4) 打印函数 print():

为了简化代码量，我添加了一个打印函数，专门用来打印数据库中的所有信息，比较简单就不过多阐述了，如下是具体的实现代码：

```

//打印学生信息
void StudentData::print() const
{
    cout << "考号\t姓名\t性别\t年龄\t报考类别" << endl;
    auto p = this->_head;
    while(p != nullptr)
    {
        cout << p->id << "\t" << p->name << "\t"
        << p->gender << "\t" << p->age << "\t"
        << p->career << endl;
        p = p->next;
    }
}

```

(5) 信息修改函数 modify():

信息修改函数的实现逻辑其实跟find()差不多，首先是遍历一遍链表如果找到学生的信息的话就提示修改，如果找不到的话就提示错误，如下是函数的具体实现代码：

```

//修改信息
void StudentData::modify()
{
    cout << "请输入要修改信息的考生的考号：" << endl;
    int id;
    cin >> id;
    auto p = _head;
    while(p != nullptr)
    {
        if(p->id == id)
        {
            cout << "请依次输入要修改的考生的考号，姓名，性别，年龄及报考的类别" <<
endl;

            cin >> p->id >> p->name >> p->gender
            >> p->age >> p->career;
            cout << "修改成功！" << endl;
            cout << "考号\t姓名\t性别\t年龄\t报考类别" << endl;
            cout << p->id << "\t" << p->name << "\t"
            << p->gender << "\t" << p->age << "\t"
            << p->career << endl;
            break;
        }
        p = p-> next;
    }
    if( p == nullptr)
    {
        cout << "找不到学生信息，请重新输入，或退出。" << endl;
    }
}

```

(6) 信息插入函数 insertData():

信息的插入功能其实是整个程序的核心功能，如果要实现插入功能我们要考虑很多的地方，比如向有表头的地方进行插入，如果把这些功能都注意到了的话，整个功能也就

```
//插入数据
void StudentData::insertData()
{
    cout << "请输入您想要插入的考生的位置:" << endl;
    int position;
    cin >> position;
    cout << "请依次输入要插入的考生的考号，姓名，性别，年龄及报考的类别" << endl;
    auto newNode = new student;
    cin >> newNode->id >> newNode->name >> newNode->gender
    >> newNode->age >> newNode->career;
    auto p = _head;
    for(int i = 0; i < position - 1; i++)
    {
        if(p->next != nullptr)
            p = p->next;
    }

    if( p -> next == nullptr)
    {
        p->next = newNode;
        this->_tail = newNode;
    }
    else
    {
        newNode->next = p->next;
        p->next = newNode;
    }
}
```

(7) 信息删除函数 deleteData():

信息修改函数的实现逻辑其实跟find()差不多，首先是遍历一遍链表如果找到学生的信息的话就提示修改，如果找不到的话就提示错误，如下是函数的具体实现代码：

```

//删除数据
void StudentData::deleteData()
{
    cout << "请输入要删除的考生的考号:" << endl;
    int id;
    cin >> id;
    if(this->_head->id == id)
    {
        auto temp = _head;
        _head = _head->next;
        delete temp;
        return;
    }
    auto p = _head;
    while(p->next != nullptr)
    {
        if(p->next->id == id)
        {
            auto q = p->next;
            p->next = p->next->next;
            delete q;
            return;
        }
        p = p->next;
    }
    if(p->next == nullptr)
    {
        cout << "找不到学生的信息" << endl;
    }
}
}

```

四、程序功能性测试：

1. 考试系统的初始化功能测试：

输入考生人数与具体信息，对链表进行初始化：

```
Build All
/Users/nickel/CLionProjects/project1/cmake-build-debug/project1
首先建立考生信息系统
请输入考生人数:2
请依次输入考生考号, 姓名, 性别, 年龄及报考类别!
1 宋天佑 男 19 软件测试工程师
2 张文喆 男 18 前端工程师
考号      姓名      性别      年龄      报考类别
1         宋天佑    男        19        软件测试工程师
2         张文喆    男        18        前端工程师
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)
```

2.考试系统的插入功能测试:

选择操作1, 然后输入考生位置, 最后输入考生信息, 最后插入成功!

```
2         张文喆    男        18        前端工程师
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)
1
请输入您想要插入的考生的位置:
0
请依次输入要插入的考生的考号, 姓名, 性别, 年龄及报考的类别
3 李向真 男 19 软件测试工程师
考号      姓名      性别      年龄      报考类别
1         宋天佑    男        19        软件测试工程师
3         李向真    男        19        软件测试工程师
2         张文喆    男        18        前端工程师
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)
```

3.考试系统的删除功能测试:

选择操作2, 然后输入考生考号, 由打印结果判定删除成功

```
2         张文喆    男        18        前端工程师
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)
2
请输入要删除的考生的考号:
3
考号      姓名      性别      年龄      报考类别
1         宋天佑    男        19        软件测试工程师
2         张文喆    男        18        前端工程师
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)
```

4.考试系统的查找功能测试:

选择操作3, 然后输入考生考号, 如果找到的话就打印信息


```
1      宋天佑    男      19      软件测试工程师
2      张文喆    男      18      前端工程师
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)
3
请输入要查找的考号:
2
考号      姓名      性别      年龄      报考类别
2      张文喆    男      18      前端工程师
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)
```

5.考试系统的修改功能测试:

选择操作4, 然后输入考生考号, 如果找到的话就重新输入学生的信息, 如图成功修改宋天佑同学的考试类别为机器学习工程师.

```
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)
4
请输入要修改信息的考生的考号:
1
请依次输入要修改的考生的考号, 姓名, 性别, 年龄及报考的类别
1 宋天佑 男 18 机器学习工程师
修改成功!
考号 姓名 性别 年龄 报考类别
1 宋天佑 男 18 机器学习工程师
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)
```

5.考试系统的统计功能测试:

选择操作5, 将显示考试信息系统中所有的数据:

```
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)
5
考号      姓名      性别      年龄      报考类别
1      宋天佑    男      18      机器学习工程师
2      张文喆    男      18      前端工程师
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)
```

五、程序健壮性测试:

1.考试系统的初始化功能健壮性测试:

程序将会保证初始化时的数字大于等于1, 否则就提示错误:

```
Build All
/Users/nickel/CLionProjects/project1/cmake-build-debug/project1
首先建立考生信息系统
请输入考生人数: 0
输入错误, 请重新输入 (输入大于等于1的数字): 1
请依次输入考生考号, 姓名, 性别, 年龄及报考类别!
1 stu1 男 15 软件测试工程师
考号      姓名      性别      年龄      报考类别
1      stu1      男      15      软件测试工程师
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)
```

2.考试系统插入功能健壮性测试：

程序将会保证插入时候的数字大于等于0，否则就提示错误：

```
1      2      3      4      5
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）
1
请输入您想要插入的考生的位置：
-1
输入错误，请重新改输入！
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）
```

3.考试系统的删除功能健壮性测试：

程序如果找不到被删除的同学就提示错误：

```
1      2      3      4      5
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）
2
请输入要删除的考生的考号：
12
找不到学生的信息
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）
|
```

4.考试系统的查找功能健壮性测试：

程序如果找不到被删除的同学就提示错误：

```
找不到学生的信息
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）
3
请输入要查找的考号：
11
找不到学生信息，请重新输入
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）
|
```