
项目说明文档

数据结构课程设计

——考试报名系统

作者姓名：_____杨鑫_____

学 号：_____1950787_____

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

1 分析	1
1.1 背景分析	1
1.2 功能分析	1
2 设计	1
2.1 数据结构设计	1
2.2 类结构设计	2
2.3 成员与操作设计	2
2.4 系统设计	4
3 实现	4
3.1 插入功能的实现	4
3.2 删除功能的实现	5
3.3 查找功能的实现	6
3.4 修改功能的实现	7
3.5 统计功能的实现	8
4 测试	9
4.1 功能测试	9
4.2 边界测试	10
4.3 出错测试	11

1 分析

1.1 背景分析

对于每一所学校而言，都有着举行考试的需求。因此，考试报名系统也就成为了一个学校所不可或缺的一部分。同时，也正因为它对于学校的教务处的重要作用，一个好的考试报名系统应该能为用户提供充分的功能。

而事实上，由于数量逐渐增加的考试人数，考试报名工作给当前的高校报名工作带来了新的挑战，也给教务管理部门增加了很大的工作量。例如在我们日常生活中，比较常接触到的考试就有四、六级考试、普通话考试、计算机水平考试等等，这些考试报名者都需要能够合理的管理，从而实现分配考场等功能。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 数据结构设计

在实际生活中，考试报名系统会涉及到大量的增删查改的工作，而我们知道数组在增加元素和删除元素方面的性能非常差。所以为了尽量提高系统的性能，我采用了链表的数据结构来进行设计，这样在增删查改的时候，所花费的时间复杂度都是 $O(n)$ ，性能比较好，可以满足实际工作的需要。同时为了简化操作，我将其设计成为了带附加头结点的双向链表，这样在编写代码的时候可以简化一些操作，也便于阅读理解。

2.2 类结构设计

实现链表需要两个最基本的类：链表节点（InfoNode 类）和链表类（InfoList 类）。InfoNode 类里面存储学生的一些基本信息和前后指针，InfoList 类则包含一个整个链表的头结点并实现增删查改等基本操作。同时为了更加方便的处理，将学生所有信息整合到一个类（StudentInfo 类）里面，设计了系统类（System 类）用于和用户交互并控制信息链表，避免用户直接操控链表产生一些异常，实现更好的封装性。

2.3 成员与操作设计

学生信息类（StudentInfo）

私有成员：

```
1. string ID; //考号
2. string name; //姓名
3. string gender; //性别
4. int age; //年龄
5. string categoryOfExamination; //报考种类
```

公有操作：

```
1. StudentInfo() = default; // 默认构造函数
2. StudentInfo(string newID, string newName, string newGender, int newAge, string newCategoryOfExamination) : ID(newID), name(newName), gender(newGender), age(newAge), categoryOfExamination(newCategoryOfExamination) {} // 带参数的构造函数
3. ~StudentInfo() {} // 析构函数，类里面不含指针，无需释放
4. StudentInfo(const StudentInfo& StuInfo); // 拷贝构造函数
5. StudentInfo& operator = (const StudentInfo& StuInfo); // 重载 = 运算符
6. friend ostream& operator << (ostream& out, const StudentInfo& StuInfo); // 友元函数，实现输出
7. friend istream& operator >> (istream& in, StudentInfo& StuInfo); // 友元函数，实现输入
8.
9. // 一些接口函数
10. string getID() { return ID; }
11. string getName() { return name; }
12. string getGender() { return gender; }
13. int getAge() { return age; }
14. string getCategoryOfExamination() { return categoryOfExamination; }
```

节点类 (InfoNode)

私有成员:

1. T data; // 数据域
2. InfoNode* pre; // 前指针
3. InfoNode* link; // 后指针

公有操作:

1. InfoNode() :pre(NULL), link(NULL) {} // 默认构造函数
2. InfoNode(T newData, InfoNode* newPre = NULL, InfoNode* newLink = NULL) : data(newData), pre(newPre), link(newLink) {} // 带参数的构造函数
3. InfoNode(const InfoNode& IN) : data(IN.data), pre(IN.pre), link(IN.link) {} // 拷贝构造函数
4. ~InfoNode() {} // 析构函数
5. friend class InfoList<T>; // 声明 InfoList 为友类
6. friend class System<T>; // 声明 System 为友类

链表类 (InfoList)

私有成员:

1. InfoNode<T>* head; // 头结点

公有操作:

1. InfoList() { head = new InfoNode<T>; } // 默认构造函数
2. ~InfoList(); // 析构函数
3. int Size(); // 返回长度
4. bool Insert(int location, InfoNode<T>* IN); // 插入
5. bool Remove(string rmID); // 删除
6. bool Search(string checkID, InfoNode<T>*& result); // 搜索
7. bool Update(string curID, T nowInfo); // 修改
8. void Print(); // 统计

系统类 (System)

私有成员:

1. InfoList<T>* Info; // 信息链表

公有操作:

1. ~System() { delete Info; } // 程序退出时释放空间
2. void init(); // 初始化系统
3. void loop(); // 主循环 (即反复询问用户的操作并执行)
4. void insert(); // 执行插入指令
5. void remove(); // 执行删除指令
6. void update(); // 执行修改指令

```
7. void search(); // 执行查找指令
8. void show(); // 执行统计指令
```

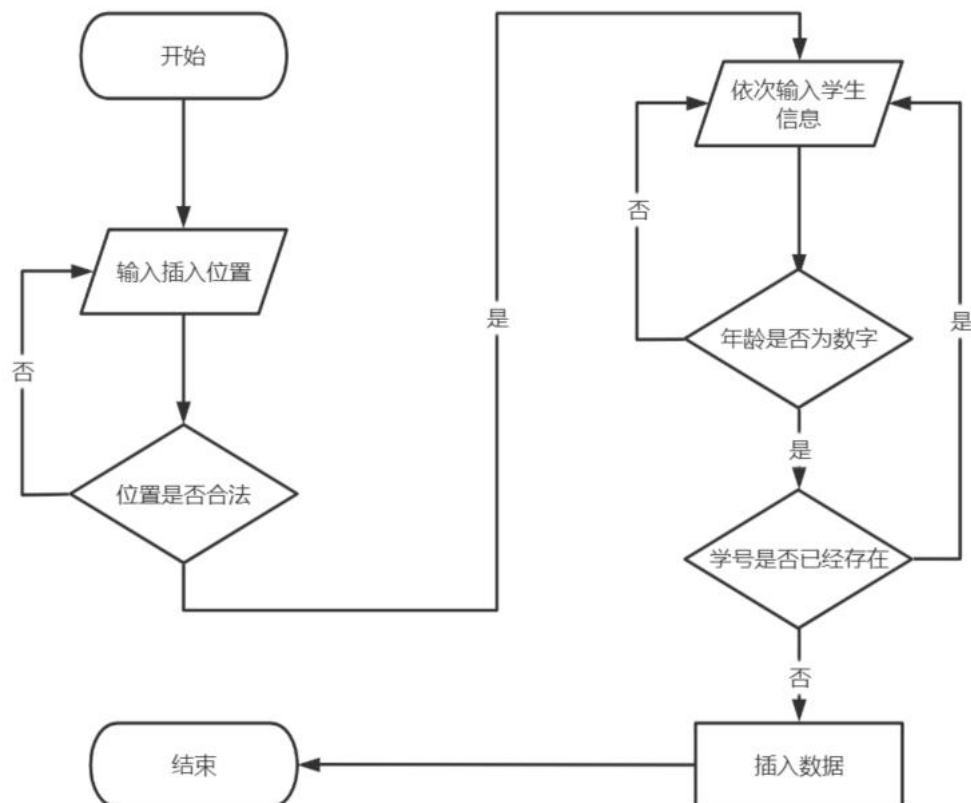
2.4 系统设计

开始程序时首先调用系统类的 init（）函数进行初始化，并要求用户建立一个原始的信息表。完成后不断接受用户的指令完成相关操作直到用户给出结束指令。程序兼容了 windows 和 LINUX 平台，在双平台下均可以正常运行。

3 实现

3.1 插入功能的实现

3.1.1 插入功能流程图



3.1.2 插入功能核心代码

```
1. IN->link = cur->link;
2. if (cur->link != NULL) cur->link->pre = IN;
3. IN->pre = cur;
4. cur->link = IN;
5. return true;
```

3.1.3 插入功能截屏示例

```
首先请建立考生信息系统！
请输入考生人数：2
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！
1001 john male 19 math
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！
1002 mary female 20 English
建立成功！

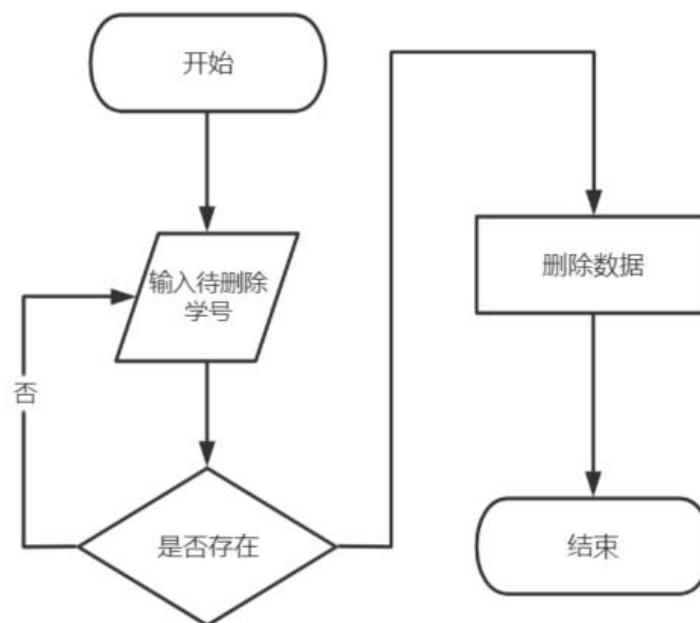
请选择您要进行的操作（1为插入，2为删除，3为查找，4为修改，5为统计，0为取消操作）
1
请输入您要插入的考生的位置！
2
请依次输入要插入的考生的考号，姓名，性别，年龄及报考类别！
1003 jack male 19 Chinese
插入成功！

考号    姓名    性别    年龄    报考类别
1001    john    male    19      math
1002    mary    female  20      English
1003    jack    male    19      Chinese

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）
```

3.2 删除功能的实现

3.2.1 删除功能流程图



3.2.2 删除功能核心代码

```
1. cur->pre->link = cur->link;
2. if (cur->link != NULL) cur->link->pre = cur->pre;
3. delete cur;
4. return true;
```

3.2.3 删除功能截屏示例

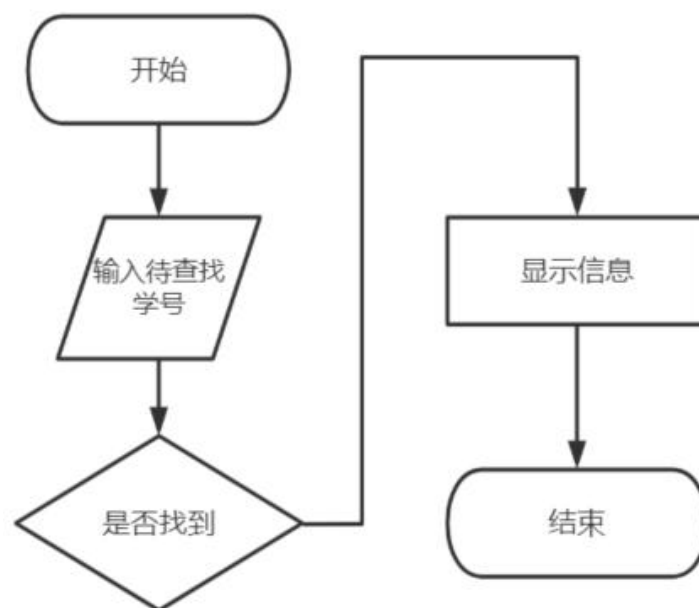
```
首先请建立考生信息系统！
请输入考生人数：2
请依次输入要插入的考生的考号，姓名，性别，年龄及报考类别！
1001 tom male 21 DataStructure
请依次输入要插入的考生的考号，姓名，性别，年龄及报考类别！
1002 amy female 20 Physics
建立成功！

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
请输入要删除的记录的学生学号：
1002
删除成功！
考号    姓名    性别    年龄    报考类别
1001    tom     male    21     DataStructure

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 查找功能的实现

3.3.1 查找功能流程图



3.3.2 查找功能核心代码

```
1. InfoNode<T>* cur = head->link;
2. // 顺序检索
3. while (cur != NULL) {
4.     if (cur->data.getID() == checkID) {
5.         result = cur;
6.         return true;
7.     }
8.     else cur = cur->link; // 搜索成功后把该节点通过引用返回
```



```

9. }
10. return false;

```

3.3.3 查找功能截屏示例

```

首先请建立考生信息系统!
请输入考生人数: 2
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!
1001 jack male 21 math
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!
1002 mary female 20 english
建立成功!

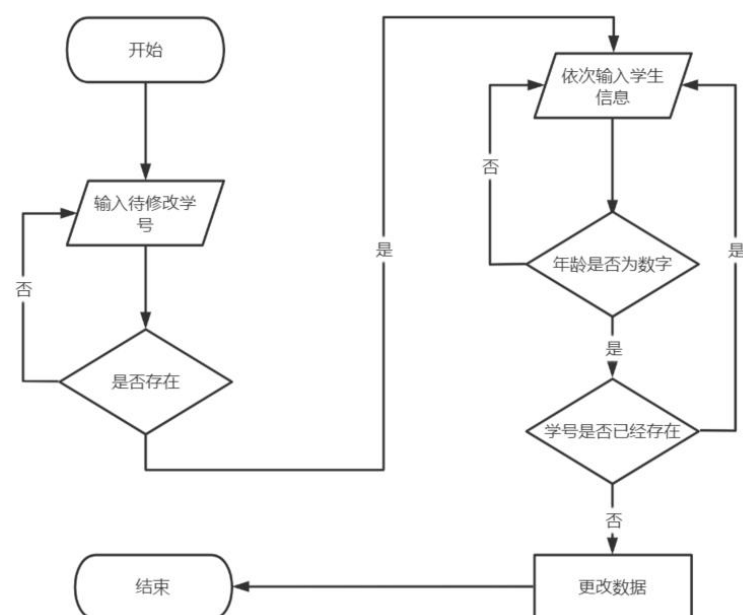
请选择您要进行的操作 (1为插入, 2为删除, 3为查找, 4为修改, 5为统计, 0为取消操作)
3
请输入待查询学生的学号! (输入 q 以退出本次操作)
1001
查询成功! 信息如下:

考号  姓名  性别  年龄  报考类别
1001  jack  male  21    math
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)

```

3.4 修改功能的实现

3.4.1 修改功能流程图



3.4.2 修改功能核心代码

```

1. InfoNode<T>* cur = head->link;
2. // 顺序检索
3. while (cur != NULL) {
4.     if (cur->data.getID() == curID) break;

```

```

5.     else cur = cur->link;
6. }
7. if (cur == NULL) {
8.     cout << "被修改的学号不存在！" << endl;
9.     return false;
10. }
11. cur->data = nowInfo;

```

3.4.3 修改功能截屏示例

```

首先请建立考生信息系统！
请输入考生人数：1
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！
1001 jack male 21 english
建立成功！

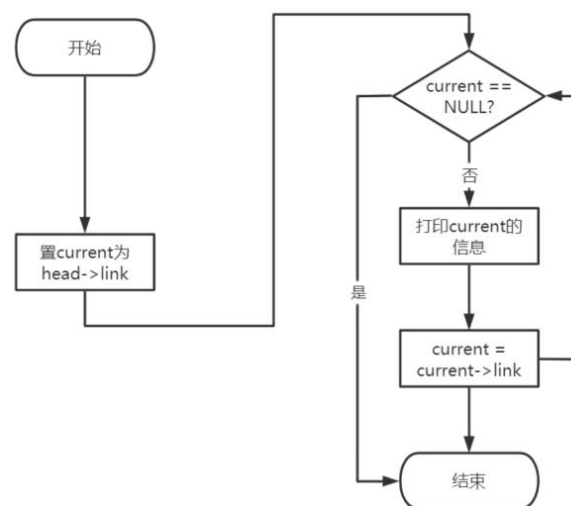
请选择您要进行的操作（1为插入，2为删除，3为查找，4为修改，5为统计，0为取消操作）
4
请输入要修改的记录的学生学号：
1001
请依次输入修改后的考号，姓名，性别，年龄及报考类别！
1001 jack male 20 math
修改成功！
考号    姓名    性别    年龄    报考类别
1001    jack    male    20     math

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）

```

3.5 统计功能的实现

3.5.1 统计功能流程图



3.5.2 统计功能核心代码

```

1. // 依次打印所有节点信息
2. while (cur != NULL) {
3.     cout << cur->data << endl; // 利用重载的 << 运算符打印信
   息
4.     cur = cur->link;

```

```
5. }  
6. cout << endl;
```

3.5.3 统计功能截屏示例

```
首先请建立考生信息系统!  
请输入考生人数: 3  
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!  
1001 jack male 21 math  
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!  
1002 mary female 20 english  
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!  
1003 amy female 18 chinese  
建立成功!  
  
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  
1001 jack male 21 math  
1002 mary female 20 english  
1003 amy female 18 chinese  
  
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 测试

4.1 功能测试

4.1.1 插入功能测试

运行结果截图:

```
首先请建立考生信息系统!  
请输入考生人数: 2  
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!  
1001 john male 19 math  
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!  
1002 mary female 20 English  
建立成功!  
  
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  
请输入您要插入的考生的位置!  
2  
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!  
1003 jack male 19 Chinese  
插入成功!  
考号 姓名 性别 年龄 报考类别  
1001 john male 19 math  
1002 mary female 20 English  
1003 jack male 19 Chinese  
  
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)
```

4.1.2 删除功能测试

运行结果截图:

```
首先请建立考生信息系统!  
请输入考生人数: 2  
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!  
1001 tom male 21 DataStructure  
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!  
1002 amy female 20 Physics  
建立成功!  
  
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  
请输入要删除的记录的学生学号:  
1002  
删除成功!  
考号 姓名 性别 年龄 报考类别  
1001 tom male 21 DataStructure  
  
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)
```

4.1.3 查找功能测试

运行结果截图：

```
首先请建立考生信息系统！
请输入考生人数：2
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！
1001 jack male 21 math
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！
1002 mary female 20 english
建立成功！

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
请输入待查询学生的学号！（输入 Q 以退出本次操作）
1001
查询成功！信息如下：

考号    姓名    性别    年龄    报考类别
1001    jack    male    21      math
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）
```

4.1.4 修改功能测试

运行结果截图：

```
首先请建立考生信息系统！
请输入考生人数：1
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！
1001 jack male 21 english
建立成功！

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
请输入要修改的记录的学生学号：
1001
请依次输入修改后的考号，姓名，性别，年龄及报考类别！
1001 jack male 20 math
修改成功！

考号    姓名    性别    年龄    报考类别
1001    jack    male    20      math
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）
```

4.1.5 统计功能测试

运行结果截图：

```
首先请建立考生信息系统！
请输入考生人数：3
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！
1001 jack male 21 math
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！
1002 mary female 20 english
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！
1003 amy female 18 chinese
建立成功！

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
1001    jack    male    21      math
1002    mary    female  20      english
1003    amy     female  18      chinese
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）
```

4.2 边界测试

4.2.1 初始化无输入数据

处理方式：当输入人数为 0 或者负数时会提示用户输入不合要求，程序不会异常退出或者崩溃。

运行结果截图：

```

首先请建立考生信息系统！
请输入考生人数：0
考生人数只能是正整数！
请重新输入考生人数：-1
考生人数只能是正整数！
请重新输入考生人数：

```

4.2.2 删除第一个结点

处理方式：系统正常运行，程序不会异常退出或者崩溃。

运行结果截图：

```

首先请建立考生信息系统！
请输入考生人数：2
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！
1001 jack male 19 math
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！
1002 mary female 20 english
建立成功！

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
请输入要删除的记录的学生学号：
1001
删除成功！
考号    姓名    性别    年龄    报考类别
1002    mary    female  20     english

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）

```

4.2.3 删除后链表为空

处理方式：系统正常运行，程序不会异常退出或者崩溃。

运行结果截图：

```

首先请建立考生信息系统！
请输入考生人数：1
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！
1001 jack male 20 math
建立成功！

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
请输入要删除的记录的学生学号：
1001
删除成功！
考号    姓名    性别    年龄    报考类别

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）

```

4.3 出错测试

4.3.1 考生人数错误

处理方式：当输入人数为 0 或者负数时会提示用户输入不合要求，程序不会异常退出或者崩溃。

运行结果截图：

```

首先请建立考生信息系统！
请输入考生人数：0
考生人数只能是正整数！
请重新输入考生人数：-1
考生人数只能是正整数！
请重新输入考生人数：

```

4.3.2 操作码错误

处理方式：系统会提示用户重新输入，程序不会异常退出或者崩溃。

运行结果截图：

```
首先请建立考生信息系统！
请输入考生人数：1
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！
1001 jack male 19 math
建立成功！

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）
0
输入操作数有误，请尝试重新输入！
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）
```

4.3.3 插入位置不存在

处理方式：系统会提示用户重新输入，程序不会异常退出或者崩溃。

运行结果截图：

```
首先请建立考生信息系统！
请输入考生人数：1
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！
1001 jack male 19 math
建立成功！

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
请输入您要插入的考生的位置！
-1
位置不合法，请重新输入：（输入 -1 以退出本次操作）
2
位置不合法，请重新输入：（输入 -1 以退出本次操作）
```

4.3.4 删除考号不存在

处理方式：系统会提示用户重新输入，程序不会异常退出或者崩溃。

运行结果截图：

```
首先请建立考生信息系统！
请输入考生人数：1
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！
1001 jack male 19 math
建立成功！

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
请输入要删除的记录的学生学号：
1002
删除元素不存在！
删除失败，输入 0 以退出本次操作，输入 1 以继续：
```

4.3.5 查找考号不存在

处理方式：系统会提示用户未查询到该考生，程序不会异常退出或者崩溃。

运行结果截图：

```

首先请建立考生信息系统！
请输入考生人数：1
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！
1001 jack male 19 math
建立成功！

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
请输入待查询学生的学号！（输入 q 以退出本次操作）
1003
未查询到该考生！
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）

```

4.3.6 修改考号不存在

处理方式：系统会提示用户重新输入或者取消本次操作，程序不会异常退出或者崩溃。

运行结果截图：

```

首先请建立考生信息系统！
请输入考生人数：1
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！
1001 jack male 19 math
建立成功！

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
请输入要修改的记录的学生学号：
2121
被修改学生不存在，请重新输入：（输入 q 以退出本次操作，输入其他任意字符以继续...）

```

4.3.7 修改考生的考号已存在

处理方式：系统会提示用户重新输入或者取消本次操作，程序不会异常退出或者崩溃。

运行结果截图：

```

首先请建立考生信息系统！
请输入考生人数：2
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！
1001 jack male 19 math
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！
1002 mary female 20 english
建立成功！

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
请输入要修改的记录的学生学号：
1001
请依次输入修改后的考号，姓名，性别，年龄及报考类别！
1002 jack male 21 math
修改后的学号已存在！
修改失败，输入 0 以退出本次操作，输入 1 以继续：

```