数据结构课程设计

——考试报名系统

作 者 姓 名： 郑超

学 号： 1751747

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 项目功能分析···············································································································2**

**2 程序设计及实现···········································································································2**

2.1 总述··········································································································································2

2.2 操作引导函数·······················································································································3

2.3 考试报名系统类等数据结构····························································································3

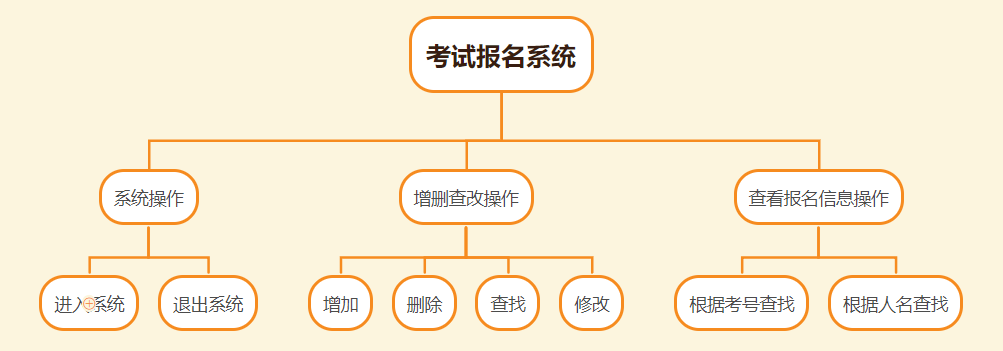
2.4 考试报名系统类相关操作函数························································································4

**3 各功能运行测试···········································································································8**

3.1 测试数据·································································································································8

3.2 测试结果·································································································································8

**1项目功能分析**



考试报名系统功能分析结构图

考试报名系统应实现以下几个基本操作：（一）系统操作；（二）增删查改操作；（三）查看报名信息操作。系统操作包括进、出系统两个部分；用户操作包括增删查改考生信息，以及查看报名情况两个操作类型。

**2 程序设计及实现**

**2.1 总述**

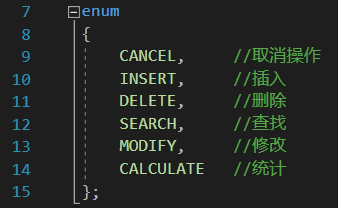
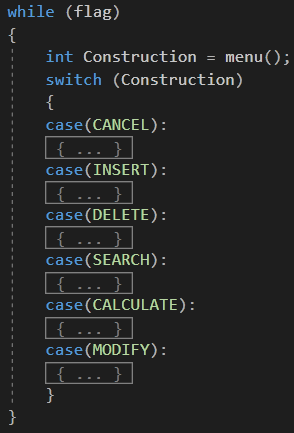
程序包括三个部分：

（一）操作指引的函数，提示输入对应的字母进行不同的操作

（二）储存考试报名系统信息的实例类。

（三）实例类实现对数据的增删查改等相关操作。

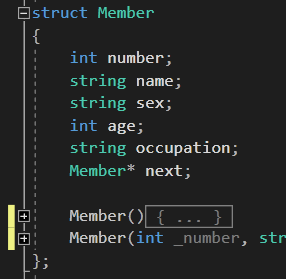
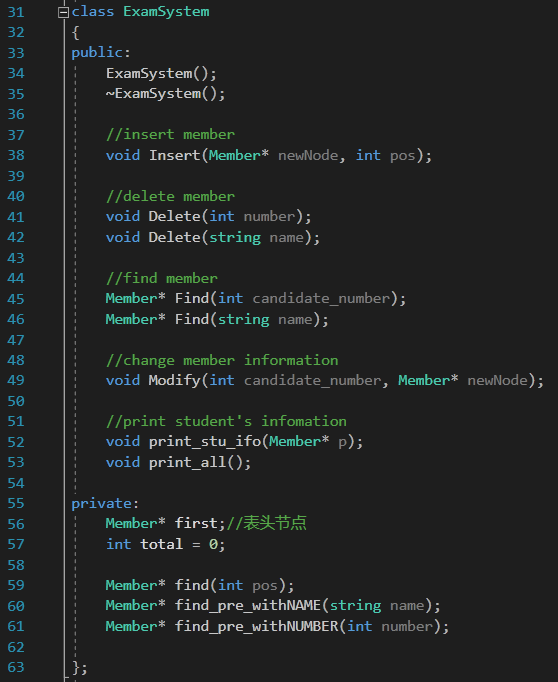
**2.2 操作指引函数**



menu()主要功能是展现字母和操作的对应关系，提示使用者输入某字母以选择想要的功能，并将该值作为函数的返回值。

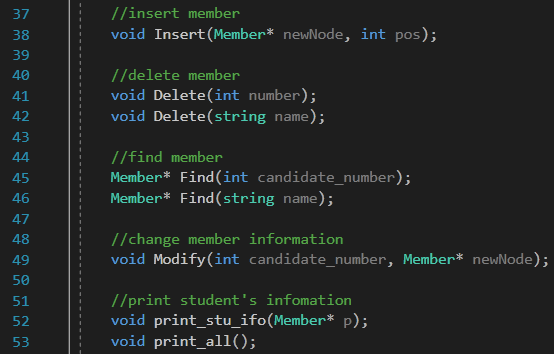
利用枚举enum实现数字和操作的对应关系，便于查看。

**2.3 考试报名系统类等数据结构**



Member是一个结构体，用于表示某个考生的信息。包括学号、姓名、性别、年龄、职位。本程序设计中主要使用链表结构来保存考生信息，因此Member还需带有一个Member\* next指针，指向下一个存储下一个考生信息的结构体。Member设置了两个构造函数，一个是默认的不带参数的构造函数，另一个可以直接依次输入考生信息，构造出一个Member的结构体。

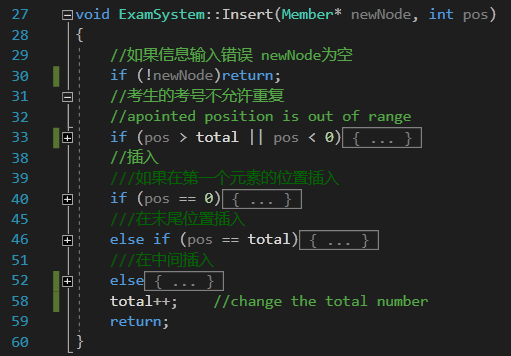
ExamSystem类包括两个基本数据，其一是Member的表头节点Member\* first；其二是int类型的total，用于保存当前报名系统中的考生人数。

**2.4 考试报名系统类相关操作函数**

考试报名系统主要实现的功能有：（一）增删查改操作；（二）查看报名信息操作。

**插入功能：**

void ExamSystem::Insert(Member\* newNode, int pos)

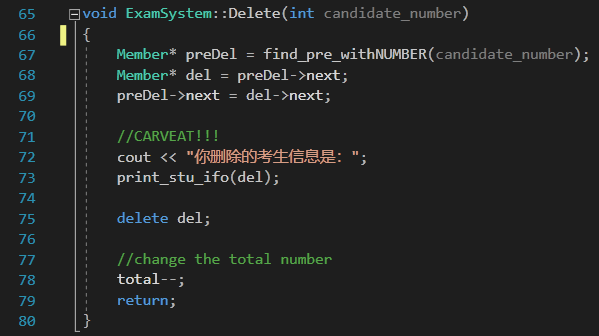


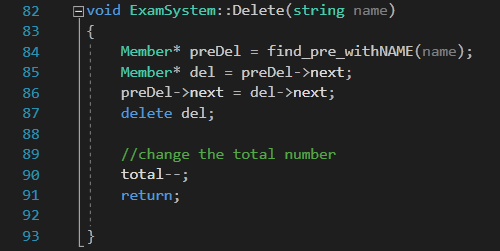
插入时先进行错误判断：（一）检查newNode是否成功初始化；（二）检查考生考号是否唯一；（三）检查考生插入位置是否合理。

如果判断错误，则直接返回，不进行其他操作；反之若均正确，则根据链表插入的操作，将newNode插入到存储考生信息的链表中，并修改total的值。

**删除操作：**

void ExamSystem::Delete(int candidate\_number)

void ExamSystem::Delete(string name) 



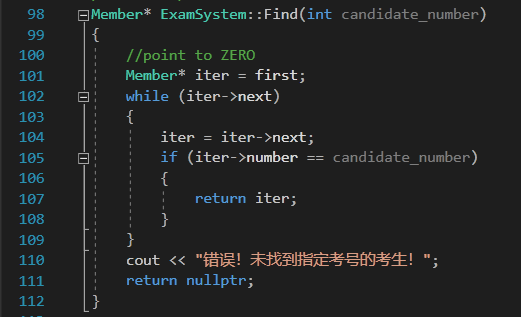
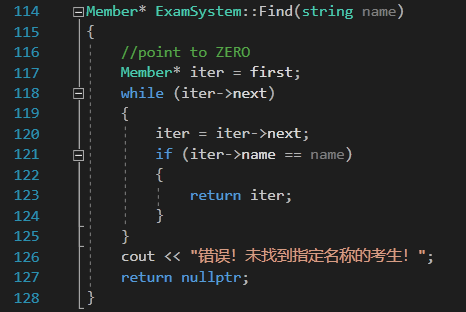
删除操作有两个：（一）删除对应考号的考生；（二）删除对应姓名的考生。

删除时，调用find函数找到符合要求考生的前一个考生，依据链表的相关操作进行节点的删除，并修改total的值。

**查找操作：**

Member\* ExamSystem::Find(int candidate\_number)

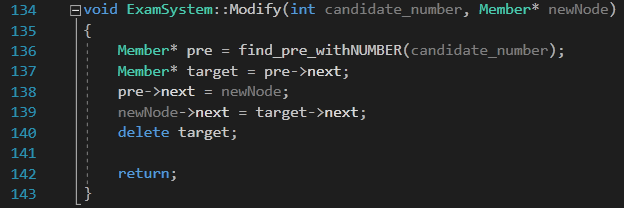
Member\* ExamSystem::Find(string name)



进行查找操作时，从表头节点开始遍历整个链表，若找到符合要求的考生，则返回对应的Member指针。若遍历完整个链表依然没有符合要求的考生，则报错，并返回一个空指针。

**修改操作**

void ExamSystem::Modify(int candidate\_number, Member\* newNode)

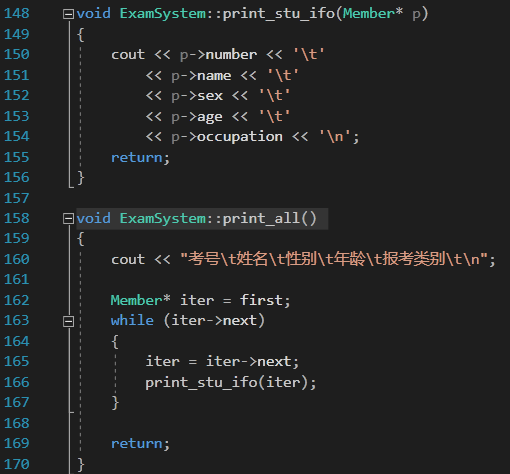


修改操作的逻辑是，根据需要修改的信息直接创建一个新的考生。通过考生的考号查找到对应考生的前一个考生，把旧的信息用新的信息替换掉，即删除原本的考生，添加一个新的考生。

**统计操作**

void ExamSystem::print\_stu\_ifo(Member\* p)

void ExamSystem::print\_all()



统计操作实质上就是从头遍历整个Member链表，根据格式输出所有考生的信息。

**3各功能运行测试**

**3.1 测试数据**

3

1 赵 男 10 语文

2 钱 女 11 数学

3 孙 男 12 英语

1

4

4 李 女 13 物理

5

1

3

3 周 男 20 生物

5

2

3

5

4

1

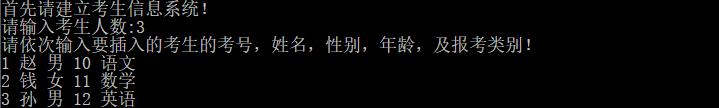
5 吴 女 15 化学

5

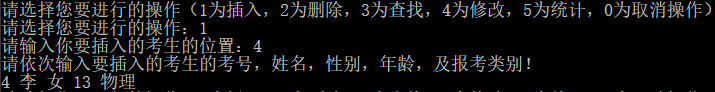
0

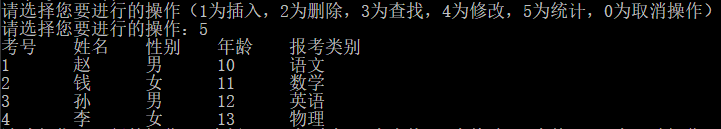
**3.2 测试结果**

初始化：建立考生信息系统

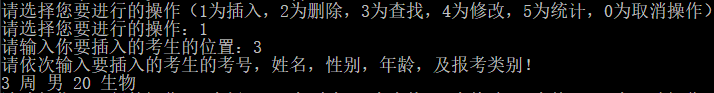


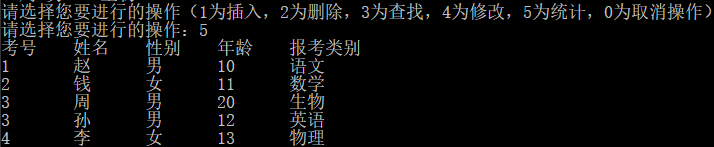
尾插入操作、统计操作



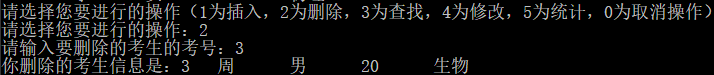


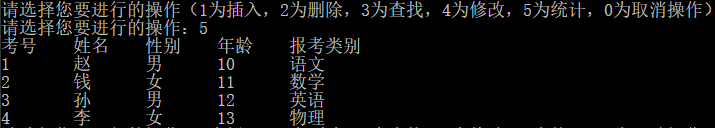
中间插入操作



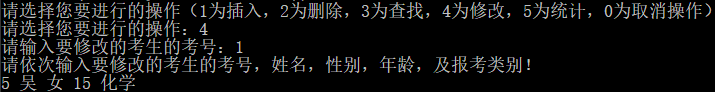


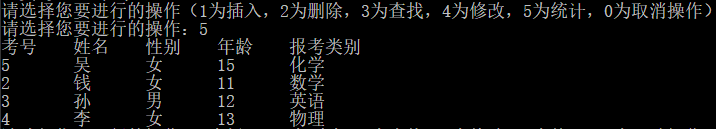
删除操作





修改操作





退出操作

