考试报名系统

作 者 姓 名： 李翠琪

学 号： 1751022

指 导 教 师： 张颖 \_\_

学院、 专业： 软件工程 \_\_

同济大学

Tongji University

目 录

[1 分析 1](#_Toc22218342)

[1.1 背景分析 1](#_Toc22218343)

[1.2 功能分析 1](#_Toc22218344)

[2 设计 2](#_Toc22218345)

[2.1 存储结构设计 2](#_Toc22218346)

[2.2 Student类结构设计 2](#_Toc22218347)

[2.3 主程序设计 2](#_Toc22218348)

[3 实现 3](#_Toc22218349)

[3.1 插入功能的实现 3](#_Toc22218350)

[3.1.1 表头插入 3](#_Toc22218351)

[3.1.2 任意位置插入 3](#_Toc22218352)

[3.2 删除功能的实现 4](#_Toc22218353)

[3.2.1 删除功能描述 4](#_Toc22218354)

[3.2.2 删除功能代码 4](#_Toc22218355)

[3.3 查找功能的实现 5](#_Toc22218356)

[3.3.1 查找功能描述 5](#_Toc22218357)

[3.3.2 查找功能代码 5](#_Toc22218358)

[3.4 修改功能的实现 5](#_Toc22218359)

[3.4.1 修改功能描述 5](#_Toc22218360)

[3.4.2 修改功能代码 6](#_Toc22218361)

[3.5 统计功能的实现 7](#_Toc22218362)

[3.6 总体系统的实现 7](#_Toc22218363)

[3.6.1 总体系统描述 7](#_Toc22218364)

[3.6.2 总体系统核心代码 7](#_Toc22218365)

[4 测试 9](#_Toc22218366)

[4.1 功能测试 9](#_Toc22218367)

[4.1.1 建立功能测试 9](#_Toc22218368)

[4.1.2 插入功能测试 10](#_Toc22218369)

[4.1.3 删除功能测试 11](#_Toc22218370)

[4.1.4 查找功能测试 11](#_Toc22218371)

[4.1.5 修改功能测试 12](#_Toc22218372)

[4.1.6 统计功能测试 12](#_Toc22218373)

[4.2 边界测试 13](#_Toc22218374)

[4.2.1 初始化无输入数据 13](#_Toc22218375)

[4.2.2 删除头结点 14](#_Toc22218376)

[4.2.3 删除后链表为空 15](#_Toc22218377)

[4.2.4 考生人数错误 16](#_Toc22218378)

[4.2.5 位置不正确 16](#_Toc22218379)

[4.2.6 id不存在 17](#_Toc22218380)

# 1 分析

## 1.1 背景分析

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。

## 1.2 功能分析

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。项目在设计时应首先确定系统的数据结构，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。采用链表实现。

综上,至少有建立，查找，插入，修改，删除,退出功能。

# 2 设计

## 2.1 存储结构设计

因为在建立学生信息表之后还需要进行大量添加,删除操作,所以采用链表存储.用模板类Node存储学生信息,模板类LinkedList存储一条由Node构成的链表,即存储了所有学生信息,并且能进行相应的操作.因为无其他要求,所以采用单向链表,且有一个头指针first.

LinkedList中,通过add(),pop(),change(),show()等方法,来对链表进行对应的操作,即实现了对学生信息表的增删改查功能.

## 2.2 Student类结构设计

Student类代表了一个学生的实体,含有该学生所需要的成员变量:准考证号，姓名，性别，年龄和报考类别.这些成员变量设为private以达到封装的效果,所以也提供了相应的get(),set()方法来对这些属性进行修改.并且有一个展示该学生信息的show()方法.

## 2.3 主程序设计

开始时建立LinkedList的对象students,输出窗口将提示引导用户逐步建立学生信息表,然后展示所建立表.随后进入循环,直到用户按0退出程序.循环中,根据用户输入,调用students对象相应的函数,执行相应的操作, 完成插入删除，查找，修改，统计功能.

# 3 实现

## 3.1 插入功能的实现

### 3.1.1 表头插入

在建立表时,就采用了简单的在表头插入,即每一个学生在插入时都处于first的位置,调用了Node的构造方法,所以很简洁

void add(T t)  
{  
 Node<T> \*oldFirst = first;  
 first = new Node<T>(t,oldFirst);  
}

### 3.1.2 任意位置插入

提供了在任意位置插入的功能,首先判断传入位置是否合理,然后移到要插入的位置的前面一个,进行插入,并且展示插入后的学生表.

void insert(int pos, T t)  
{  
 if(pos<=0) return;  
 //相当于add  
 if(pos==1)  
 {  
 add(t);  
 }  
 else  
 {  
 Node<T> \*cur = first;  
 for (int i = 0; i < pos - 2; ++i)  
 {  
 if(cur== nullptr)  
 return;  
 cur = cur->next;  
 }  
 Node<T> \*newNode = new Node<T>(t,cur->next);  
 cur->next = newNode;  
 }  
 show();  
}

## 3.2 删除功能的实现

### 3.2.1 删除功能描述

首先先判断first不为nullptr,否则继续程序可能会报错,然后根据传入的id进行判断,如果是首元素,直接让first指向它的下一个元素,即使只有一个元素的情况下也是正确的.

如果不是首元素,同理,通过检测下一个元素是否为空小心的向后移动,找到下一个元素id为传入id的元素,删除之,展示删除之后的学生表.

### 3.2.2 删除功能代码

void pop(const string & id)  
{  
 if(first== nullptr)  
 return;  
 if(id==first->value.getId())//相当于add  
 {  
 first = first->next;  
 }  
 else  
 {  
 for (Node<T> \* x=first; x!= nullptr ; x=x->next)  
 {  
 if(x->next && x->next->value.getId()==id)  
 {  
 Node<T> \* temp = x->next;  
 x->next = x->next->next;  
 delete temp;  
 }  
 }  
 }  
 show();  
}

## 3.3 查找功能的实现

### 3.3.1 查找功能描述

遍历链表,找到学生表中符合传入id的元素,通过重载了的<<把该学生信息打印出来.

### 3.3.2 查找功能代码

void show(const string & id)  
{  
 cout<<endl<<"学号 姓名 性别 年龄 报考类别"<<endl;//空四格  
 for (Node<T> \* x=first; x!= nullptr ; x=x->next)  
 {  
 if(id==x->value.getId())  
 {  
 cout<<x->value<<endl;  
 return;  
 }  
 }  
 cout << "No such student";  
}

## 3.4 修改功能的实现

### 3.4.1 修改功能描述

修改功能首先根据传入的id找到对应的学生,然后根据传入的pos,找到想要修改的属性,并且把该属性设置为传入的value值,需要注意的是,如果传入的是age的string,需要转换为int.重新展示学生信息表.

### 3.4.2 修改功能代码

void change(const string & id,int pos,string value)  
{  
 //pos 0,1,2,3,4 id,name,gender,age,examCategory  
 for (Node<T> \* x=first; x!= nullptr ; x=x->next)  
 {  
 if(id==x->value.getId())  
 {  
 switch (pos)  
 {  
 case 0:  
 x->value.setId(value);  
 break;  
 case 1:  
 x->value.setName(value);  
 break;  
 case 2:  
 x->value.setGender(value);  
 break;  
 case 3:  
 x->value.setAge(atoi(value.c\_str()));  
 break;  
 case 4:  
 x->value.setExamCategory(value);  
 break;  
 default:  
 break;  
 }  
 }  
 }  
 show();  
}

## 3.5 统计功能的实现

void show()  
{  
 cout<<endl<<"学号 姓名 性别 年龄 报考类别"<<endl;//空四格  
 for (Node<T> \* x=first; x!= nullptr ; x=x->next)  
 {  
 cout<<x->value<<endl;  
 }  
}

## 3.6 总体系统的实现

### 3.6.1 总体系统描述

引导用户输入建立表的总人数N,创建表studentList,并引导用户逐个输入每个学生的每个属性.展示学生表.进入循环,根据用户输入的operate,switch之,并且根据不同的operate调用studentList里相应的函数.

### 3.6.2 总体系统核心代码

bool is\_continue=true;  
while(is\_continue)  
{  
 cout<<"请选择你要进行的操作：";  
 int operate;//操作  
 int pos;//位置  
 string id,name,gender,exam\_category;  
 int age;  
 string value;//修改时指定的属性  
 Student student;  
 cin>>operate;  
 //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  
 switch(operate)  
 {  
 case 1:  
 cout<<"请输入你要插入的考生的位置：";  
 cin>>pos;  
 //插入一个考生信息  
 cout << "请依次输入该考生的信息: ";  
 cin>>id>>name>>gender>>age>>exam\_category;  
 student.setId(id);  
 student.setName(name);  
 student.setGender(gender);  
 student.setAge(age);  
 student.setExamCategory(exam\_category);  
 studentList.insert(pos,student);  
 break;  
 case 2:  
 cout<<"请输入要删除的考生的考号：";  
 //注意是操作考号，所有得是string  
 cin>>id;  
 studentList.pop(id);  
 break;  
 case 3:  
 cout << "请输入要查找的考生的考号：";  
 cin >> id;  
 studentList.show(id);  
 break;  
 case 4:  
 cout << "请输入要修改的考生的考号：";  
 cin >> id;  
 cout << "请输入要修改的位置：";  
 cin >> pos;  
 cout << "请输入要修改的值：";  
 cin >> value;  
 studentList.change(id,pos,value);  
 break;  
 case 5:  
 studentList.show();  
 break;  
 case 0:  
 is\_continue=false;  
 break;  
 default:  
 break;  
 }  
}

# 4 测试

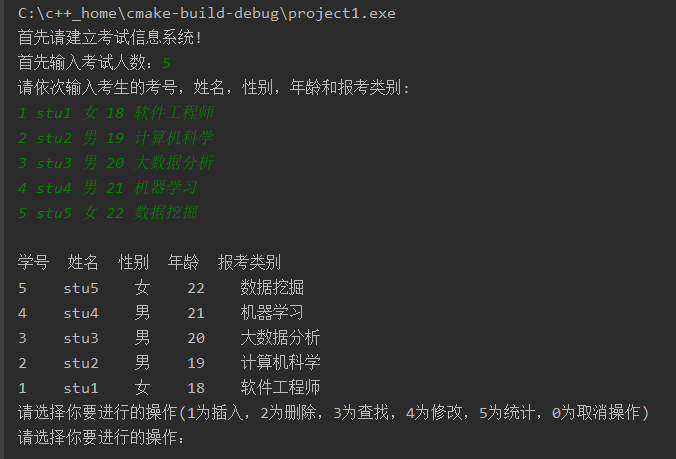
## 4.1 功能测试

### 4.1.1 建立功能测试

**测试用例**：

5  
1 stu1 女 18 软件工程师  
2 stu2 男 19 计算机科学  
3 stu3 男 20 大数据分析  
4 stu4 男 21 机器学习  
5 stu5 女 22 数据挖掘

**实验结果**

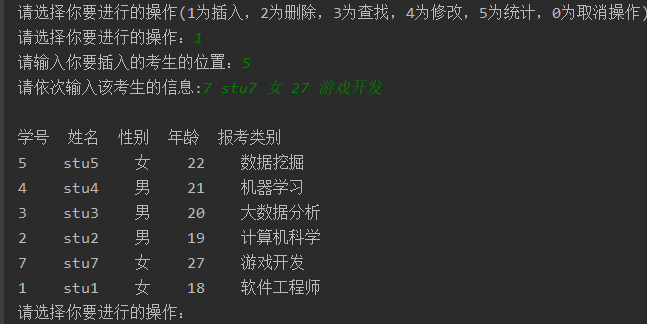


### 4.1.2 插入功能测试

**测试用例：**

1  
5  
7 stu7 女 27 游戏开发

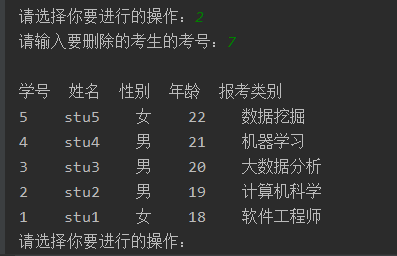
**实验结果：**



### 4.1.3 删除功能测试

**测试用例：**删除考号为7的考生

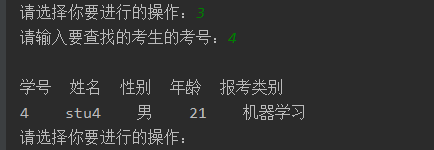
**实验结果：**



### 4.1.4 查找功能测试

**测试用例：**查找考号为4的考生

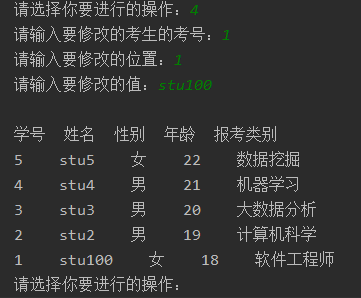
**实验结果：**



### 4.1.5 修改功能测试

**测试用例：**修改考号为1的学生的姓名为stu100

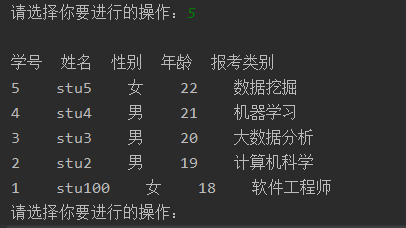
**实验结果：**



### 4.1.6 统计功能测试

**测试用例：**统计所有学生信息

**实验结果：**



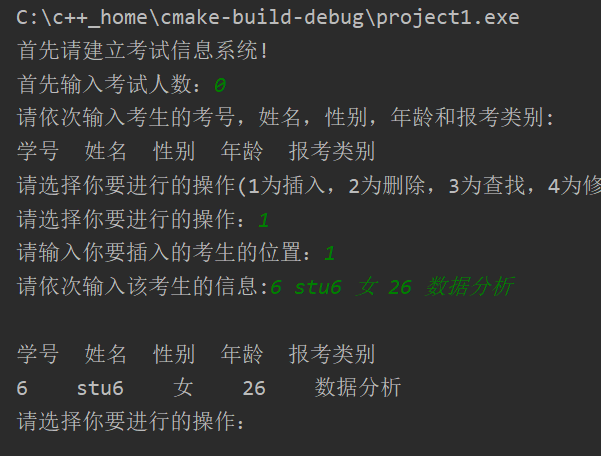
## 4.2 边界测试

### 4.2.1 初始化无输入数据

**测试用例：**初始无输入数据

预期结果:程序不报错,且进入循环中

**实验结果：**

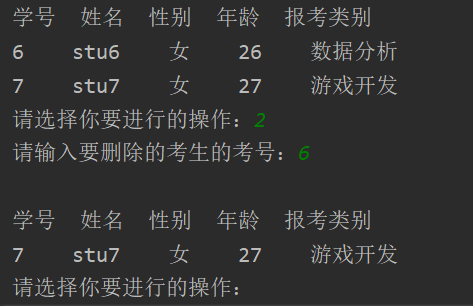


### 4.2.2 删除头结点

**测试用例：**删除头结点

**预期结果：**程序正常运行，不崩溃。

**实验结果：**

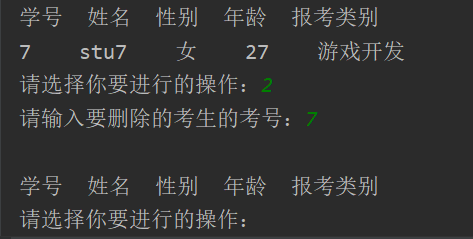


### 4.2.3 删除后链表为空

**测试用例：**删除前链表只有一个结点，删除后链表为空

**预期结果：**程序正常运行，不崩溃。

**实验结果：**

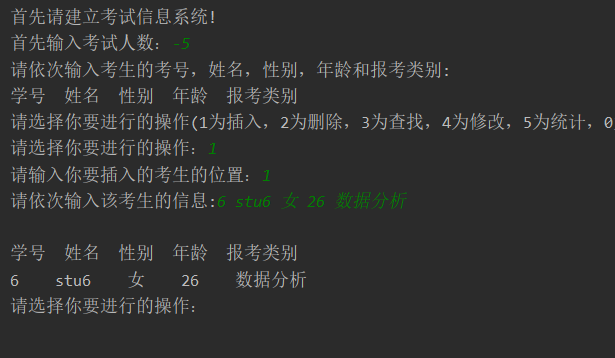


### 4.2.4 考生人数错误

**测试用例：**输入考生人数为负数

**预期结果：**程序正常运行不崩溃。

**实验结果：**

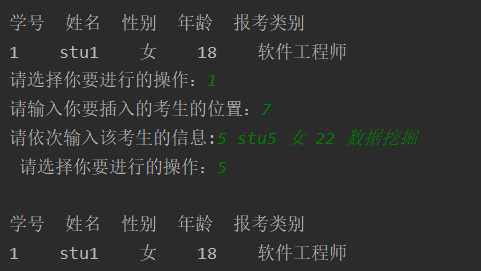


### 4.2.5 位置不正确

**测试用例：**在不正确的位置进行插入

**预期结果：**程序正常运行不崩溃。

**实验结果**



### 4.2.6 id不存在

**测试用例：**删除不存在的id

**预期结果：**程序正常运行不崩溃。

**实验结果**

