

# 数据结构项目文档

题目：考试报名系统

指导教师：张颖

姓名：王亮

学号：1653340

## 数据结构项目文档

### 一、题目分析

1. 项目简介
2. 功能需求
3. 设计思路

### 二、设计

1. 数据结构设计
2. 类的设计
  - 2.1 StudentData
  - 2.2 LinkNode
  - 2.3 LinkList
3. 主函数设计

### 三、实现

1. 建立系统
  - 1.1 建立功能流程图
  - 1.2 建立功能核心代码
  - 1.3 建立功能截屏示例
2. 插入功能
  - 2.1 插入功能流程图
  - 2.2 插入功能核心代码
  - 2.3 插入功能截屏示例
3. 查找功能
  - 3.1 查找功能流程图
  - 3.2 查找功能核心代码
  - 3.3 查找功能截屏示例
4. 修改功能
  - 4.1 修改功能流程图
  - 4.2 修改功能核心代码
  - 4.3 修改功能截屏示例
5. 删除功能
  - 5.1 删除功能流程图
  - 5.2 删除功能核心代码
  - 5.3 删除功能截屏示例
6. 统计功能
  - 6.1 统计功能流程图
  - 6.2 统计功能核心代码

6.3 统计功能截屏示例	
7. 主函数	
7.1 主函数流程图	
7.2 主函数核心代码	
7.3 主函数截屏示例	
四、测试	
1. 功能测试	
2. 边界测试	
2.1 初始化无输入数据	
2.2 删除头结点	
2.3 删除后链表为空	
3. 出错测试	
3.1 考生人数错误	
3.2 操作码错误	
3.3 插入位置不合法	
3.4 删除、查找、修改的考生不存在	

## 一、题目分析

---

### 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. 功能需求

本项目的实质是完成对考生信息的建立，查找，插入，修改，删除等功能。其中考生信息包括准考证号，姓名，性别，年龄和报考类别等信息。

### 3. 设计思路

首先确定项目采用链表作为数据结构，定义类的成员变量和成员函数；然后实现各成员函数以完成对数据操作的相应功能；最后完成主函数以验证各个成员函数的功能并得到运行结果。

## 二、设计

---

### 1. 数据结构设计

考试报名系统用于管理考生的相关信息，至少需要具有：建立、插入、查找、修改、删除、退出等功能。

由于系统的插入、删除等操作频繁，且考虑到链表具有以下优点：①插入和删除的效率高；②不要求存储空间连续。链表适合解决该情景下的问题，因此采用单链表作为数据结构。

## 2. 类的设计

为了实现链表数据结构，需要实现两个类：LinkNode（结点类）和LinkedList（链表类），协同表示单链表。

结点类和链表类的聚合关系主要有以下四种：复合类；嵌套类；基类和派生类；用struct定义LinkNode类。考虑到复合类的定义方式能够提供更大的灵活性，因此采用复合的方式实现两个类的定义。

为了方便地管理结点的数据域和指针域，将数据域中的信息（考生信息）抽象成一个struct。

具体设计如下：

### 2.1 StudentData

```
struct StudentData{
    string id;
    string name;
    string sex;
    int age;
    string category;
};
```

### 2.2 LinkNode

```
class LinkNode
{
    friend class LinkedList;    //声明LinkedList类为友元类

private:
    StudentData data;          //数据元素域
    LinkNode* next;            //链指针域

public:
    //无参构造函数
    LinkNode()
    {
        this->data.id = "";
        this->data.name = "";
        this->data.sex = "";
        this->data.age = 0;
        this->data.category = "";
        this->next = nullptr;
    }
    //有参构造函数
    LinkNode(const StudentData& input, LinkNode* ptr = nullptr)
    {
        this->data = input;
        this->next = ptr;
    }
};
```

```
    }  
  
};
```

## 2.3 LinkList

```
class LinkList  
{  
private:  
    LinkNode* first;           //链表头指针  
    int cntNode;              //结点计数  
  
public:  
    LinkList(){first = new LinkNode(); cntNode = 0;} //无参构造函数  
    ~LinkList(){makeEmpty(); delete first;} //析构函数  
  
    void makeEmpty(); //将链表置为空表  
    LinkNode* getHead() const {return first;} //获取链表头  
    void outputTitle() const; //输出表格第一行  
  
    void init(); //后插法建立链表  
    LinkNode* locate(int location); //定位  
    void insert(); //插入  
    void search(); //查找  
    void modify(); //修改  
    void deleteData(); //删除  
    void output() const; //统计  
  
};
```

## 3. 主函数设计

实例化一个LinkList对象用于实现考试报名系统，首先通过后插法建立链表并显示建立结果，然后通过一个while循环实现系统的功能。每层while循环内，根据用户选择的命令实现不同的功能。

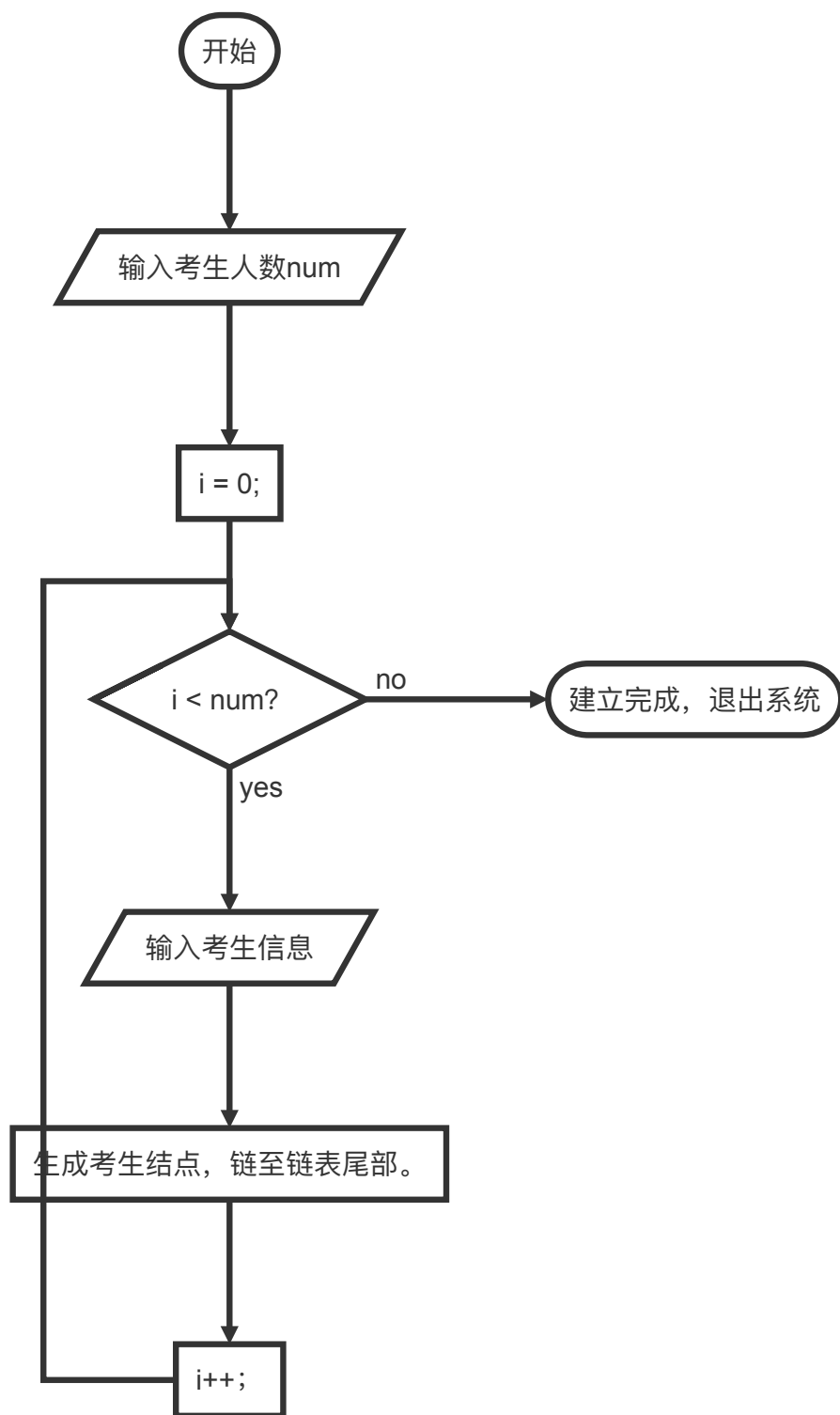
## 三、实现

### 1. 建立系统

采用后插法建立链表。

后插法需要设置一个尾指针 last，总是指向新链表中最后一个结点，新链表链接到它所指链尾结点的后面。last 最初要置于附加头结点位置。

#### 1.1 建立功能流程图



## 1.2 建立功能核心代码

```
/*  
    @brief 建立考生信息系统, 实质是采用后插法建立单链表  
*/  
void LinkList::init()  
{  
    LinkNode* newNode, *last = this->first;
```

```

std::cout<<"首先请建立考生信息系统! "<<std::endl<<"请输入考生人数: ";
int cntStudent;
std::cin>>cntStudent;

std::cout<<"请依次输入考生的考号, 姓名, 性别, 年龄及报考类别! "<<std::endl;

StudentData tempData;                                //用于暂时存储输入数据
for(int cnt=1; cnt<=cntStudent; cnt++)
{

std::cin>>tempData.id>>tempData.name>>tempData.sex>>tempData.age>>tempData
.category;
    newNode = new LinkNode(tempData);
    if(newNode == nullptr)
    {
        std::cerr<<"存储分配错误! "<<std::endl;
        exit(1);
    }
    last->next = newNode;
    last = newNode;
    this->cntNode++;                                    //链表结点数+1
}
}

```

### 1.3 建立功能截屏示例

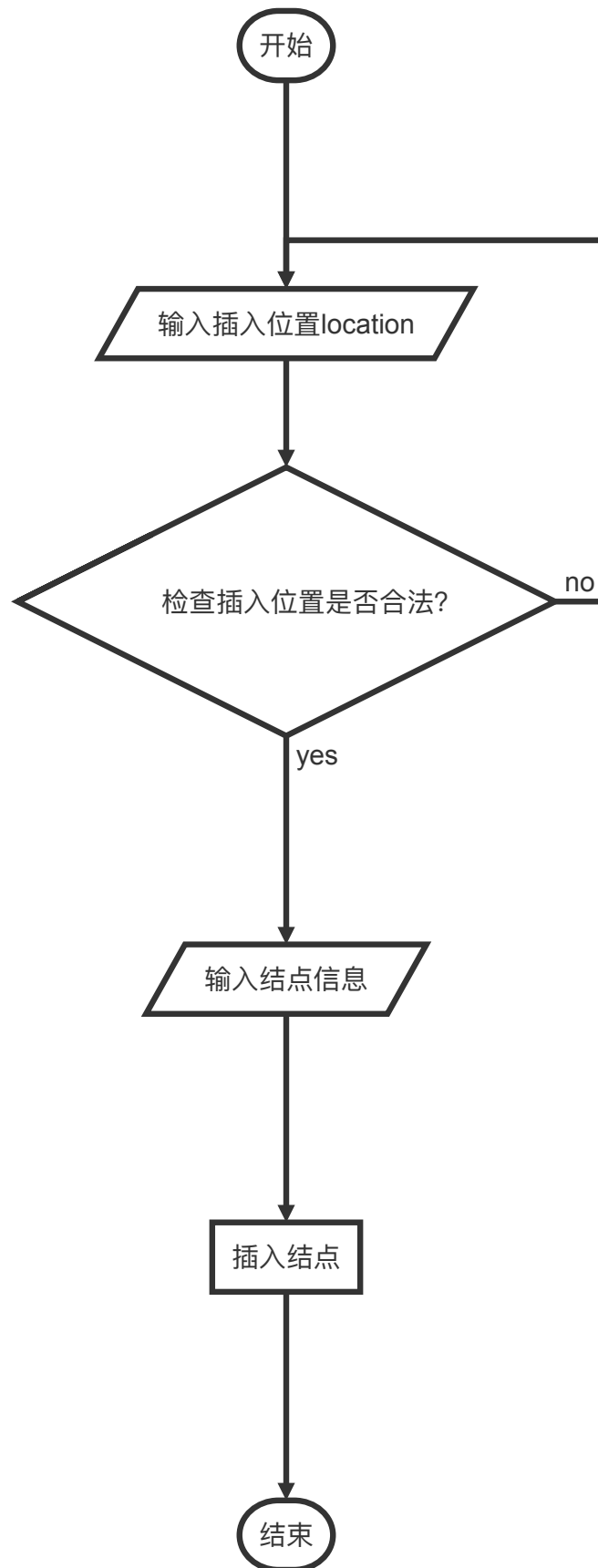
```

首先请建立考生信息系统!
请输入考生人数: 5
请依次输入考生的考号, 姓名, 性别, 年龄及报考类别!
1 stu1 女 20 软件设计师
2 stu2 男 21 软件工程师
3 stu3 女 19 软件开发师
5 stu5 男 23 群众演员
6 stu6 男 22 测试工程师
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      软件工程师
3        stu3    女      19      软件开发师
5        stu5    男      23      群众演员
6        stu6    男      22      测试工程师

```

## 2. 插入功能

### 2.1 插入功能流程图



## 2.2 插入功能核心代码

```

/*
@brief 插入函数，将新元素newData插入在链表中第location个结点之后，最后输入数据库
*/
void LinkList::insert()
{
    int location;
    StudentData newData;

    std::cout<<"请输入你要插入的考生的位置： ";
    std::cin>>location;
    LinkNode* current = locate(location - 1);
    while (current == nullptr)
    {
        std::cout<<"您输入的位置不合法，请重新输入： ";
        std::cin>>location;
        current = locate(location - 1);
    }
    std::cout<<"请依次输入要插入的考生的考号，姓名，性别，年龄及报考类别！ "
    <<std::endl;

    std::cin>>newData.id>>newData.name>>newData.sex>>newData.age>>newData.cate
    gory;
    LinkNode* newNode = new LinkNode(newData);
    if(newNode == nullptr)
    {
        std::cerr<<"存储分配错误！ " <<std::endl;
        exit(1);
    }
    newNode->next = current->next;
    current->next = newNode;
    this->cntNode++;
    output();
}
//链表结点数+1，插入成功

```

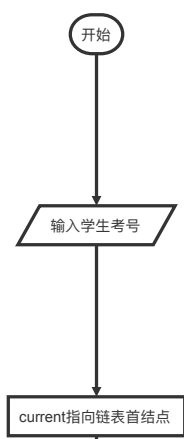
## 2.3 插入功能截屏示例

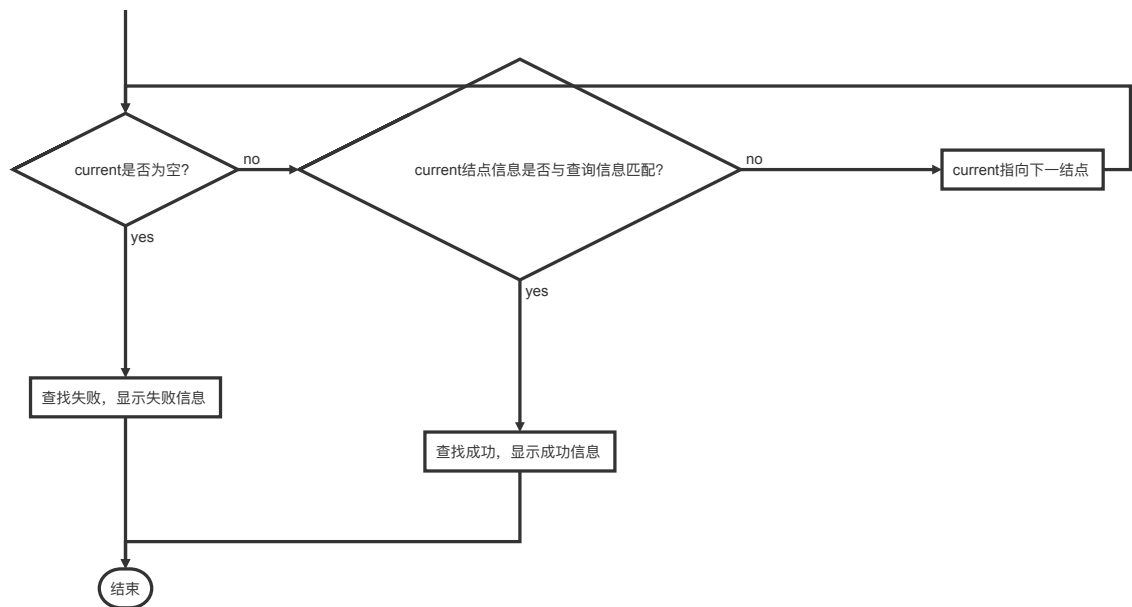


```
☐ ☐ ☐ ☐ ☐ ☐ ☐ | DataStructure_1
首先请建立考生信息系统!
请输入考生人数: 5
请依次输入考生的考号, 姓名, 性别, 年龄及报考类别!
1 stu1 女 20 软件设计师
2 stu2 男 21 软件工程师
3 stu3 女 19 软件开发师
5 stu5 男 23 群众演员
6 stu6 男 22 测试工程师
考号    姓名    性别    年龄    报考类别
1      stu1    女      20      软件设计师
2      stu2    男      21      软件工程师
3      stu3    女      19      软件开发师
5      stu5    男      23      群众演员
6      stu6    男      22      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):
1
请输入你要插入的考生的位置: 4
请依次输入要插入的考生的考号, 姓名, 性别, 年龄及报考类别!
4 stu4 男 18 实习生
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      软件工程师
3      stu3    女      19      软件开发师
4      stu4    男      18      实习生
5      stu5    男      23      群众演员
6      stu6    男      22      测试工程师
```

### 3. 查找功能

### 3.1 查找功能流程图





### 3.2 查找功能核心代码

```
/*
    @brief 查找考号为id的考生数据并输出
    */
void LinkList::search()
{
    string id;
    std::cout<<"请输入要查找的考生的考号: ";
    std::cin>>id;
    LinkNode* current = this->first->next;
    while (current != nullptr)
    {
        if(current->data.id == id)
        {
            outputTitle();
            std::cout<<setiosflags(ios::left);
```

```

        std::cout <<setw(7)<<current->data.id <<setw(8)<<current->data.name <<setw(8)<<current->data.sex <<setw(7)<<current->data.age <<setw(7)<<current->data.category << std::endl;
        return;
    }
    else
    {
        current = current->next;
    }
}
std::cout<<"您要查找的考生不存在。"<<std::endl;
}

```

### 3.3 查找功能截屏示例

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	软件工程师
3	stu3	女	19	软件开发师
4	stu4	男	18	实习生
5	stu5	男	23	群众演员
6	stu6	男	22	测试工程师

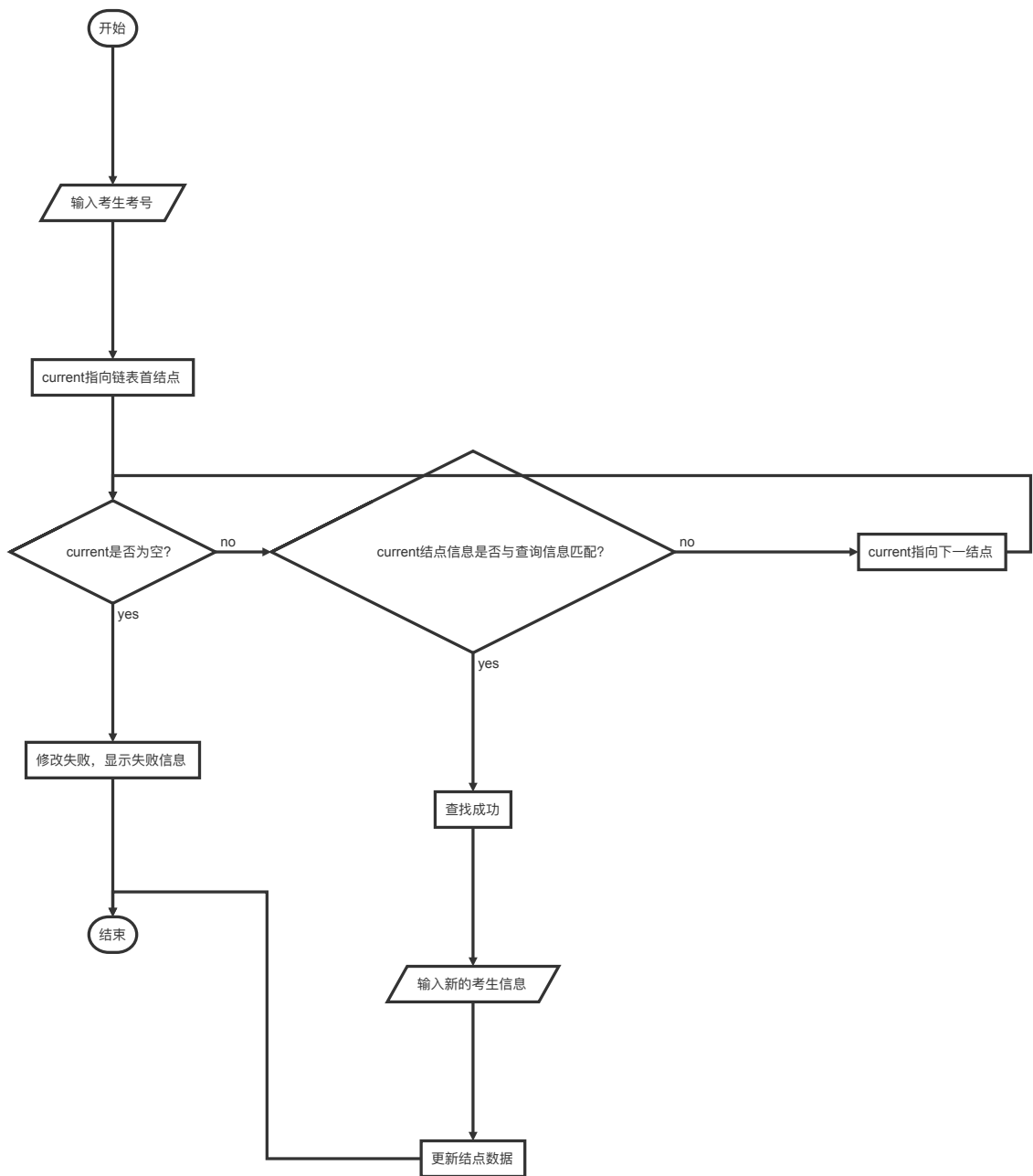
请选择您要进行的操作（1为插入，2为删除，3为查找，4为修改，5为统计，0为取消操作）：  
3  
请输入要查找的考生的考号：5

考号	姓名	性别	年龄	报考类别
5	stu5	男	23	群众演员

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  
请输入要查找的考生的考号：7  
您要查找的考生不存在。

## 4. 修改功能

### 4.1 修改功能流程图



## 4.2 修改功能核心代码

```
/*
@brief 修改考号为id的考生数据并输出修改后的数据库
*/
void LinkList::modify()
{
    string id;
    std::cout<<"请输入要修改的考生的考号： ";
    std::cin>>id;
    LinkNode* current = this->first->next;
    while (current != nullptr)
    {
        if(current->data.id == id)
        {
            std::cout<<"请依次输入要修改的考生的考号，姓名，性别，年龄及报考类别！ "
            <<std::endl;
            std::cin >> current->data.id >> current->data.name >> current->
            data.sex >> current->data.age >> current->data.category;
            output();
            return;
        }
        else
        {
            current = current->next;
        }
    }
    std::cout<<"您要修改的考生不存在。"<<std::endl;
}
```

## 4.3 修改功能截屏示例

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	软件工程师
3	stu3	女	19	软件开发师
4	stu4	男	18	实习生
5	stu5	男	23	群众演员
6	stu6	男	22	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）：

4

请输入要修改的考生的考号：8

您要修改的考生不存在。

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

请输入要修改的考生的考号：3

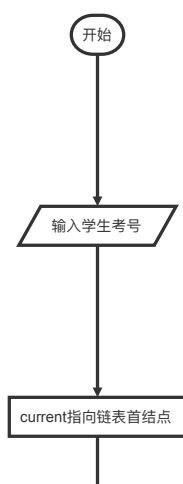
请依次输入要修改的考生的考号，姓名，性别，年龄及报考类别！

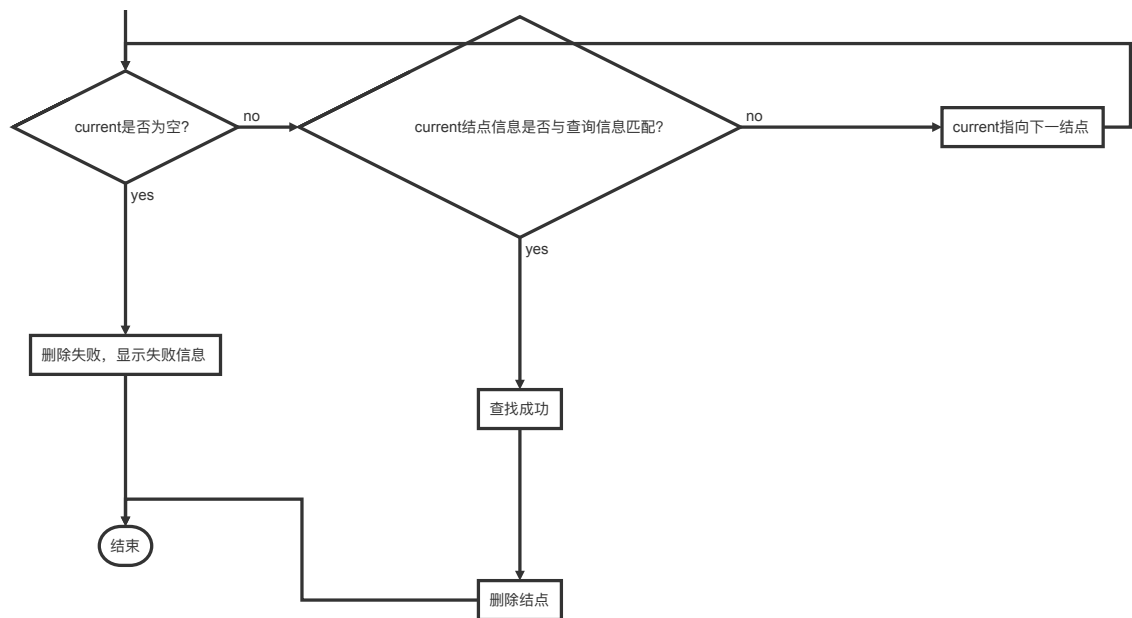
3 newStu3 男 29 硬件工程师

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	软件工程师
3	newStu3	男	29	硬件工程师
4	stu4	男	18	实习生
5	stu5	男	23	群众演员
6	stu6	男	22	测试工程师

## 5. 删除功能

### 5.1 删除功能流程图





## 5.2 删除功能核心代码

```

/*
    @brief 删除考号为id的考生数据并输出删除后的数据库
    */
void LinkList::deleteData()
{
    string id;
    std::cout<<"请输入要删除的考生的考号: ";
    std::cin>>id;
    LinkNode* current = this->first;
    while (current->next != nullptr)
    {
        if(current->next->data.id == id)
        {
            LinkNode* needDelete = current->next;
            current->next = needDelete->next;
        }
    }
}
  
```

```

        delete needDelete;
        this->cntNode--;
        output();
        return;
    }
    else
    {
        current = current->next;
    }
}
std::cout<<"您要删除的考生不存在。"<<std::endl;
}

```

### 5.3 删除功能截屏示例

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	软件工程师
3	newStu3	男	29	硬件工程师
4	stu4	男	18	实习生
5	stu5	男	23	群众演员
6	stu6	男	22	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

请输入要删除的考生的考号：9

您要删除的考生不存在。

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

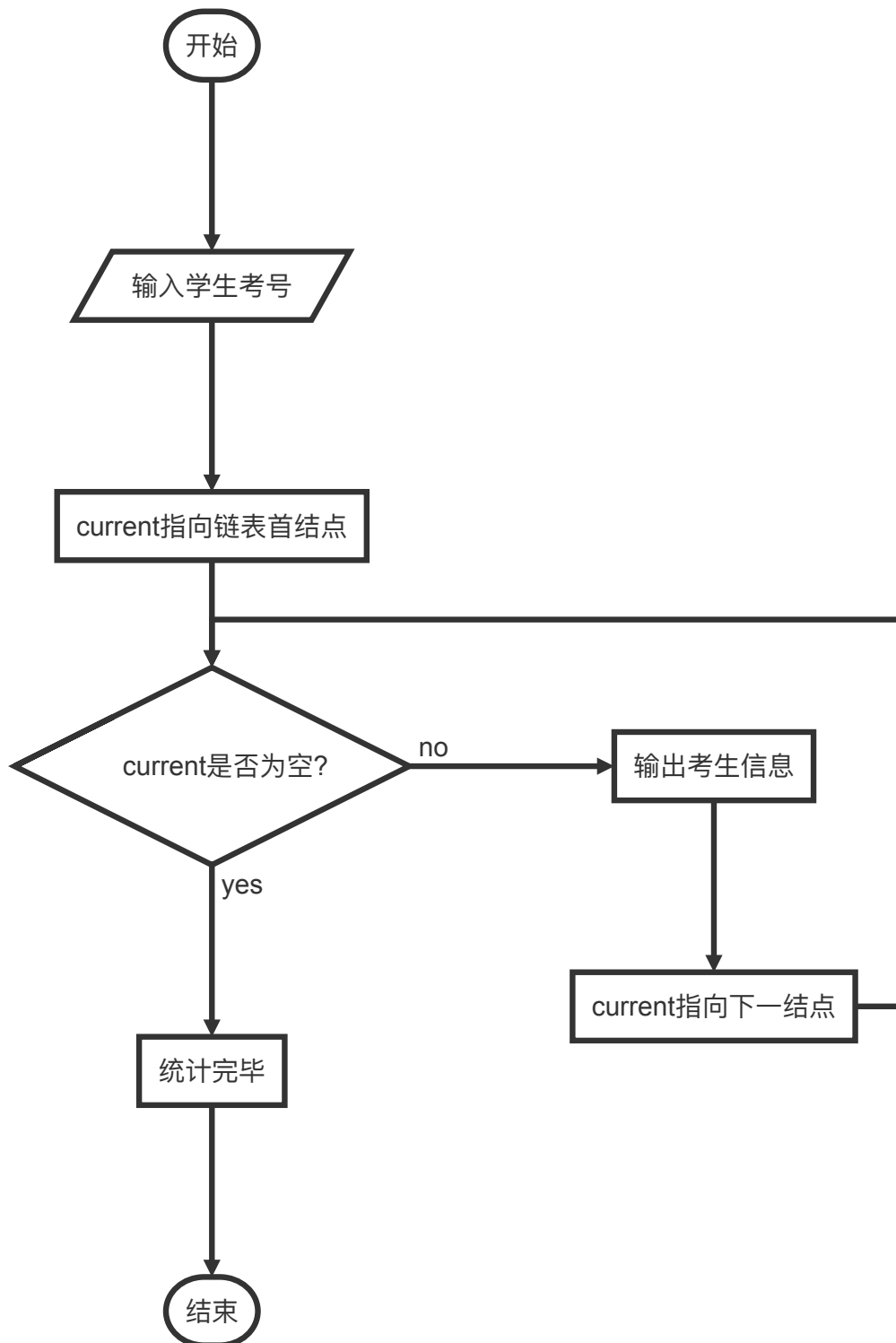
请输入要删除的考生的考号：6

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	软件工程师
3	newStu3	男	29	硬件工程师
4	stu4	男	18	实习生
5	stu5	男	23	群众演员

## 6. 统计功能

### 6.1 统计功能流程图





## 6.2 统计功能核心代码

```
/*  
    @brief 统计所有考生信息，按顺序输出链表数据  
    */  
void LinkList::output() const  
{
```

```

outputTitle();

LinkNode* current = first->next;
while(current != nullptr)
{
    std::cout<<setiosflags(ios::left);
    std::cout <<setw(7)<<current->data.id <<setw(8)<<current->data.name
    <<setw(8)<<current->data.sex <<setw(7)<<current->data.age <<setw(7)
    <<current->data.category << std::endl;
    current = current->next;
}
}

```

## 6.3 统计功能截屏示例

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）：

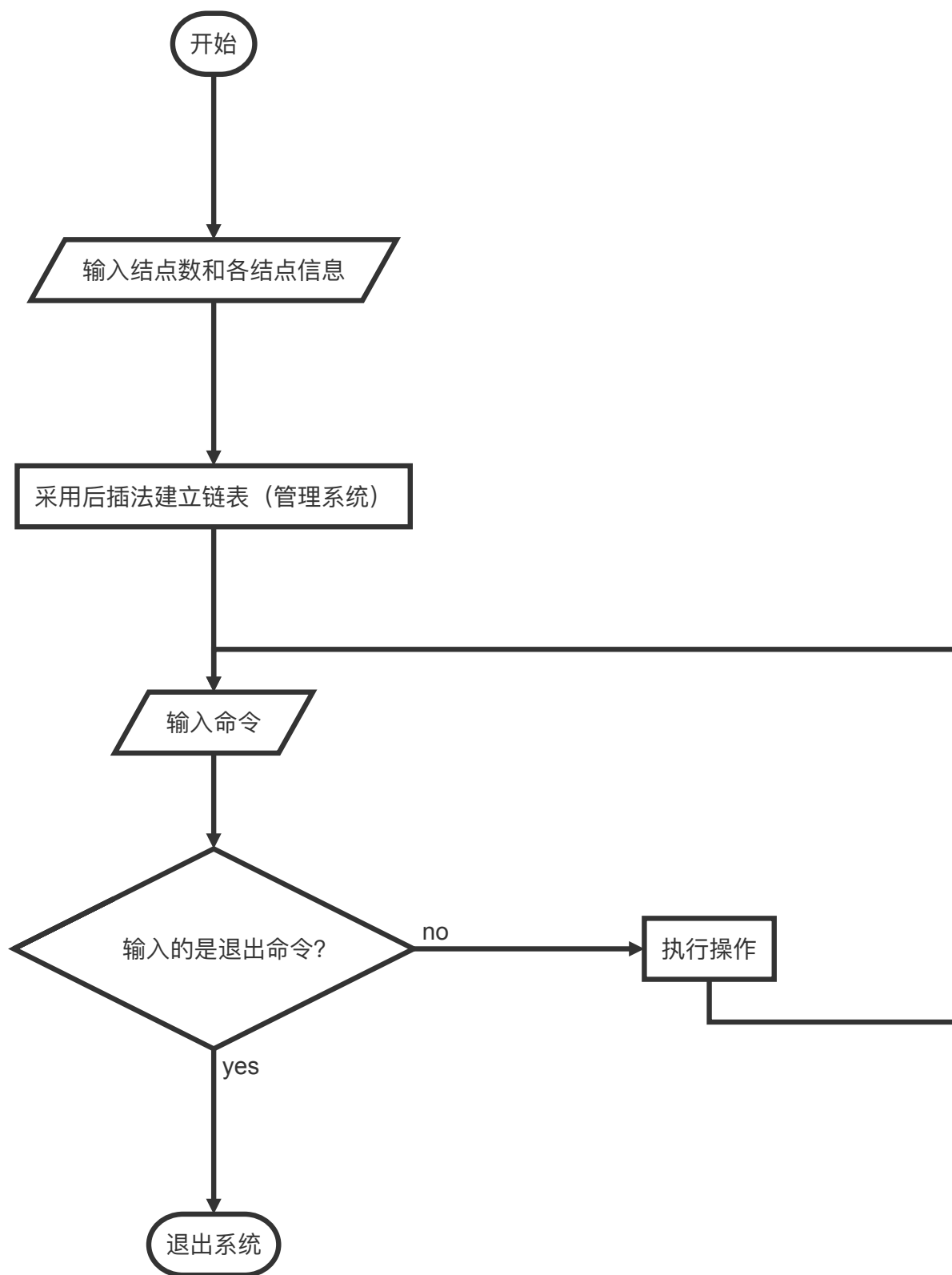
```

5
考号    姓名    性别    年龄    报考类别
1      stu1    女      20      软件设计师
2      stu2    男      21      软件工程师
3      newStu3 男      29      硬件工程师
4      stu4    男      18      实习生
5      stu5    男      23      群众演员

```

## 7. 主函数

### 7.1 主函数流程图



## 7.2 主函数核心代码

```
//建立链表并输出结果  
linkList->init();  
linkList->output();
```

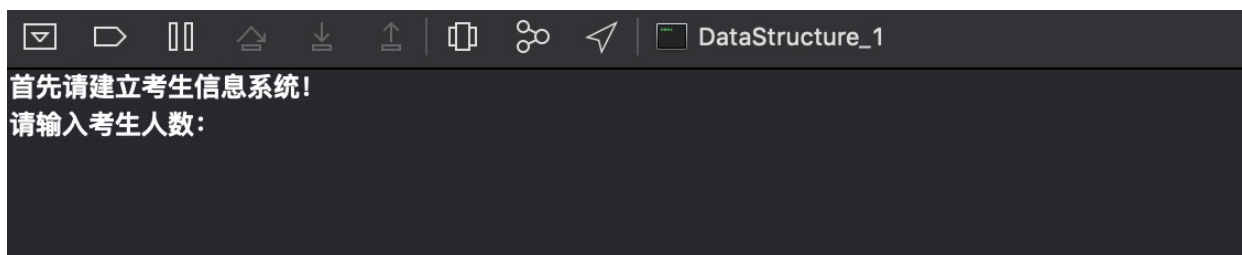
```

//循环处理命令模式
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:
            linkList->insert();
            break;
        case 2:
            linkList->deleteData();
            break;
        case 3:
            linkList->search();
            break;
        case 4:
            linkList->modify();
            break;
        case 5:
            linkList->output();
            break;
        case 0:
            return 0;

        default:
            break;
    }
}

```

### 7.3 主函数截屏示例



## 四、测试

### 1. 功能测试

在“三、实现”部分中已经通过截屏示例的方式展示了对功能的测试。经测试，建立、插入、查找、修改、删除、统计等功能均能输出正确且符合预期的结果。

### 2. 边界测试

## 2.1 初始化无输入数据

预期结果：给出错误提示，程序运行正常不崩溃。

测试结果：

```
☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ | DataStructure_1
首先请建立考生信息系统！
请输入考生人数：0
至少输入一名考生的信息，请重新输入考生人数：
```

## 2.2 删除头结点

预期结果：程序正常运行，不崩溃。

测试结果：

```
☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ | DataStructure_1
首先请建立考生信息系统！
请输入考生人数：2
请依次输入考生的考号，姓名，性别，年龄及报考类别！
1 stu1 女 20 软件设计师
2 stu2 男 21 软件工程师
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：1
考号  姓名  性别  年龄  报考类别
2     stu2  男    21    软件工程师
```

## 2.3 删除后链表为空

预期结果：程序正常运行，不崩溃。

测试结果：

```
☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ | DataStructure_1
首先请建立考生信息系统！
请输入考生人数：2
请依次输入考生的考号，姓名，性别，年龄及报考类别！
1 stu1 女 20 软件设计师
2 stu2 男 21 软件工程师
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：1
考号  姓名  性别  年龄  报考类别
2     stu2  男    21    软件工程师
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：2
考号  姓名  性别  年龄  报考类别
```

### 3. 出错测试

#### 3.1 考生人数错误

测试结果：

```

[Icons] DataStructure_1
首先请建立考生信息系统！
请输入考生人数：-1
至少输入一名考生的信息，请重新输入考生人数：

```

#### 3.2 操作码错误

测试结果：

```

[Icons] DataStructure_1
首先请建立考生信息系统！
请输入考生人数：2
请依次输入考生的考号，姓名，性别，年龄及报考类别！
1 stu1 女 20 软件设计师
2 stu2 男 21 软件工程师
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）：
8
操作码错误，请重新选择
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）：

```

#### 3.3 插入位置不合法

测试结果：

```

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
请输入你要插入的考生的位置：4
您输入的位置不合法，请重新输入：|

```

#### 3.4 删除、查找、修改的考生不存在

测试结果：

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	软件工程师
3	stu3	女	19	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）：

2

请输入要删除的考生的考号：5

您要删除的考生不存在。

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

请输入要查找的考生的考号：4

您要查找的考生不存在。

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

请输入要修改的考生的考号：6

您要修改的考生不存在。

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）：