项目说明文档

数据结构课程设计

——考试报名系统

作 者 姓 名： 李佳诺

学 号： 1751188

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 3

1.1背景介绍 3

1.2 功能分析 4

2 设计 4

2.1 数据结构设计 4

2.2 类结构设计 5

2.3 成员与操作设计 5

*链表结点（student）* 5

*链表类（StudentInformation）* 5

3 实现 7

3.1 构造函数StudentInformation（）: 7

构造函数流程图 7

3.2 插入函数insertData（）： 9

插入函数流程图 9

3.3 删除函数deleteData（）： 11

删除功能流程图 11

3.4 修改函数setData（）： 13

修改功能流程图 13

3.5 查找函数getData（）： 15

查找功能流程图 15

3.6 打印函数print（）： 17

3.7 主函数（运行函数）main（）： 18

主函数流程图 18

4 测试 20

4.1功能性测试 20

4.1.1 考试系统初始化功能测试： 20

4.1.2 插入功能测试： 21

4.1.3 删除功能测试： 21

4.1.4 查找功能测试： 22

4.1.5 修改功能测试： 22

4.1.6 统计功能测试： 23

4.2 健壮性测试： 24

4.2.1 运行函数初始化健壮性测试： 24

4.2.2 插入功能健壮性测试： 24

4.2.3 删除功能健壮性测试： 25

4.2.4 查找功能健壮性测试： 25

4.2.5 修改功能健壮性测试： 25

# 1 分析

## 1.1背景介绍

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，所以一个良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

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

如上功能分析所述，该系统要求大量的增加、删除、修改操作，而链表进行增加、删除等操作十分简便，因此考虑使用链表数据结构。

## 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。为方便处理，本系统采用struct描述链表结点类（student），这样使得链表类（StudentInformation）可以访问链表结点。

## 2.3 成员与操作设计

##### 链表结点（student）

一个结点即存储了一位考生的所有信息。

//定义学生结点

struct student

{ int number; //考号

string name; //姓名

string gender; //性别

int age; //年龄

string type; //报考类别

student\* next; }; //结点指针域

##### 链表类（StudentInformation）

整个考试管理系统的核心，以链表结构存储了所有考生的信息。这个类里包含了初始化、插入、删除、查询、打印以及系统运行的函数。

//建立考生信息链表

class StudentInformation

{

private:

student\* first; //表头

student\* last; //表尾

public:

StudentInformation(); //构造函数

void print() const; //输出提示指令

void insertData(); //增添数据

void deleteData(); //删除数据

void setData(); //修改数据

void getData(); //获得数据

};

# 3 实现

## 3.1 构造函数StudentInformation（）:

#### 构造函数流程图

开 始

输入考生人数

输入是否合法？

录入各考生信息

打印考生信息

结 束

否

是

构造函数的主要功能是构建考试报名系统，引导用户输入学生信息，完善链表结构，从而使得建立起安全可靠的考试报名系统，具体代码如下：

//初始化考试报名系统

StudentInformation::StudentInformation()

{

first = nullptr;

cout << "首先建立考生系统" << endl << "请输入考生人数：";

int studentNum; //考生人数

cin >> studentNum;

while (studentNum < 0) //若考生人数小于1则报错，并提示重新输入

{

cout << "输入错误，请输入大于等于1的数! " << endl;

cin >> studentNum;

}

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

first = new student;

cin >> first->number >> first->name >> first->gender >> first->age >> first->type;

this->last = first;

for (int i = 1; i < studentNum; i++)

{

last->next = new student;

last = last->next;

last->next = nullptr;

cin >> last->number >> last->name >> last->gender >> last->age >> last->type;

}

last->next = nullptr;

print();

}

## 3.2 插入函数insertData（）：

#### 插入函数流程图

开 始

结 束

输入插入考生位置

是否合法

输入插入考生信息

否

是

插入函数其基本功能是在用户输入的位置插入新的考生的报名信息，以此来完善考试报名系统，具体代码如下：

//插入数据

void StudentInformation::insertData()

{

cout << "请输入您想要插入的考生位置：" << endl;

int position;

cin >> position;

while (position < 1) //如果输入的数小于1则报错，并提示用户重新输入

{

cout << "输入错误，请输入大于等于1的数！" << endl;

cin >> position;

}

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" << endl;

auto newNode = new student;

newNode->next = nullptr;

cin >> newNode->number >> newNode->name>>newNode->gender >> newNode->age >> newNode->type;

auto p = first;

for (int i = 0; i < position - 1; i++)

{

if (p->next != nullptr)

{

p = p->next;

}

}

if (p->next == nullptr)

{

p->next = newNode;

this->last = newNode;

}

else

{

newNode->next = p->next;

p->next = newNode;

}

}

## 3.3 删除函数deleteData（）：

#### 删除功能流程图

开 始

查 找

是否找到

输入删除考生考号

结 束

输出报错

删 除

是

否

删除函数的基本功能是删除用户输入考号的考生信息，其实现的基本逻辑是在遍历链表的时候，检查当前结点的下一个，是否为所要删除的结点，如果是就将下一个结点从整个考生报名链表中摘除。当遍历完成仍未找到，则返回错误信息，具体代码如下：

//删除数据

void StudentInformation::deleteData()

{

cout << "请输入要删除的考生的考号：" << endl;

int num;

cin >> num;

//头节点即为所要删除结点

if (this->first->number == num)

{

auto temp = first;

first = first->next;

delete temp;

return;

}

//删除结点非头节点的情况

auto p = first;

while (p->next!=nullptr)

{

if (p->next->number == num)

{

auto s = p->next;

p->next = s->next;

delete s;

return;

}

p = p->next;

}

//如果遍历完仍未找到要删除的学生，报错

if (p->next == nullptr)

{

cout << "找不到学生的信息" << endl;

}

}

## 3.4 修改函数setData（）：

#### 修改功能流程图

是否找到

开 始

输入修改的考生号

查 找

结 束

修 改

输出改后信息

输出报错

否

是

修改函数的基本功能是修改用户输入考号的考生信息，其实现的基本逻辑是在遍历链表的时候，检查当前结点是否为所要修改的结点，若果是则按照用户输入信息修改。当遍历完成仍未找到，则返回错误信息，具体代码如下：

//修改信息

void StudentInformation::setData()

{

cout << "请输入要修改的学生的考号：" << endl;

int number;

cin >> number;

auto p = first;

while (p != nullptr)

{

if (p->number == number)

{

cout << "请以此输入要修改的考生的考号，姓名，性别，年龄及报考的类别：" << endl;

cin >> p->number >> p->name >> p->gender >> p->age >> p->type;

cout << "修改成功！" << endl;

cout << "考号" << "\t";

cout << "姓名" << "\t";

cout << "性别" << "\t";

cout << "年龄" << "\t";

cout << "报考类别" << endl;

cout << p->number << "\t";

cout << p->name << "\t";

cout << p->gender << "\t";

cout << p->age << "\t";

cout << p->type << endl;

break;

}

p = p->next;

}

if (p == nullptr)

{

cout << "找不到学生信息，请重新输入或者退出。" << endl;

}

}

## 3.5 查找函数getData（）：

#### 查找功能流程图

开 始

是否找到

输入修改的考生号

查 找

结 束

输出改后信息

输出报错

否

是

查找函数的基本功能是在考试报名系统中查询用户所输入考号的考生信息。其实现的基本逻辑是，在遍历链表的时候，检查当前结点是否为所要查询的考生结点，如果是则退出遍历并打印该考生的所有信息，如果找不到则输出提示信息并返回上一级，具体代码如下：

//查找个人信息

void StudentInformation::getData()

{

cout << "请输入要查找的考号:" << endl;

int number;

cin >> number;

auto p = first;

while (p != nullptr)

{

if (p->number == number)

{

cout << "考号" << "\t";

cout << "姓名" << "\t";

cout << "性别" << "\t";

cout << "年龄" << "\t";

cout << "报考类别" << endl;

cout << p->number << "\t";

cout << p->name << "\t";

cout << p->gender << "\t";

cout << p->age << "\t";

cout << p->type << endl;

break;

}

p = p->next;

}

if (p == nullptr)

{

cout << "找不到学生信息，请重新输入" << endl;

}

}

## 3.6 打印函数print（）：

打印函数的基本功能就是打印当前结点的考生信息，具体代码如下：

//打印学生信息

void StudentInformation::print() const

{

cout << "考号" << "\t";

cout << "姓名" << "\t";

cout << "性别" << "\t";

cout << "年龄" << "\t";

cout << "报考类别" << endl;

auto p = this->first;

while (p != nullptr)

{

cout << p->number << "\t";

cout << p->name << "\t";

cout << p->gender << "\t";

cout << p->age << "\t";

cout << p->type << endl;

p = p->next;

}

}

## 3.7 主函数（运行函数）main（）：

#### 主函数流程图

开 始

统 计

插 入

查 找

删 除

退 出

修 改

主函数的基本功能就是将各函数通过while循环和switch语句连起来，使得整个考试报名系统构成一个有机的整体，核心代码如下：

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:

data.insertData();

data.print();

break;

case 2:

data.deleteData();

data.print();

break;

case 3:

data.getData();

break;

case 4:

data.setData();

break;

case 5:

data.print();

break;

case 0:

cout << "\n程序已经成功退出!\n " << endl;

}

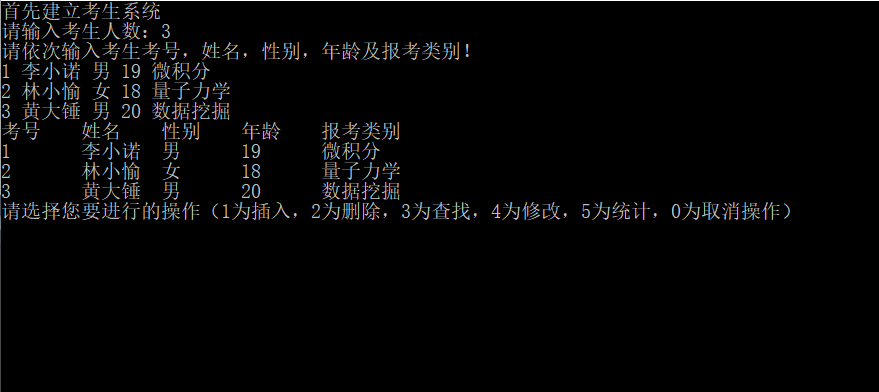
}

# 4 测试

## 4.1功能性测试

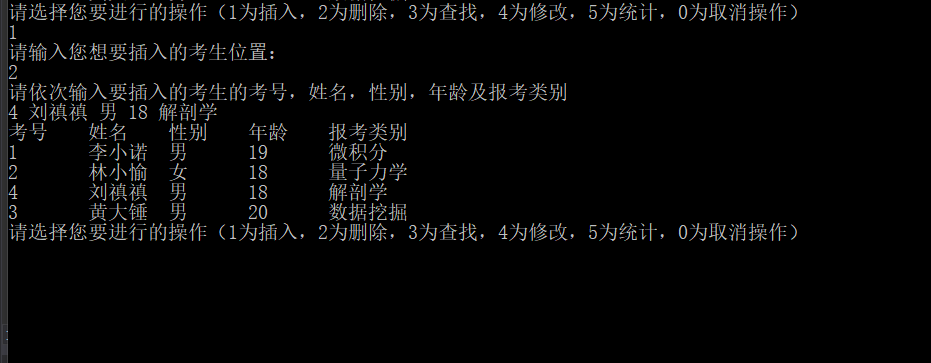
##### 4.1.1 考试系统初始化功能测试：

按提示输入相应的考生信息，进行考试系统初始化。运行结果如下：



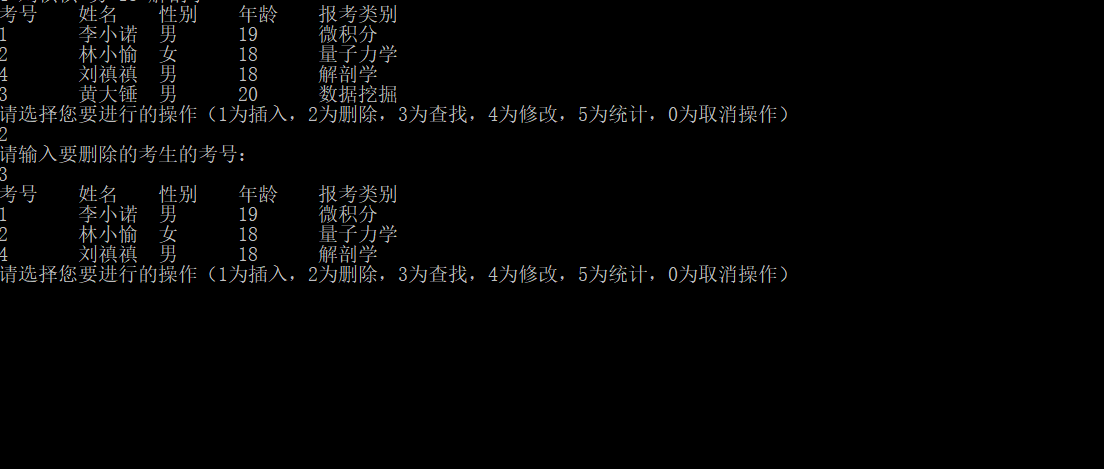
##### 4.1.2 插入功能测试：

选择操作1，然后输入考生的具体位置和信息，最后插入成功。运行结果如下：



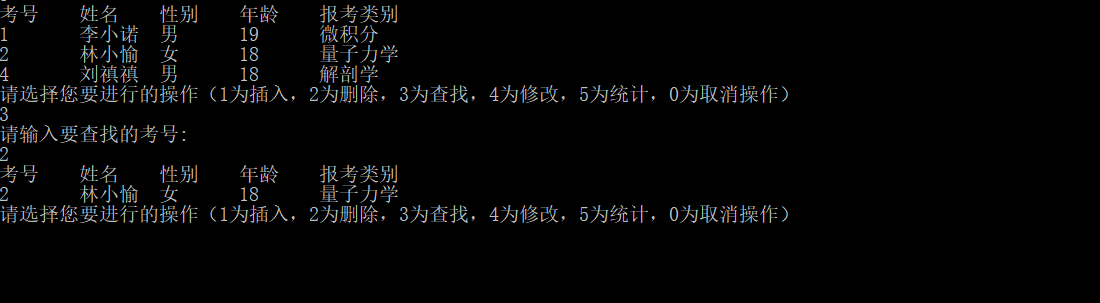
##### 4.1.3 删除功能测试：

选择操作2，然后输入考生的考号，由打印的结构判断删除成功。运行结果如下：



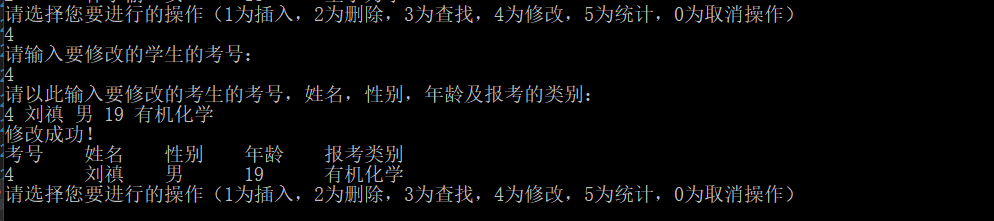
##### 4.1.4 查找功能测试：

选择操作3，然后输入考生的考号，返回该考生的信息。运行结果如下：



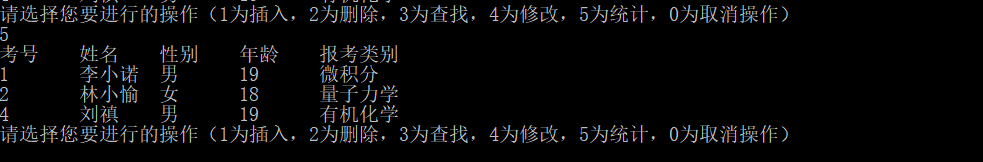
##### 4.1.5 修改功能测试：

选择操作4，然后输入考生的考号以及修改的信息，最终打印出修改后的信息。运行结果如下：



##### 4.1.6 统计功能测试：

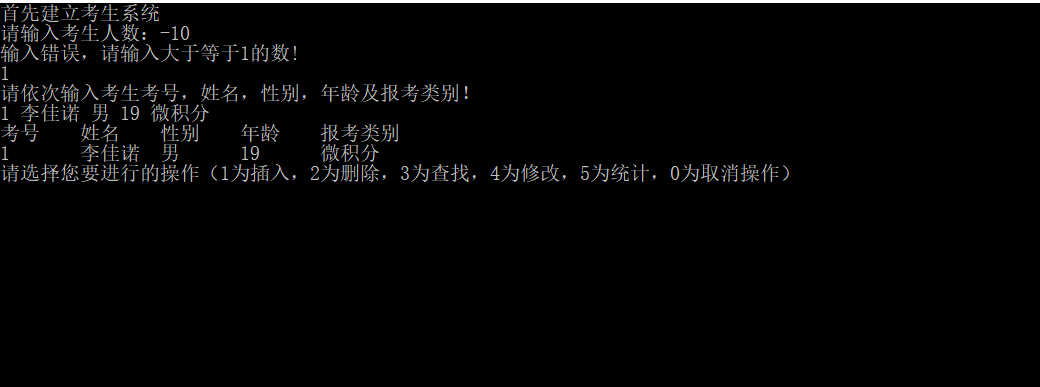
选择操作5，则显示当前考生报名系统的所有考生信息。运行结果如下：



## 4.2 健壮性测试：

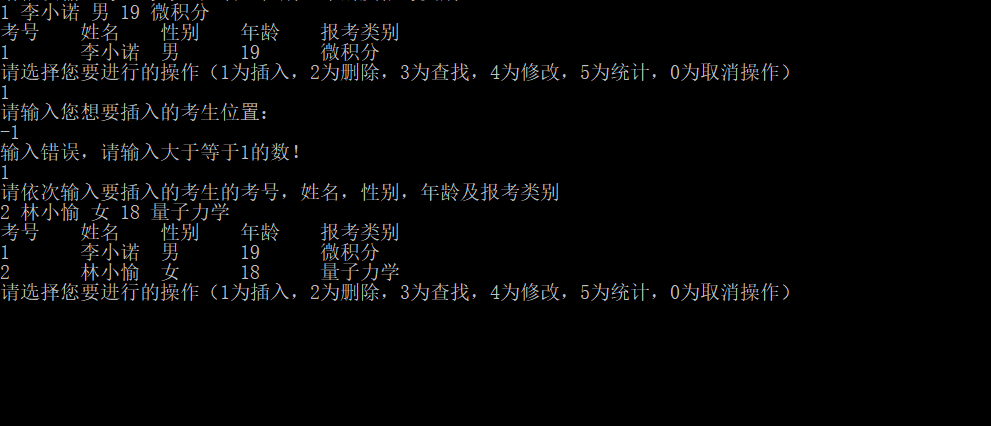
##### 4.2.1 运行函数初始化健壮性测试：

如果初始化时考生的人数小于0，便会报错，并提示用户重新输入考生人数。运行结果如下：



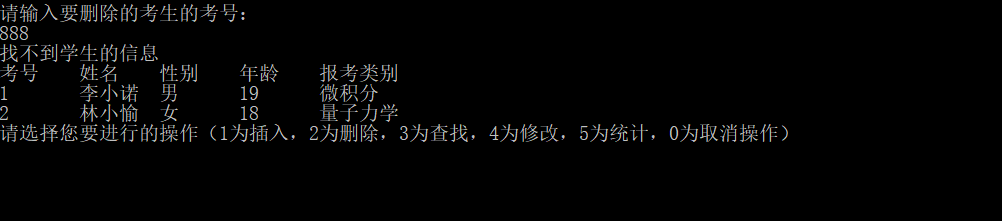
##### 4.2.2 插入功能健壮性测试：

如果输入的插入位置小于1，则报错，并提示用户重新输入。运行结果如下：



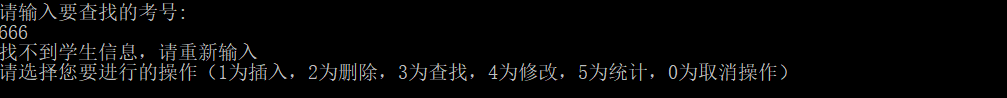
##### 4.2.3 删除功能健壮性测试：

如果删除的考生不存在则会提示错误信息。运行结果如下：



##### 4.2.4 查找功能健壮性测试：

如果所查找的考生不存在则会提示错误信息。运行结果如下：



##### 4.2.5 修改功能健壮性测试：

如果所修改的考生不存在则会提示 错误信息。运行结果如下：

