



同濟大學  
TONGJI UNIVERSITY

# 项目说明文档

考试报名系统

指导教师：张颖

1751984 王舸飞

## 1.分析

### 1.1 背景分析

### 1.2 功能分析

## 2.设计

### 2.1 数据结构设计

### 2.2 类结构设计

### 2.3 成员与操作设计

### 2.4 系统设计

## 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

#### 3.1.3 插入功能截屏示例

### 3.2 删除功能的实现

#### 3.1.1 删除功能流程图

#### 3.1.2 删除功能核心代码

#### 3.1.3 删除功能截屏示例

### 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

#### 3.3.3 查找功能截屏示例

### 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

#### 3.4.3 修改功能截屏示例

### 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

#### 3.5.3 统计功能截屏示例

## 3.6 总体功能的实现

### 3.6.1 总体功能流程图

### 3.6.2 总体功能核心代码

### 3.6.3 总体功能截屏示例

## 4.测试

### 4.1 错误测试

#### 4.1.1 考生人数错误

#### 4.1.2 输入操作码错误

#### 4.1.3 插入位置不存在

#### 4.1.4 在空表中删除

#### 4.1.5 删除学号不存在

#### 4.1.6 查找表为空

#### 4.1.7 修改考号不存在

# 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

---

本次设计的项目虽然只是对考试报名系统的简单模拟，但也应该具有一套考试系统的基本功能，鉴于使用的对象是数据库的管理者，它所具有的功能应包括数据的输入、输出（展示）、查询、添加、修改和删除。

另外，在输入的数据非法或出现意料之外的操作时，应使程序停止运行并报错，防止程序的异常崩溃，保证程序的稳定性。

# 2.设计

## 2.1 数据结构设计

---

上述要实现的功能包括数据的插入、删除和修改，如果采用顺序表的方式存储数据，将会使这些操作的时间复杂度大大增加，因此可以基本确定链表为更合适的选择。在此基础上，为了减少编程工作量并提高程序的可读性，在链表之前添加一个附加头节点。

## 2.2 类结构设计

---

经典的链表一般包括两个抽象数据类型（ADT）——链表结点类（LNode）与链表类

（LinkList），而两个类之间的耦合关系可以采用嵌套、继承等多种关系。本程序中将链表节点类作为链表类的友元类，使得链表可以访问链表节点。

## 2.3 成员与操作设计

---

### 链表节点类 (Student)

#### 受保护的成员:

```
string number;    //用于表示学生的学号，为了防止非法输入均采用string
string name;      //用于表示学生的姓名
string sex;       //用于表示学生的性别
string age;       //用于表示学生的年龄，
string test_type; //用于表示学生的报考类别
Student *next;    //链表节点的指针域
```

#### 公有操作:

```
Student(string p_number="",
string p_name="",
string p_sex="",
string p_age="",
string p_test_type="",
Student *p_next=NULL):
number(p_number),
name(p_name),
sex(p_sex),
age(p_age),
test_type(p_test_type),
next(p_next){}    //用参数初始化表的形式完成构造函数
void Set_value(string,string,string,string,string);//赋值函数
```

### 链表类 (Student\_list)

#### 受保护的成员:

```
Student *first; //链表的头指针
int list_length; //表长，可增加代码可读性
```

#### 公有操作:

```
Student_list(){first=new Student;}    //构造函数，有一个附加头节点
~Student_list(){Make_empty();} //与Make_empty()函数结合的析构函数，释放内存防止内存泄露
void Set_length(int);    //设置链表的长度
void Input_data();    //输入链表长度个数据，采用后插法
void Make_empty();    //收回内存
void Display()const; //显示数据
void Insert();    //插入一个数据
void Delete();    //删除一个数据
void Change();    //改变一个数据
void Find()const; //查找一个数据
```

## 2.4 系统设计

---

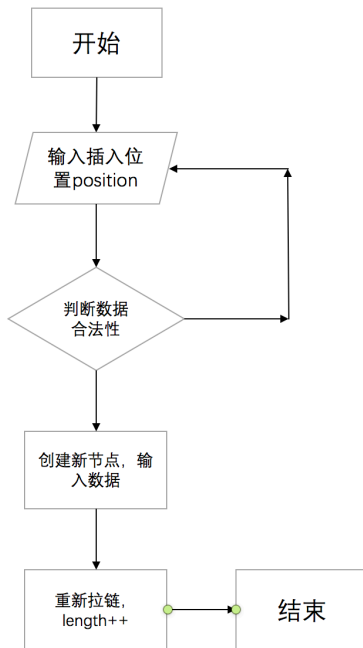
程序运行之后，系统会立刻创建一个链表类s1，之后要求用户输入所要建立表的人数和考生信息，在表建立后，通过用户输入的操作码执行相关成员函数，实现功能。

# 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

```
newstudent->Set_value(p_number,p_name,p_sex,p_age,p_test_type);
```

```
newstudent->next=current->next;    //拉链, 由于有头指针, 不必特殊分类
```

```
current->next=newstudent;
```

```
list_length++;
```

### 3.1.3 插入功能截屏示例

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）

1

请输入您要插入的考生位置：2

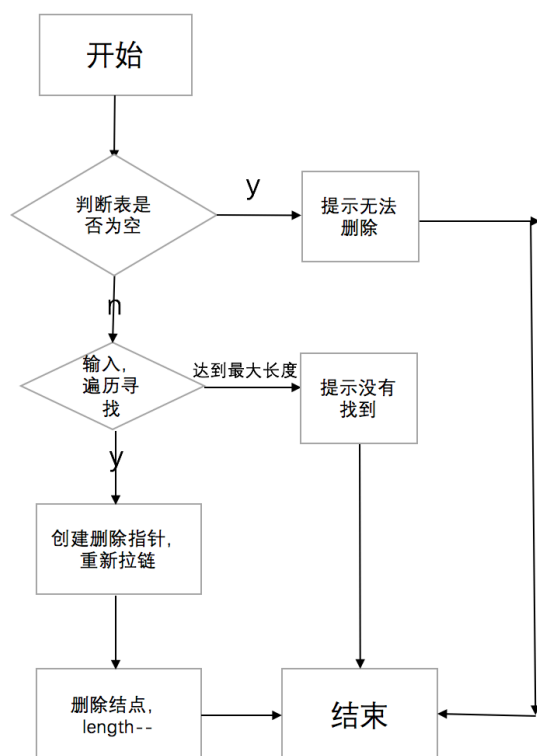
请依次输入要插入学生的学号、姓名、性别、年龄和报考类别：

3 stu3 male 20 network

考号	姓名	性别	年龄	报考类别
1	stu1	male	20	software
3	stu3	male	20	network
2	stu2	female	21	test

## 3.2 删除功能的实现

### 3.1.1 删除功能流程图



### 3.1.2 删除功能核心代码

```

void Student_list::Delete(){
    if(list_length==0)cout<<"这个表已经没有可以删除的项了"<<endl; //先判断是否为空，再进行删除操作
    else{string d_number;Student *current=first;Student *d_student;
    cout<<"请输入你要删除的考生考号：";cin>>d_number;
    for(int i=1;i<=list_length;i++){ //用for循环遍历寻找，调用strcmp函数比较
        if(strcmp(current->next->number.c_str(),d_number.c_str())==0)
        {d_student=current->next; //找到后，拉链
        current->next=d_student->next;
        cout<<"你删除的数据是："<<d_student->number<<"      "<<d_student->name<<"      "<<d_student->sex<<"      "<<d_student->
        >age<<"      "<<d_student->test_type<<endl;
        cout<<"删除后的表为："<<endl;Display();
        delete d_student;list_length--;break; //删除后，退出循环，不参与下面的判断语句
        }
        if(i==list_length)cout<<"未找到删除学号，故本次没有删除任何数据"<<endl;
        current=current->next;
    }
}
}
}

```

### 3.1.3 删除功能截屏示例

考号	姓名	性别	年龄	报考类别
1	stu1	male	21	software
2	stu2	female	23	test

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

请输入你要删除的考生考号：1

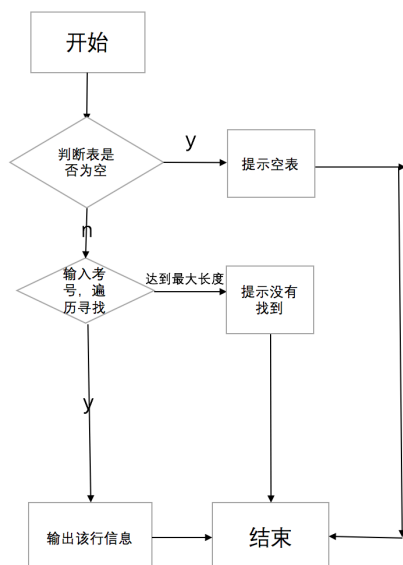
你删除的数据是：1      stu1      male      21      software

删除后的表为：考号      姓名      性别      年龄      报考类别

2	stu2	female	23	test
---	------	--------	----	------

## 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

```
void Student_list::Find()const{
    if(list_length==0)cout<<"你要在空表里找什么呢? "<<endl; //判断有值后再查找
    else{
        string f_number;Student *current=first->next;
        cout<<"请输入你要查找的考生考号: ";cin>>f_number;
        for(int i=1;i<=list_length;i++){ //for循环遍历寻找
            if(strcmp(current->number.c_str(),f_number.c_str())==0)
            { cout<<current->number<<"      "<<current->name<<"      "<<current->sex<<"      "<<current->age<<"
              "<<current->test_type<<endl;
              break; //找到后退出循环, 不参与下面的判断语句
            }
            if(i==list_length)cout<<"查无此人! "<<endl;
            current=current->next;
        }
    }
}
```

### 3.3.3 查找功能截屏示例

考号	姓名	性别	年龄	报考类别
1	stu1	male	21	software
2	stu2	female	22	test

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

请输入你要查找的考生考号：1

1	stu1	male	21	software
---	------	------	----	----------

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）

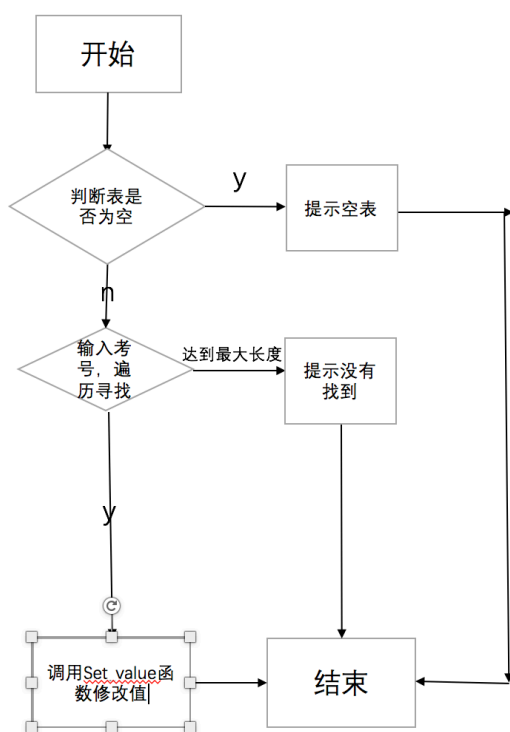
请输入你要查找的考生考号：0

查无此人！

## 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

```

void Student_list::Change(){
    if(list_length==0)cout<<"这个表已经没有可以修改的项了"<<endl; //先看表是否为空, 有时进行操作
    else{string c_number;Student *current=first->next;
        cout<<"请输入要修改学生的学号: ";cin>>c_number;
        for(int i=1;i<=list_length;i++){if(strcmp(current->number.c_str(),c_number.c_str())==0){ //遍历寻找
            cout<<"请依次输入新的学号、姓名、性别、年龄和报考类别: "<<endl;
            string p_number;
            string p_name;
            string p_sex;
            string p_age;
            string p_test_type;
            cin>>p_number>>p_name>>p_sex>>p_age>>p_test_type;
            current->Set_value(p_number,p_name,p_sex,p_age,p_test_type); //修改值
            Display();
            break;
        }
        if(i==list_length)cout<<"想要修改的考号不存在"<<endl;
        current=current->next;
    }
}
}
}

```

### 3.4.3 修改功能截屏示例

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）

4

1请输入要修改学生的学号：

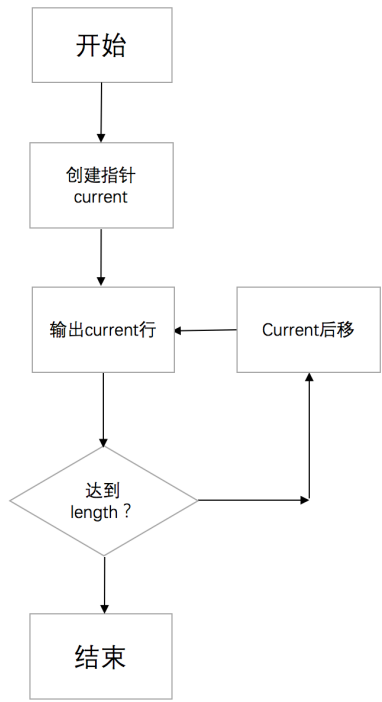
请依次输入新的学号、姓名、性别、年龄和报考类别：

3 stu3 male 23 network

考号	姓名	性别	年龄	报考类别
3	stu3	male	23	network
2	stu2	female	22	test

# 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

```
void Student_list::Display()const{Student *current; //一个普通的输出函数
cout<<"考号"<<"      "<<"姓名"<<"      "<<"性别"<<"      "<<"年龄"<<"      "<<"报考类别"<<endl;
for(current=first->next;current!=NULL;current=current->next){
    cout<<current->number<<"      ";
    cout<<current->name<<"      ";
    cout<<current->sex<<"      ";
    cout<<current->age<<"      ";
    cout<<current->test_type<<"      ";
    cout<<endl;
}
}
```

## 3.5.3 统计功能截屏示例

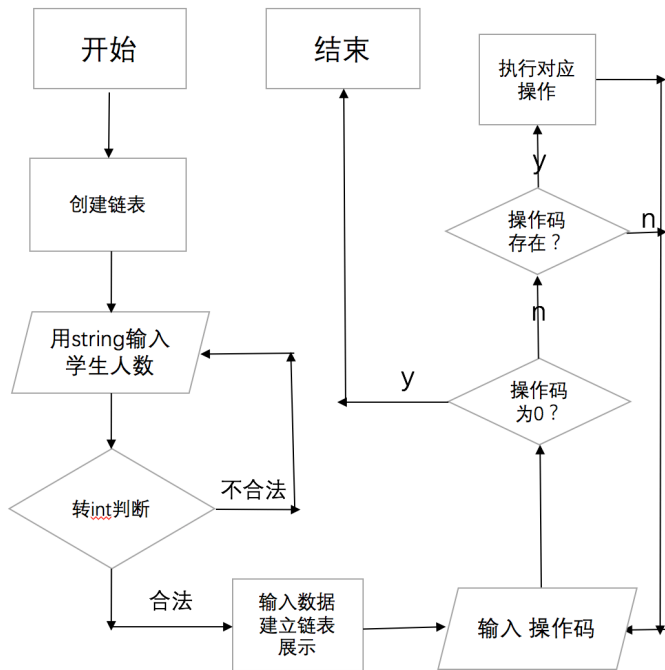
请选择您要进行的操作（1为插入，2为删除，3为查找，4为修改，5为统计，0为取消操作）

5

考号	姓名	性别	年龄	报考类别
1	stu1	male	21	software
2	stu2	female	22	test

## 3.6 总体功能的实现

### 3.6.1 总体功能流程图



### 3.6.2 总体功能核心代码

```
Student_list s1;    //创建时自动执行构造函数，结束时执行析构函数，防止内存泄露
cout<<"请输入考生人数: "<<endl;
string strlength;cin>>strlength;
int length=atoi(strlength.c_str());    //这里采用string类型输入表长，再通过atoi()函数转为int赋值，可以防止输入字母时无限循环的bug
while(length<=0){cout<<"输入的人数没有意义! 请重新输入: "<<endl;cin>>strlength;length=atoi(strlength.c_str());}
s1.Set_length(length);
cout<<"请依次输入考生的考号, 姓名, 性别, 年龄及报考类别: "<<endl;
s1.Input_data();    //调用输入函数
s1.Display();    //展示当前表
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;
char operate_code;cin>>operate_code;
while(operate_code!='0')    //当操作码为0时，程序结束
{switch(operate_code)    //对操作码进行筛选，调用对应的函数
{
    case '1':s1.Insert();break;
    case '2':s1.Delete();break;
    case '3':s1.Find();break;
    case '4':s1.Change();break;
    case '5':s1.Display();break;
    default:cout<<"抱歉, 您想要进行的操作不存在, 请重新输入! "<<endl;break;
}
}
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>>operate_code;
}
```

### 3.6.3 总体功能截屏示例

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 male 21 software

2 stu2 female 22 test

考号	姓名	性别	年龄	报考类别
1	stu1	male	21	software
2	stu2	female	22	test

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 错误测试

---

#### 4.1.1 考生人数错误

测试用例：输入考生人数为零、负数甚至字母

预期结果：程序给出提示信息，程序正常运行不崩溃

实验结果：

请输入考生人数：

0

输入的人数没有意义！请重新输入：

-1

输入的人数没有意义！请重新输入：

a

输入的人数没有意义！请重新输入：

.

#### 4.1.2 输入操作码错误

测试用例：输入操作码不存在甚至字母

预期结果：程序给出提示信息，程序正常运行不崩溃

实验结果：

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）

9

抱歉，您想要进行的操作不存在，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）

a

抱歉，您想要进行的操作不存在，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）

|

### 4.1.3 插入位置不存在

测试用例：给出不存在的插入位置

预期结果：程序给出提示信息，程序正常运行不崩溃

实验结果：

考号	姓名	性别	年龄	报考类别
1	stu1	male	20	software
3	stu3	male	20	network
2	stu2	female	21	test

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位置：6

无效的插入位置！请重新输入：

---

### 4.1.4 在空表中删除

测试用例：当前表为空，但要求执行删除操作

预期结果：程序给出提示信息，程序正常运行不崩溃

实验结果：

删除后的表为：

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）

2

这个表已经没有可以删除的项了

#### 4.1.5 删除学号不存在

测试用例：给出一个不存在的学号要求执行删除操作

预期结果：程序给出提示信息，程序正常运行不崩溃

实验结果：

**请输入你要删除的考生考号：3**

**未找到删除学号，故本次没有删除任何数据**

**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）**

#### 4.1.6 查找表为空

测试用例：当前表为空，但要求执行查找操作

预期结果：程序给出提示信息，程序正常运行不崩溃

实验结果：

**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）**

**3**

**你要在空表里找什么呢？**

#### 4.1.7 修改考号不存在

测试用例：修改的考号不存在，但要求执行修改操作

预期结果：程序给出提示信息，程序正常运行不崩溃

实验结果：

**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）**

**4**

**请输入要修改学生的学号：0**

**想要修改的考号不存在**