项目说明文档

数据结构课程设计

——考试报名系统

作 者 姓 名： 李圣

学 号： 1952723

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji Universit

目录

[1 分析 1](#_Toc10145)

[1.1 背景分析 1](#_Toc22250)

[1.2 功能分析 1](#_Toc14677)

[2 设计 1](#_Toc18861)

[2.1 文件结构设计 1](#_Toc6280)

[2.2 数据结构设计 1](#_Toc30784)

[2.3 类结构设计 1](#_Toc20588)

[2.4 成员与函数设计 2](#_Toc3421)

[3 实现 5](#_Toc17375)

[3.1 插入功能的实现 5](#_Toc24394)

[3.1.1 插入功能流程图 5](#_Toc31590)

[3.1.2 插入功能核心代码 6](#_Toc6763)

[3.1.3 插入功能示例 6](#_Toc26625)

[3.2 删除功能的实现 7](#_Toc12611)

[3.2.1 删除功能流程图 7](#_Toc27038)

[3.2.2 删除功能核心代码 8](#_Toc17815)

[3.2.3 删除功能示例 8](#_Toc7740)

[3.3 查找功能的实现 10](#_Toc7057)

[3.3.1 查找功能流程图 10](#_Toc26356)

[3.3.2 查找功能核心代码 11](#_Toc28438)

[3.3.3 查找功能示例 11](#_Toc14981)

[3.4 修改功能的实现 13](#_Toc1192)

[3.4.1 修改功能流程图 13](#_Toc22813)

[3.4.2 修改功能核心代码 14](#_Toc23186)

[3.4.3 修改功能示例 14](#_Toc21353)

[3.5 统计功能的实现 16](#_Toc17452)

[3.5.1 统计功能流程图 16](#_Toc10045)

[3.5.2 统计功能核心代码 17](#_Toc18547)

[3.5.3 统计功能示例 17](#_Toc1607)

[3.6 总体系统的实现 18](#_Toc9813)

[3.6.1 总体系统流程图 18](#_Toc16311)

[3.6.2 总体系统核心代码 19](#_Toc13405)

[3.6.3 总体系统示例 20](#_Toc19389)

[4 测试 20](#_Toc16024)

[4.1 鲁棒性测试 20](#_Toc12355)

[4.1.1 正常案例测试 20](#_Toc23801)

[4.1.2 初始化测试 22](#_Toc28768)

[4.1.3 插入测试 22](#_Toc19844)

[4.1.4 删除测试 23](#_Toc1090)

[4.1.5 查找测试 24](#_Toc30969)

[4.1.6 修改测试 24](#_Toc1456)

[4.1.7 指令测试 25](#_Toc15850)

# 分析

## 背景分析

考试报名系统是一个学校不可缺少的部分，拥有着大量的需求。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能够极大地提高学校和学生的效率。开发一套考试报名系统十分重要。

此外，考虑到对Linux的需求以及自身对Linux学习的需要，因此在电脑上装了Linux子系统并通过vscode对其进行调试。

## 功能分析

考试报名系统至少应该具有的基本功能：

1. 输入
2. 输出
3. 插入
4. 删除
5. 修改
6. 退出

# 设计

## 文件结构设计

考虑到该项目所需文件不多，因此只包含一个文件，如下：

main.cpp

## 数据结构设计

显然，该系统要求大量的增加、删除、修改操作，因此数据结构考虑使用带表头的单向链表。

## 类结构设计

main中总共包含四个类：Student、Node、List、Guide。

Student类提供学生数据存储以及输入输出相关接口。

Node类为List节点类，提供List操作相关接口。

List为单向带头节点链表类，为本项目主要类，提供插入删除等主要功能。

Guide为指导类，主要负责可视化，全局控制以及输入输出提示。

## 成员与函数设计

**Student类:**

**保护变量(附带set与get函数):**

string \_id;

string \_name;

string \_gender;

string \_age;

string \_subject;

**公有变量:**

static const Student example1;

static const Student example2;

static const Student example3;

**公有函数:**

Student();

Student(string id, string name, string gender, string age, string subject);

Student(const Student &st);

~Student();

friend istream &operator>>(istream &is, Student &st);

friend ostream &operator<<(ostream &os, const Student &st);

**Node类:**

**保护变量(附带set与get函数):**

Student \_data;

Node\* \_next;

**公有函数:**

Node();

Node(const Student &st)

void deleteAll();

**List类:**

**私有变量：**

Node \*\_head;

int \_length;

**公有变量:**

static const string title[5];

**公有函数:**

static void outputTitle()；

List();

~List();

void initialize();

void clear();

int length();

bool input(int num);

void insert(int pos, const Student &st);

bool remove(const string &id);

bool find(const string &id);

bool change(const string &id, const Student &st);

void showAll();

**Guide类:**

**私有变量：**

List \_list;

bool \_isBuilt;

**公有变量:**

static const string toInput;

static const string orderGuide;

**公有函数:**

Guide();

void insert();

void remove();

void find();

void change();

bool runOrder(string order);

bool getOrder();

bool initialize();

bool loop();

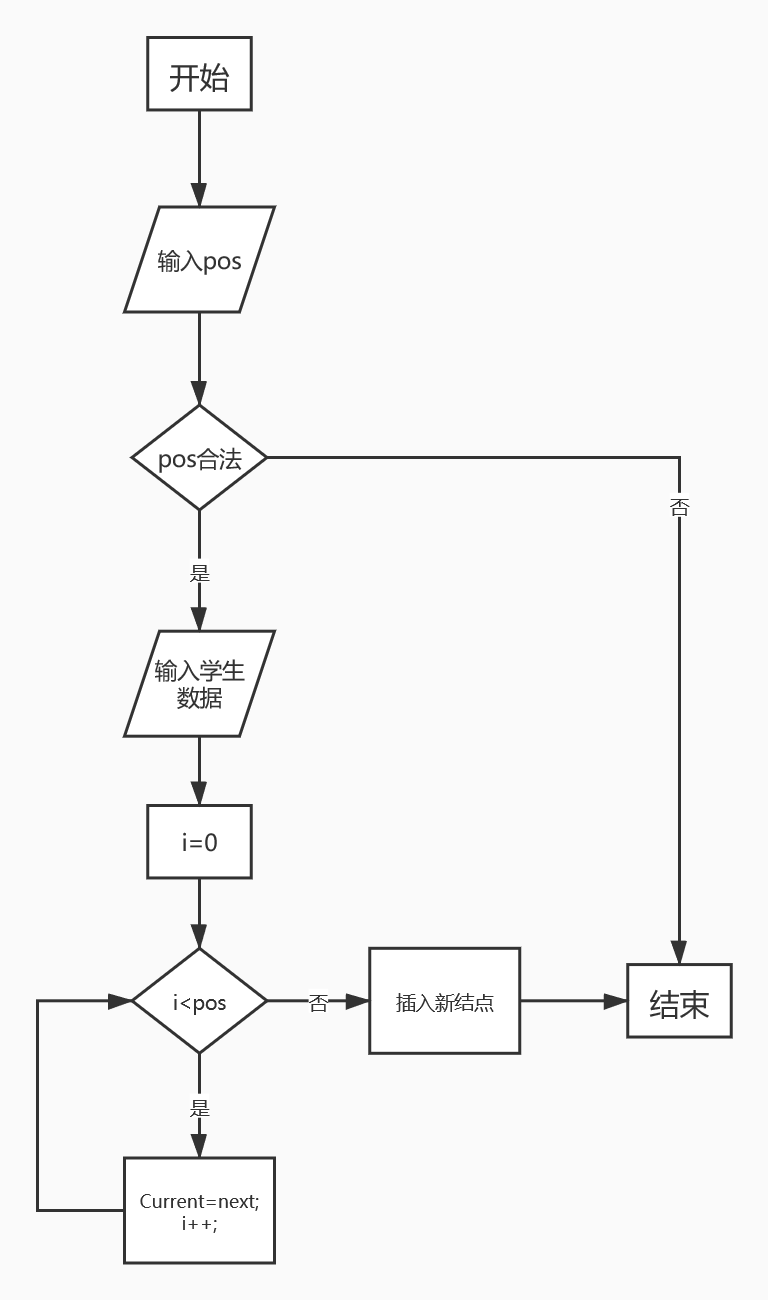
void showAll();

void main();

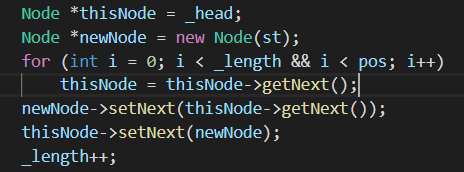
# 实现

## 插入功能的实现

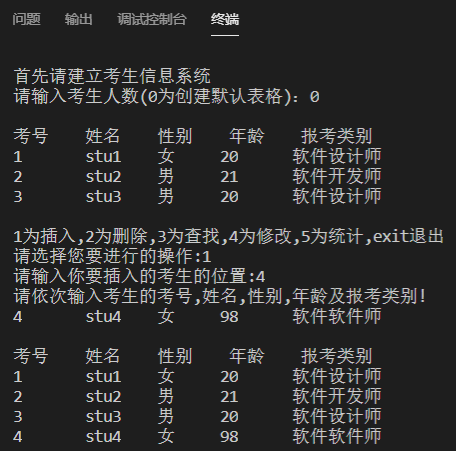
### 插入功能流程图

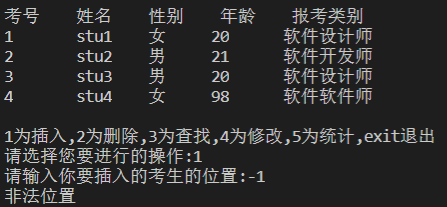


### 插入功能核心代码



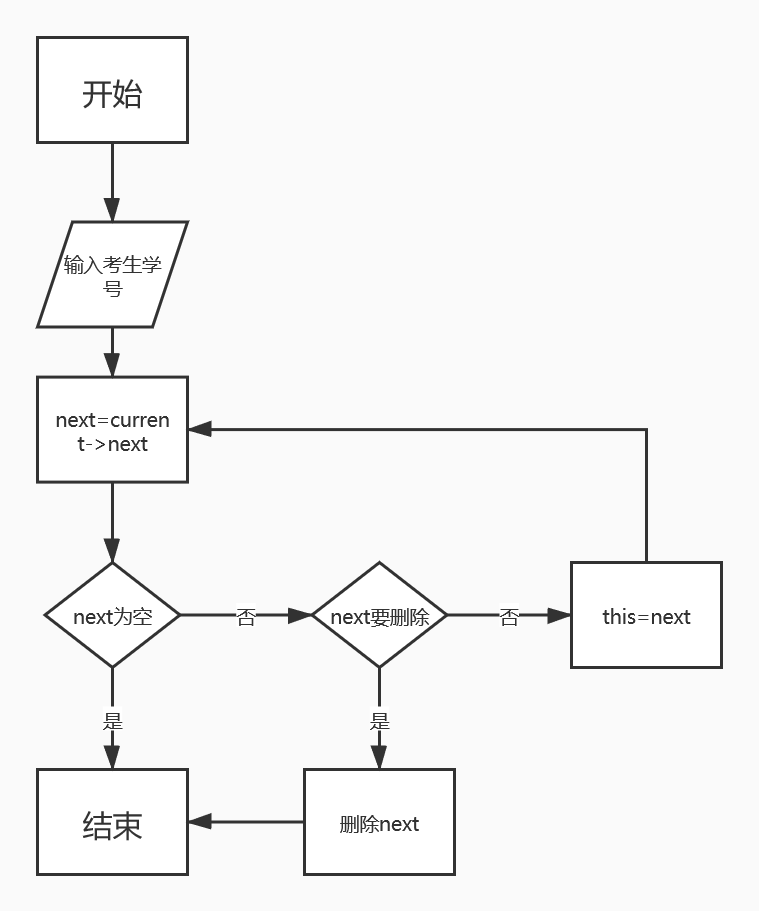
### 插入功能示例



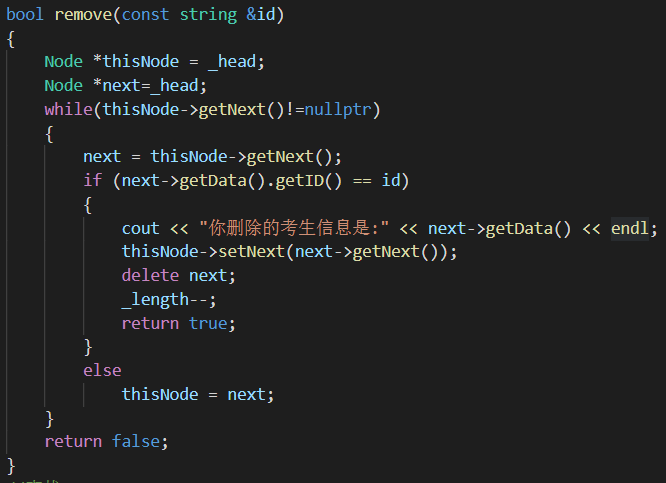


## 删除功能的实现

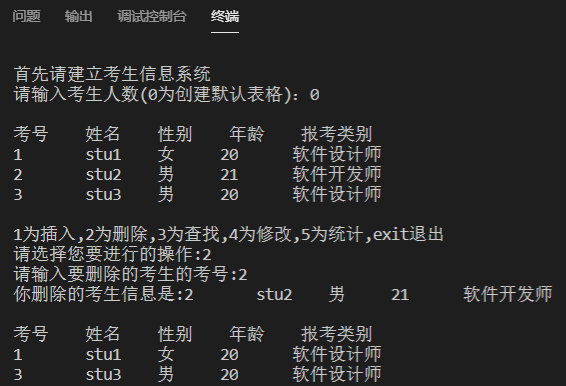
### 删除功能流程图

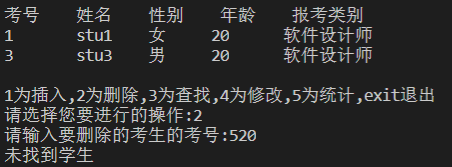


### 删除功能核心代码



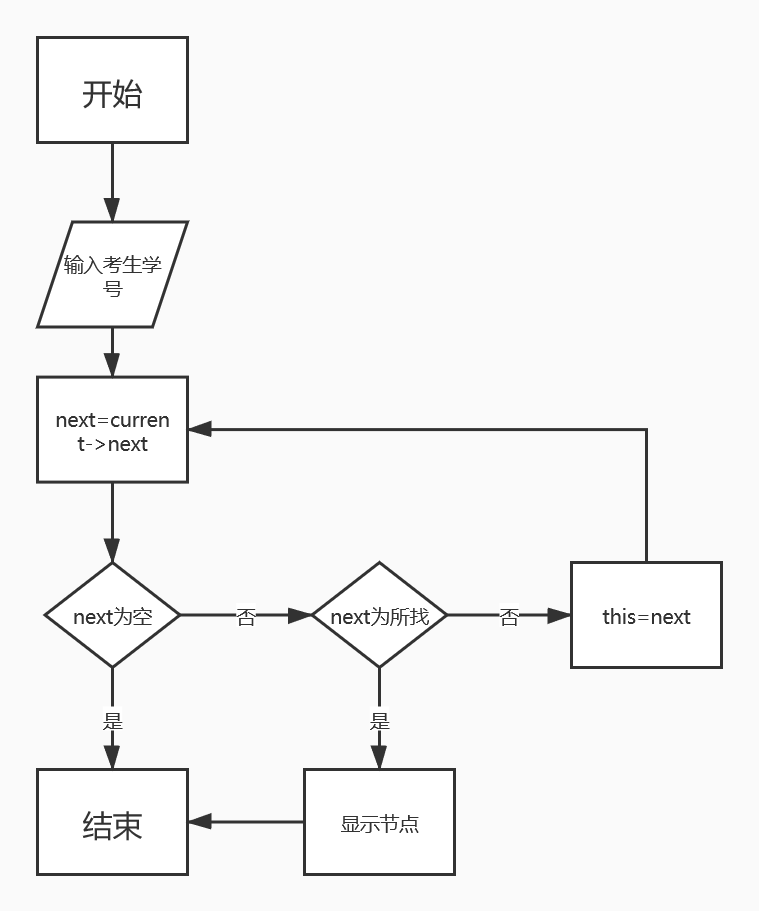
### 删除功能示例



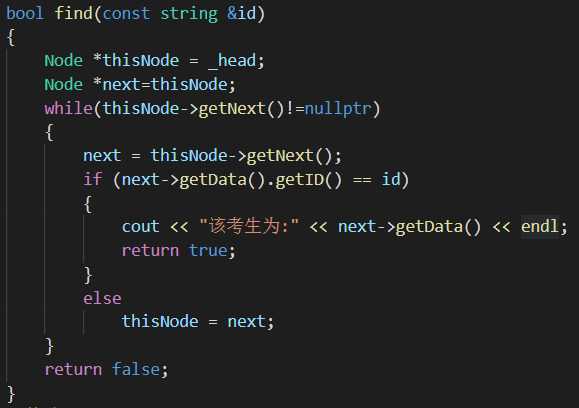


## 查找功能的实现

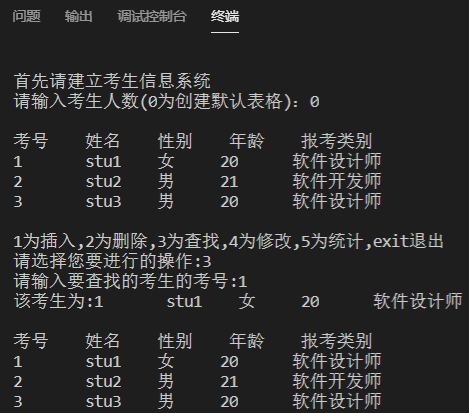
### 查找功能流程图

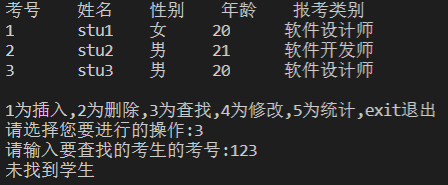


### 查找功能核心代码



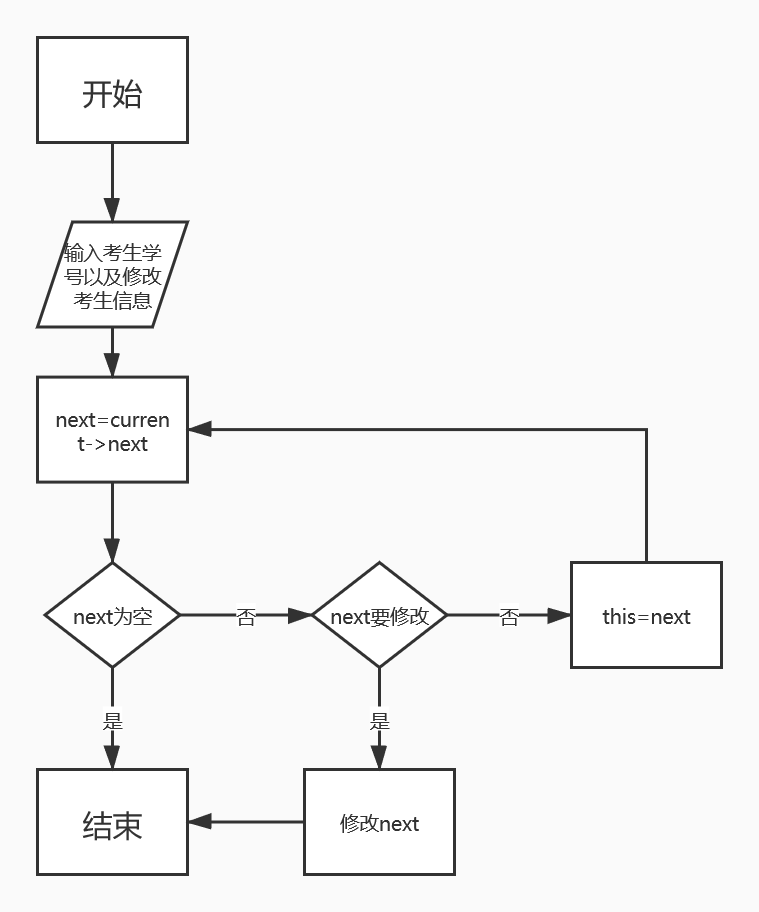
### 查找功能示例



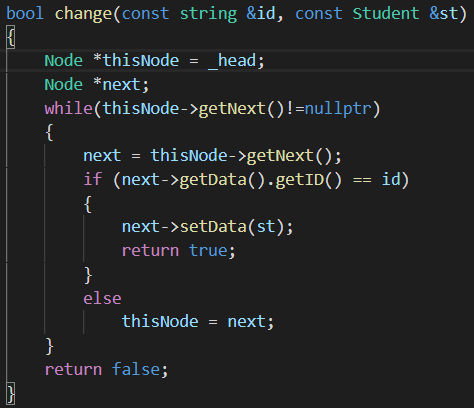


## 修改功能的实现

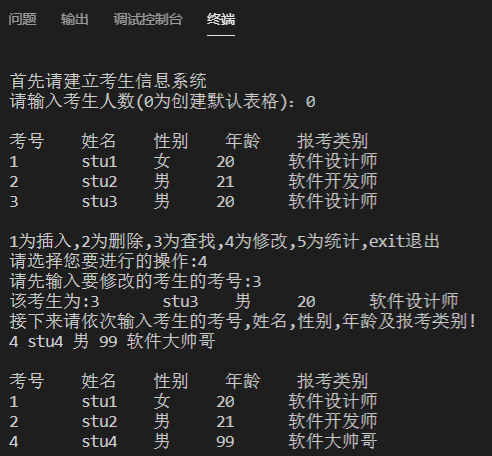
### 修改功能流程图

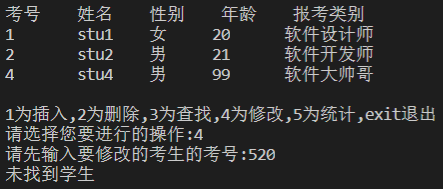


### 修改功能核心代码



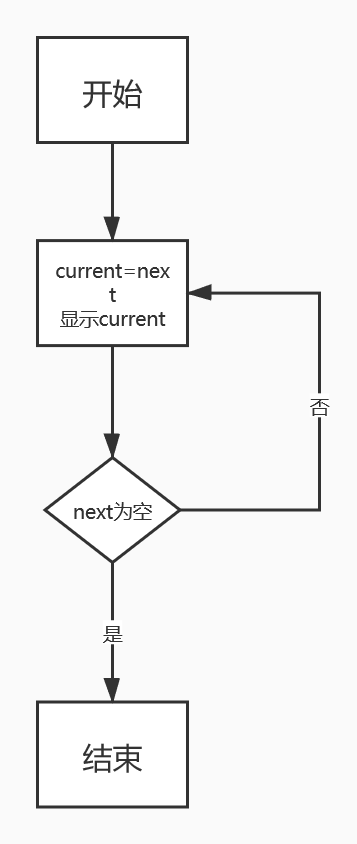
### 修改功能示例



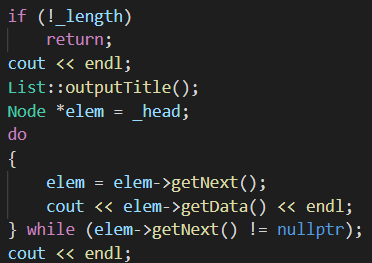


## 统计功能的实现

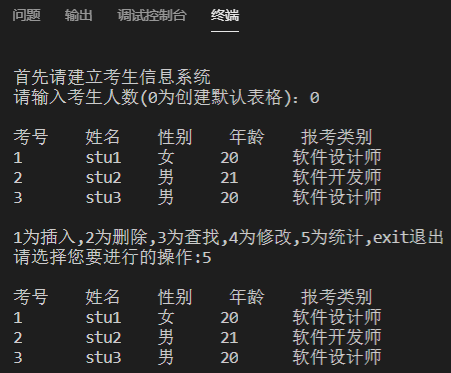
### 统计功能流程图



### 统计功能核心代码

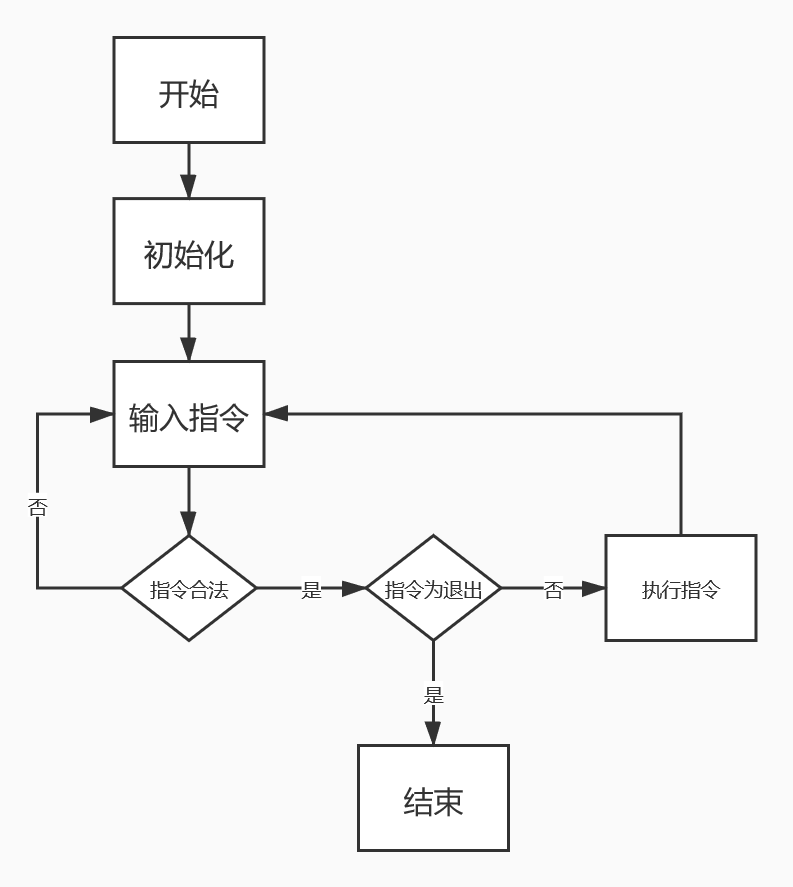


### 统计功能示例

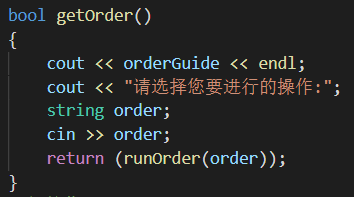


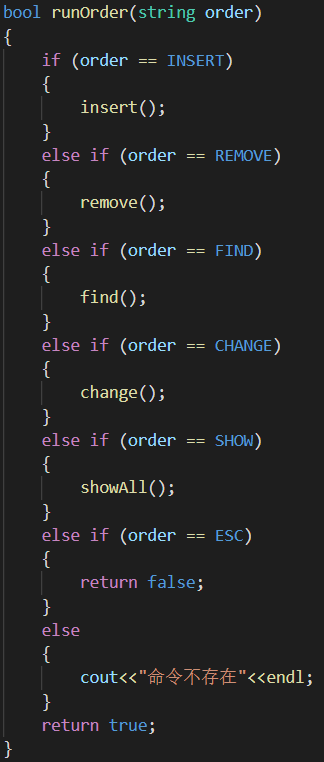
## 总体系统的实现

### 总体系统流程图

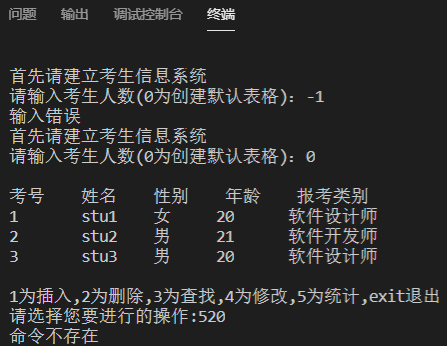


### 总体系统核心代码





### 总体系统示例



# 测试

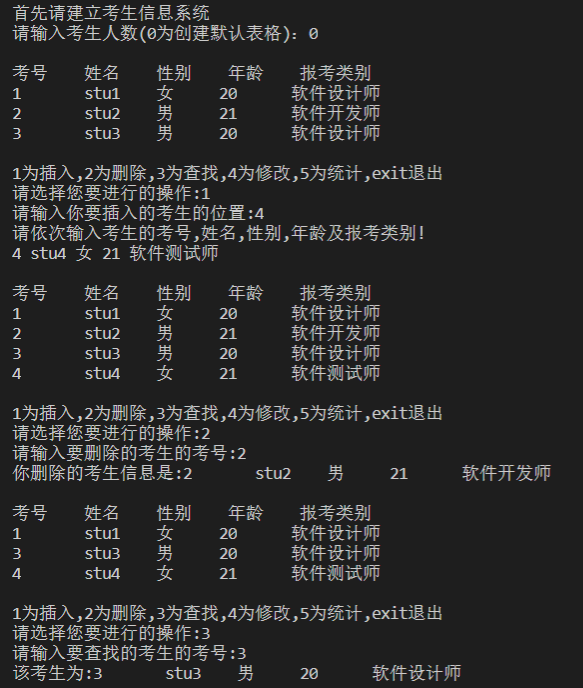
## 鲁棒性测试

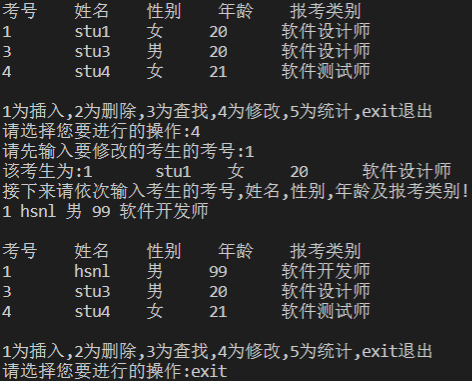
### 正常案例测试

**测试用例：**正常的合法案例

**预期结果：**输出对应的结果

**测试结果：**



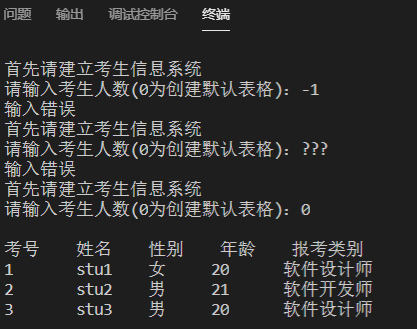


### 初始化测试

**测试用例：**初始化时输入非法

**预期结果：**程序继续运行

**测试结果：**

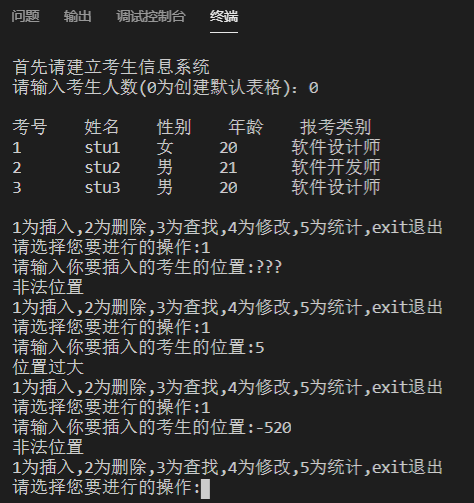


### 插入测试

**测试用例：**插入时输入非法

**预期结果：**程序继续运行

**测试结果：**

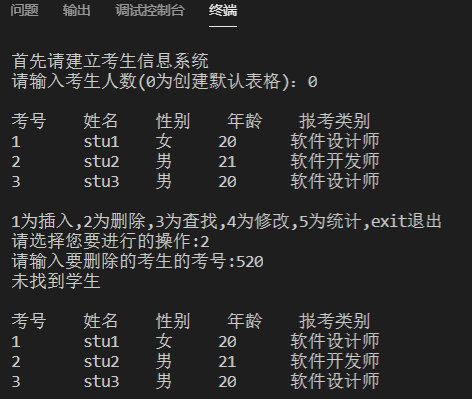


### 删除测试

**测试用例：**不存在的学号

**预期结果：**不会进行删除，程序继续运行

**测试结果：**

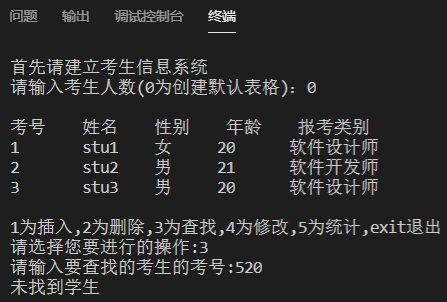


### 查找测试

**测试用例：**不存在的学号

**预期结果：**提示未找到，程序继续运行

**测试结果：**

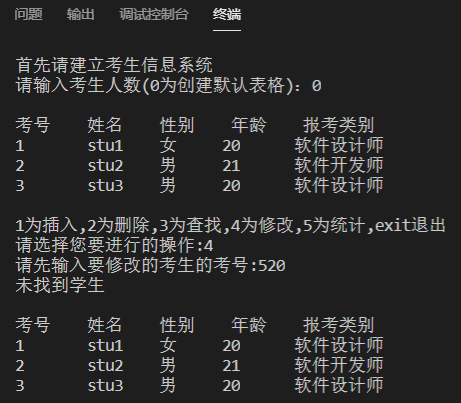


### 修改测试

**测试用例：**不存在的学号

**预期结果：**提示未找到，程序继续运行

**测试结果：**



### 指令测试

**测试用例：**不存在的指令

**预期结果：**提示指令不存在，程序继续运行

**测试结果：**

