

项目说明文档

数据结构课程设计

——考试报名系统

作者姓名： 陈垚昕

学 号： 

指导教师： 张颖

学院、专业： 软件学院 软件工程

同济大学

Tongji University

目录

可通过按住 **Ctrl** 并单击访问说明文档各个模块：

1. 分析

- 1.1 背景分析
- 1.2 功能分析

2. 设计

- 2.1 主要数据结构设计
 - 1. 代表节点的节点类 **StudentNode**
 - 2. 链表主体类 **StudentList**

3. 核心功能实现

- 3.1 插入功能的实现
 - 功能简介,流程图与核心代码
- 3.2 删除功能的实现
 - 功能简介,流程图与核心代码
- 3.3 查找功能的实现
 - 功能简介,流程图与核心代码
- 3.4 修改功能的实现
 - 功能简介,流程图与核心代码
- 3.5 统计功能的实现
 - 功能简介、流程图与核心代码
- 3.6 考试报名总体管理系统的实现
 - 功能简介、流程图与核心代码
- 3.7 学生信息修正函数
 - 功能简介、流程图与核心代码

4. 测试

- 4.1 常规结果测试
 - Task 1 插入功能测试
 - Task 2 删除功能测试
 - Task 3 查找功能测试
 - Task 4 修改功能测试
 - Task 5 统计功能测试
- 4.2 边界测试
 - Task 1 删除头结点
 - Task 2 删除后链表为空
- 4.3 错误结果测试
 - Task 1 初始化输入不合法学生人数
 - Task 2 操作码错误
 - Task 3 插入位置不存在

Task 4 删除考号不存在

Task 5 查找考号不存在

Task 6 修改考号不存在

Task 7 学生信息输入有误

1.初始化链表时输入错误

2.插入操作时输入错误

3.修改时输入错误

1. 分析

1.1 背景分析

考试报名系统是一个学校不可缺少的部分，它对于学校的管理者和学生来说都至关重要，所以一个好的考试报名系统应该能够为用户提供充足的信息和功能。考试报名系统对于学校加强考试管理有极其重要的作用。随着学生数量和考试数量的日益庞大，如何管理如此庞大的数据显得极为复杂，传统的手工管理工作量大且容易出错。

随着计算机科学技术的不断成熟，使用计算机对考试报名系统进行管理，具有手工管理所无法比拟的优势。这些优点能够极大地提高学校和学生的效率，也是学校走向信息化、科学化、国际化的重要条件。因此，开发一套考试报名系统具有十分重要的意义。

1.2 功能分析

作为一个最简易的考试报名系统，首先应该有的功能就是输入同学们的考试报名情况并且可以予以显示。其次，考试报名系统还应该具有插入、删除、修改功能，以保证同学可以随时更改自己的考试报名情况。最后，考试报名系统软件还应该确保软件可以正常关闭。

综上所述，一个考试报名系统至少应该具有输入、输出、插入、删除、修改、退出的功能。

2.设计

2.1 主要数据结构设计

若如题目所述，该考试报名系统的信息操作一般是输入、输出、插入、删除、修改、退出的功能，考虑到该报名系统需要频繁的插入，删除，使用链表类可以较快速地解决这类问题。

本程序使用带有附加头结点的单向链表实现考试报名系统，程序主要的类设计如下

1. 代表节点的节点类 StudentNode

基本描述：**class StudentNode** 为 **StudentList** 的结点类，以 **StudentList** 为友元，实现良好的封装。在包含学生信息，以及链表上的下一个存储单元的同时，有构造析构函数，结点信息打印与输入函数的实现，并重载了运算符==与!=，方便后续程序的编写与可读性

```
class StudentNode {
    friend class StudentList;
public:
    //构造，析构函数
    StudentNode() :_next(nullptr) {}
    ~StudentNode() = default;
    inline void inputStudentInfo();

    //学生信息的操作：输入(修改),输出
    inline void printStudentInfo()const;
    bool operator==(const StudentNode& rhs); //重载运算符:==,!=
    bool operator!=(const StudentNode& rhs);
private:
    int _candidateNumber;    //考号
    string _name;            //名字
    string _gender;          //性别
    int _age;                //年龄
    string _examType;        //考试类别
    //链表单元
    StudentNode* _next;
};
```

2. 链表主体类 StudentList

基本描述：**class StudentList** 类为考试报名系统的主题类，私有成员包含一个 **StudentNode** 附加头结点，以及用于记录内含学生信息个数的 **int** 类型成员外，实现了类运行程序，输入输出方法，构造析构函数，操作输入指令等，实现良好的封装

```
class StudentList {
public:
    /*类运行程序*/
    void inputList();                // 初始构造链表
    void printList()const;          // 打印链表内容
    void runClassProgram();          // 执行输入指令的操作

    // 异常处理，对 StudentNode 类指针所指的 student
    void correctStudentInfo(StudentNode* student);

    // 构造析构函数
    StudentList() : _header(new StudentNode()), _length(0) {}
    ~StudentList() { clear(); }

    // 链表操作输入指令
    void inputAndInsert();           // 输入学生信息，插入链表
    void inputAndDelete();           // 输入学生信息，从链表中删除
    void inputFindAndPrintStudent()const; // 输入学生信息查找打印学生信息
    void inputAndModify();           // 输入学生信息，查找并修改

    // 链表单元层面上的操作

    // 根据考号查找，返回对应考号学生节点的前一个节点
    StudentNode* findFront(int candNumber)const;
    // 链表层面操作
    void clear();                    // 清空，释放内存，在程序结束时使用

private:
    StudentNode* _header;            // 附加的头结点
    int _length;                     // 链表存储的 StudentNode 个数
};
```

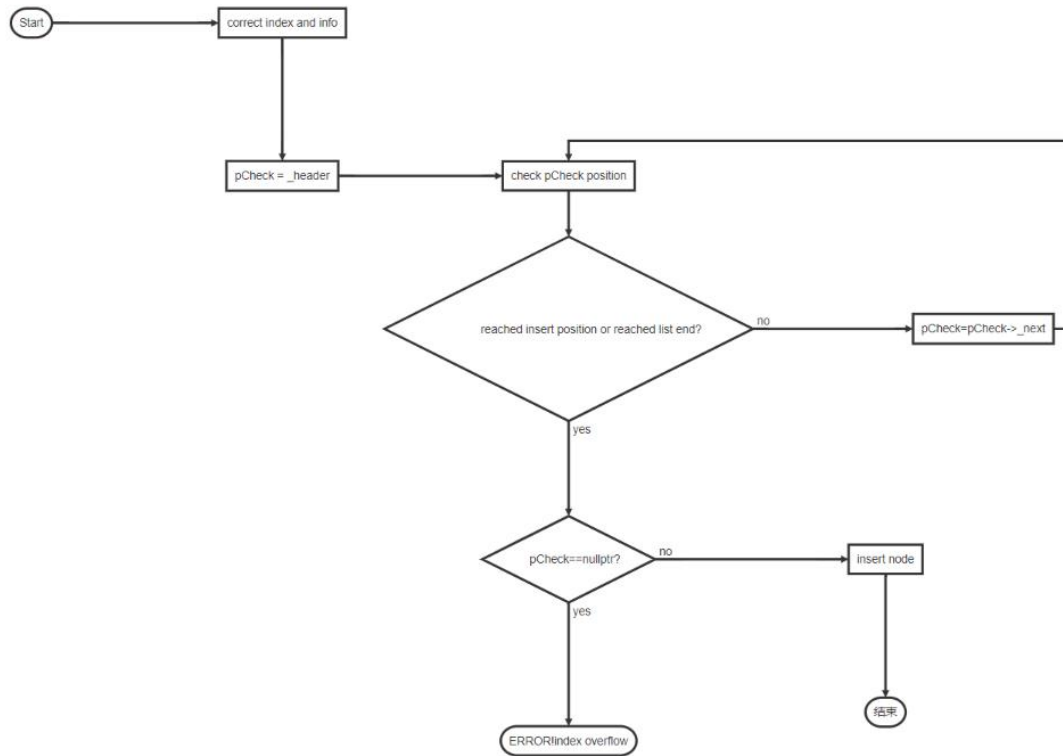
3.核心功能实现

3.1 插入功能的实现

功能简介,流程图与核心代码

描述：输入插入的位置与学生信息，随后插入链表并打印表显示当前状态

异常处理:1.对输入有误的考生信息（性别格式有误，考号重复）,提示错误并要求重新输入直到输入格式正确为止;2.删除的考生若不存在，提示考生不存在



```

for (int i = 1; i < insertIndex; i++)
{
    if (!pCheck) break;
    pCheck = pCheck->_next;
}

if (!pCheck)
{
    std::cerr << "错误：索引溢出" << std::endl;
}
else
{
    newStudent->_next = pCheck->_next;
}
    
```

```

        pCheck->_next = newStudent;
        ++_length;
    }

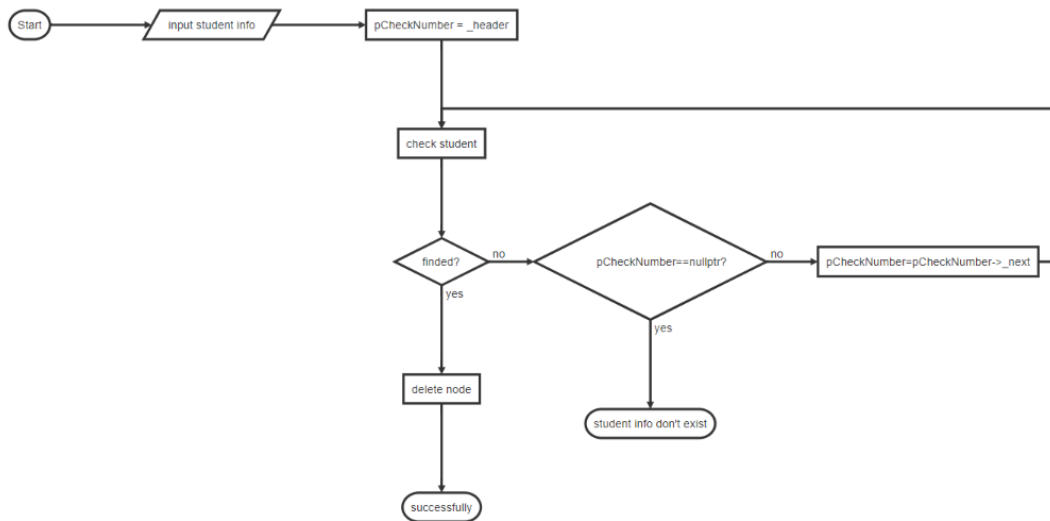
```

3.2 删除功能的实现

功能简介,流程图与核心代码

描述：输入需要删除的学生考号，从链表中删除

异常处理:1.对输入有误的考生信息（性别格式有误，考号重复）,提示错误并要求重新输入直到输入格式正确为止;2.插入的考生位置如为不符合要求的索引,提示插入位置有误并要求重复输入直至输入合法的索引



```

StudentNode* pFindFront = findFront(targetCandidateNumber);
if (!pFindFront->_next)
{
    std::cout << "抱歉，您要删除的学生不存在" << std::endl;
}
else
{
    std::cout << "你删除的考生信息是：";

    StudentNode* pDelete = pFindFront->_next;
    pDelete->printStudentInfo();
}

```



```

        pFindFront->_next = pDelete->_next;
        delete pDelete;

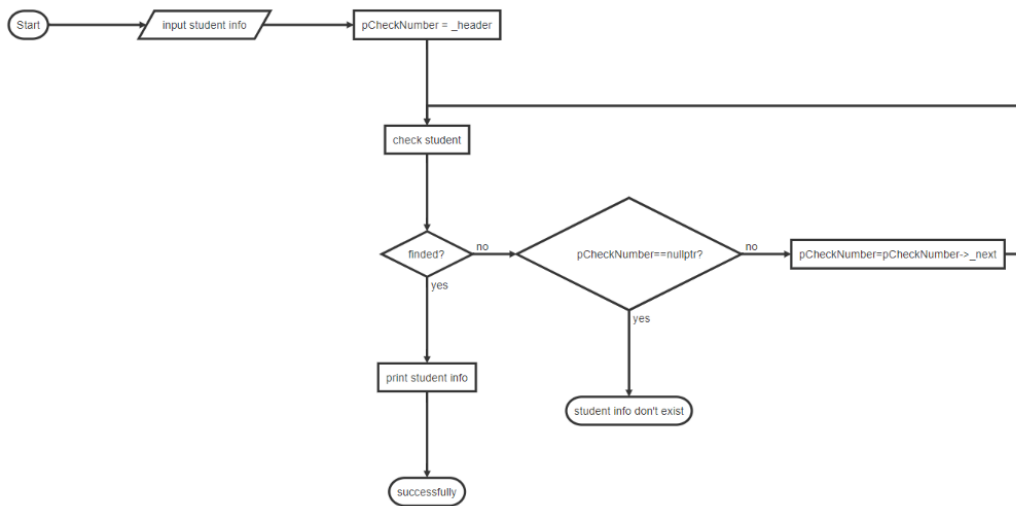
        --_length;
    }

```

3.3 查找功能的实现

功能简介,流程图与核心代码

描述：遍历链表查找对应考生信息，若不存在，提示考生不存在；若存在，打印考生信息



```

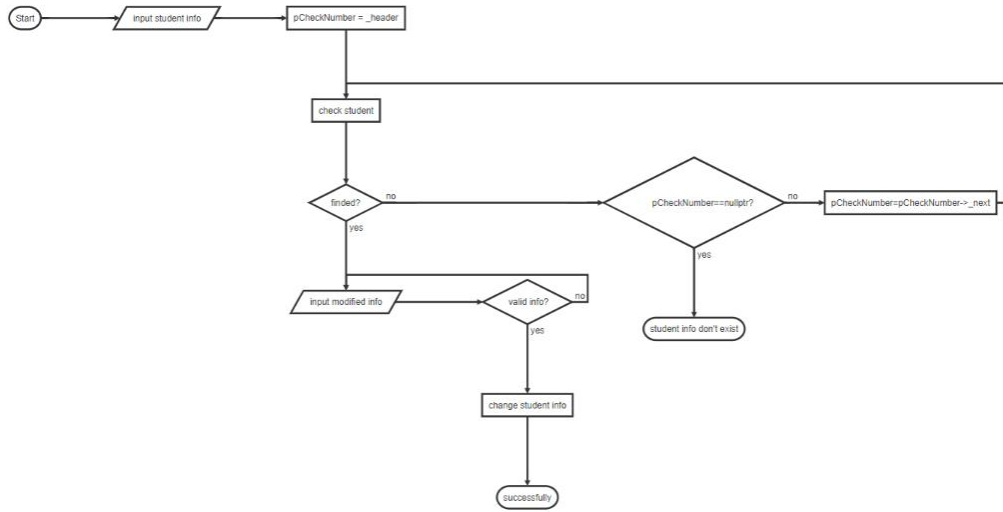
StudentNode* pFindFront = findFront(targetCandidateNumber);
if (!pFindFront->_next)
{
    std::cout << "抱歉，您要查找的学生不存在" << std::endl;
}
else
{
    std::cout << "你查找的考生信息是：";
    pFindFront->_next->printStudentInfo();
}

```

3.4 修改功能的实现

功能简介，流程图与核心代码

查找功能部分同 3.3 的实现，对查找到的学生，重新输入考生信息并检查输入的信息是否正确



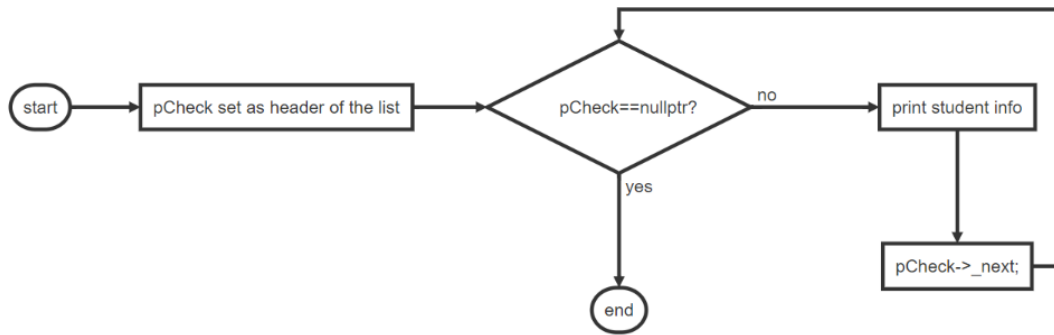
```

StudentNode* pFindFront = findFront(targetCandidateNumber);
if (!pFindFront->_next)
{
    std::cout << "抱歉，您要修改的学生不存在" << std::endl;
}
else
{
    std::cout << "请依次输入考生考号，姓名，性别，年龄及报考类别！"
<< std::endl;
    pFindFront->_next->inputStudentInfo();
    correctStudentInfo(pFindFront->_next);
}
    
```

3.5 统计功能的实现

功能简介、流程图与核心代码

描述：打印表列，从表头到表尾的顺序输出链表内容



```

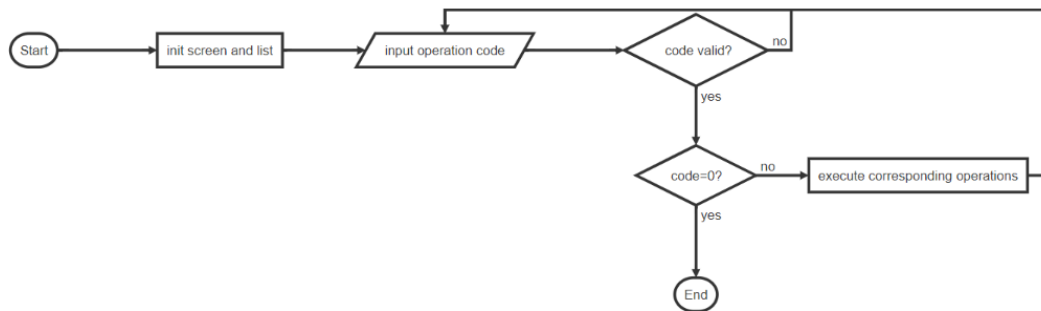
void StudentList::printList()const
{
    StudentNode* pCheck = _header->_next;
    std::cout << "考号" << "      " << "姓名" << "      " << "性别" << "
    "
    << "年龄" << "      " << "报考类别" << std::endl;
    while (pCheck)
    {
        pCheck->printStudentInfo();
        pCheck = pCheck->_next;
    }
}
    
```

3.6 考试报名总体管理系统的实现

功能简介、流程图与核心代码

描述：类的运行程序:按照 io 指示输入各种操作的对应指令:1 为插入,2 为删除, 3 为查找, 4 为修改, 5 为统计, 0 为取消操作退出程序

异常处理：对不在 0-5 内的操作指令,进行提示并要求重新输入



case 1:

//进行插入操作 打印表

```
inputAndInsert();
printList();
break;
```

case 2:

//进行删除操作 打印表

```
inputAndDelete();
printList();
break;
```

case 3:

//进行查找考号操作,打印查找的学生信息

```
inputFindAndPrintStudent();
break;
```

case 4:

//进行修改操作, 打印表

```
inputAndModify();
printList();
break;
```

case 5:

//统计, 打印表

```
printList();
break;
```

case 0:

```
std::cout << "结束操作, 已退出考试报名系统程序, 感谢您的使用" <
```

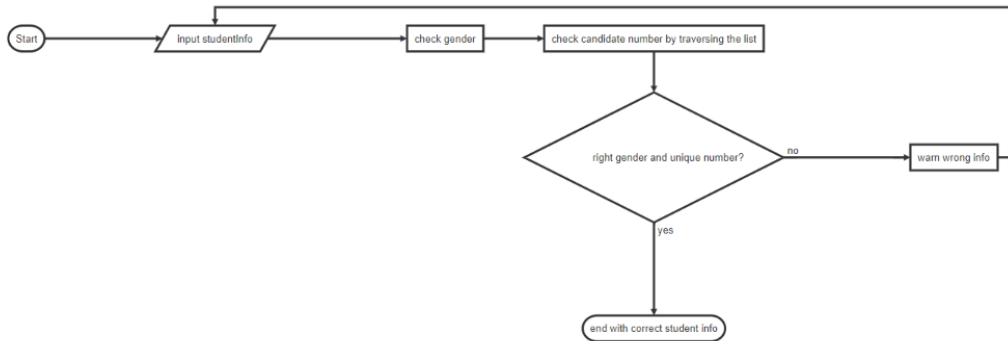
```
< std::endl;
    clear();
    return;
default:
    std::cout << "无效的操作序号" << std::endl;
    break;
```

3.7 学生信息修正函数

功能简介、流程图与核心代码

描述：传入所要检查的学生节点参数，检查所输入的学生考号是否重复，性别输入是否正确，若不正确，提示有误的信息并重新输入，直到学生信息无误后，退出函数

异常处理:由于考试类别未给出，且考虑重名的情况，仅对性别，考号进行纠错，排除考号重复与性别格式有误（不是"男"或"女"）的情况



//检查性别输入

```
if (student->_gender != "男" && student->_gender != "女")
{
    rightGender = false;
}
```

//遍历链表，检查考号是否重复

```
StudentNode* pCheck = _header->_next;
while (pCheck)
{
```

```
        if (pCheck->_candidateNumber == student->_candidateNumber && *pCheck != *student)
        {
            rightCandidateNumber = false;
            break;
        }
        else
        {
            pCheck = pCheck->_next;
        }
    }
```

4.测试

4.1 常规结果测试

Task 1 插入功能测试

测试样例：

插入信息：

0 Barret 男 25 Sub0

原有表列：

1 Cloud 男 20 Sub1

2 Tifa 女 19 Sub2

预期结果：

0 Barret 男 25 Sub0

1 Cloud 男 20 Sub1

2 Tifa 女 19 Sub2

实际结果：

```
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 Cloud 男 20 Sub1
2 Tifa 女 19 Sub2
考号    姓名    性别    年龄    报考类别
1      Cloud    男      20      Sub1
2      Tifa     女      19      Sub2
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）

请选择您要进行的操作:1
请输入你要插入的考生位置:1
请依次输入考生考号，姓名，性别，年龄及报考类别！
0 Barret 男 25 Sub0
考号    姓名    性别    年龄    报考类别
0      Barret    男      25      Sub0
1      Cloud    男      20      Sub1
2      Tifa     女      19      Sub2
```

Task 2 删除功能测试

测试样例：

删除考号为 1 的考生

原有表列：

1 Cloud 男 20 Sub1
2 Tifa 女 19 Sub2

预期结果：

2 Tifa 女 19 Sub2

实际结果：

```
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 Cloud 男 20 Sub1
2 Tifa 女 19 Sub2
考号    姓名    性别    年龄    报考类别
1        Cloud  男      20      Sub1
2        Tifa   女      19      Sub2
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）

请选择您要进行的操作:2
请输入要删除的考生的考号：
1
你删除的考生信息是：1   Cloud   男      20      Sub1
考号    姓名    性别    年龄    报考类别
2        Tifa   女      19      Sub2
```

Task 3 查找功能测试

测试样例：

查找考号为 3 的考生

原有表列：

1 Tifa 女 20 Sub1
2 Cloud 男 21 Sub2
3 Aerith 女 22 Sub3

预期结果：

3 Aerith 女 22 Sub3

实际结果：


```

Aerith 女 22 Sub3
考号 姓名 性别 年龄 报考类别
1 Tifa 女 20 Sub1
2 Cloud 男 21 Sub2
3 Aerith 女 22 Sub3
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）

请选择您要进行的操作:3
请输入要查找的考生的考号:
3
你查找的考生信息是: 3 Aerith 女 22 Sub3
    
```

Task 4 修改功能测试

测试样例:

将考号 1 学生修改为:

4 Zack 男 21 Sub2

原有表列:

```

1 Tifa 女 19 Sub1
2 Cloud 男 20 Sub2
3 Aerith 女 21 Sub3
    
```

预期结果:

```

4 Zack 男 21 Sub2
2 Cloud 男 20 Sub2
3 Aerith 女 21 Sub3
    
```

实际结果:

```

考号 姓名 性别 年龄 报考类别
1 Tifa 女 19 Sub1
2 Cloud 男 20 Sub2
3 Aerith 女 21 Sub3
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）

请选择您要进行的操作:4
请输入要修改的考生的考号:
1
请依次输入考生考号，姓名，性别，年龄及报考类别！
4 Zack 男 21 Sub2
考号 姓名 性别 年龄 报考类别
4 Zack 男 21 Sub2
2 Cloud 男 20 Sub2
3 Aerith 女 21 Sub3
    
```

Task 5 统计功能测试

测试样例:

原有表列：

1	Tifa	女	19	Sub1
2	Cloud	男	20	Sub2
3	Aerith	女	21	Sub3

预期结果：

1	Tifa	女	19	Sub1
2	Cloud	男	20	Sub2
3	Aerith	女	21	Sub3

实际结果：

```

考号  姓名  性别  年龄  报考类别
1     Tifa  女    19    Sub1
2     Cloud 男    20    Sub2
3     Aerith 女    21    Sub3
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）
请选择您要进行的操作:5
考号  姓名  性别  年龄  报考类别
1     Tifa  女    19    Sub1
2     Cloud 男    20    Sub2
3     Aerith 女    21    Sub3
    
```

4.2 边界测试

Task 1 删除头结点

测试样例：

删除头结点(此例头结点序列为 1)

原有表列：

1	Cloud	男	20	Sub1
2	Tifa	女	19	Sub2
3	Sephi	男	28	Sub3

预期结果：

程序正常运行，不崩溃。

现有表列：

2	Tifa	女	19	Sub2
3	Sephi	男	28	Sub3

实际结果：

```

考号    姓名    性别    年龄    报考类别
1      Cloud    男      20      Sub1
2      Tifa     女      19      Sub2
3      Sephi    男      28      Sub3
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）

请选择您要进行的操作:2
请输入要删除的考生的考号:
1
你删除的考生信息是: 1   Cloud    男      20      Sub1
考号    姓名    性别    年龄    报考类别
2      Tifa     女      19      Sub2
3      Sephi    男      28      Sub3
    
```

Task 2 删除后链表为空

测试样例:

删除前链表只有一个结点，删除后链表为空

原有表列:

3 Sephi 男 28 Sub3

预期结果:

程序正常运行，不崩溃。

现有表列:

实际结果:

```

考号    姓名    性别    年龄    报考类别
3      Sephi    男      28      Sub3
请选择您要进行的操作:2
请输入要删除的考生的考号:
3
你删除的考生信息是: 3   Sephi    男      28      Sub3
考号    姓名    性别    年龄    报考类别
    
```

4.3 错误结果测试

Task 1 初始化输入不合法学生人数

测试样例:

初始无输入数据(考生数为 0)

预期结果：

给出错误提示，程序运行正常不崩溃。

实际结果：

```
首先建立考生信息系统
请输入考生人数
0
考生人数应为正整数，请重新输入：
```

Task 2 操作码错误

测试样例：

输入操作码错误(例如输入 9)

预期结果：

程序给出提示信息，程序正常运行不崩溃。

实际结果：

```
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）
请选择您要进行的操作:9
无效的操作序号（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）
请选择您要进行的操作: _
```

Task 3 插入位置不存在

测试样例：

输入不合法插入位置（本例中为 5）

原有表列：

4	Zack	男	21	Sub2
2	Cloud	男	20	Sub2
3	Aerith	女	21	Sub3

预期结果：

程序给出提示信息（合法的区间范围，本例中为 1-4），要求重新输入，程序正常运行不崩溃。

实际结果:

```

考号      姓名      性别      年龄      报考类别
4         Zack      男        21        Sub2
2         Cloud      男        20        Sub2
3         Aerith     女        21        Sub3

请选择您要进行的操作:1
请输入您要插入的考生位置:5
插入的位置有误,请重新输入一个1到4之间的整数:
    
```

Task 4 删除考号不存在

测试样例:

输入不存在的考号 (本例中为 4)

原有表列:

1	Cloud	男	20	Sub1
2	Aerith	女	21	Sub0
3	Tifa	女	19	Sub2

预期结果:

程序给出错误信息,程序正常运行不崩溃。

实际结果:

```

0 Aerith 女 21 Sub0
考号      姓名      性别      年龄      报考类别
1         Cloud      男        20        Sub1
0         Aerith     女        21        Sub0
2         Tifa       女        19        Sub2

请选择您要进行的操作:2
请输入要删除的考生的考号:
4
抱歉,您要删除的学生不存在
    
```

Task 5 查找考号不存在

测试样例:

输入不存在的考号（本例中为 3）

原有表列：

1	Cloud	男	20	Sub1
0	Aerith	女	21	Sub0
2	Tifa	女	19	Sub2

预期结果：

程序给出错误信息，程序正常运行不崩溃。

实际结果：

```

考号    姓名    性别    年龄    报考类别
1       Cloud   男      20      Sub1
0       Aerith   女      21      Sub0
2       Tifa     女      19      Sub2

请选择您要进行的操作:3
请输入要查找的考生的考号:
3
抱歉，您要查找的学生不存在
    
```

Task 6 修改考号不存在

测试样例：

输入不存在的考号（本例中为 3）

原有表列：

1	Cloud	男	20	Sub1
0	Aerith	女	21	Sub0
2	Tifa	女	19	Sub2

预期结果：

程序给出错误信息，程序正常运行不崩溃。

实际结果：

考号	姓名	性别	年龄	报考类别
1	Cloud	男	20	Sub1
0	Aerith	女	21	Sub0
2	Tifa	女	19	Sub2

```

请选择您要进行的操作:4
请输入要修改的考生的考号:
3
抱歉，您要修改的学生不存在
    
```

Task 7 学生信息输入有误

测试样例：

输入学生信息时分别输入重复的学号，错误的性别格式

预期结果：

程序给出错误信息，程序正常运行不崩溃。

实际结果：

1.初始化链表时输入错误

```

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 Sephiroth 男 20 Sub1
1 Cloud 男 19 Sub2
考生信息有误:考号已被使用
请重新输入考生信息，依次依次输入考生考号，姓名，性别，年龄及报考类别：
2 Cloud None 19 Sub2
考生信息有误:性别格式有误（应输入男/女）
请重新输入考生信息，依次依次输入考生考号，姓名，性别，年龄及报考类别：
1 Cloud None 19 Sub2
考生信息有误:考号已被使用 性别格式有误（应输入男/女）
请重新输入考生信息，依次依次输入考生考号，姓名，性别，年龄及报考类别：
2 Cloud 男 19 Sub2
考号    姓名    性别    年龄    报考类别
1      Sephiroth    男      20      Sub1
2      Cloud    男      19      Sub2
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.插入操作时输入错误

```

考号      姓名      性别      年龄      报考类别
1         Cloud    男       20       Sub1
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）

请选择您要进行的操作:1
请输入你要插入的考生位置:1
请依次输入考生考号，姓名，性别，年龄及报考类别！
1 Tifa 女 19 Sub2
考生信息有误:考号已被使用
请重新输入考生信息，依次依次输入考生考号，姓名，性别，年龄及报考类别：
2 Tifa None 19 Sub2
考生信息有误:性别格式有误（应输入男/女）
请重新输入考生信息，依次依次输入考生考号，姓名，性别，年龄及报考类别：
    
```

3.修改时输入错误

```

考号      姓名      性别      年龄      报考类别
1         Cloud    男       20       Sub1
0         Aerith   女       21       Sub0
2         Tifa     女       19       Sub2
请选择您要进行的操作:4
请输入要修改的考生的考号：
1
请依次输入考生考号，姓名，性别，年龄及报考类别！
2 Cloud 男 20 Sub1
考生信息有误:考号已被使用
请重新输入考生信息，依次依次输入考生考号，姓名，性别，年龄及报考类别：
    
```