**考试报名系统项目文档**

**项目简介**：

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。

**项目功能要求**：

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。（建议采用链表实现）

**基本构思**：

本项目的基本思想就是使用双向链表来存储考生的基本信息。插入、删除、查找、编辑等要求的功能都通过修改链表或者链表中的数据来实现。

链表考虑使用List链表类来完成，而考生的数据则装入一个结构体Student中。将Student指针类型的链表头指针放入List类的私有类中。在整个系统运作过程中，结构体动态地产生，动态地分配内存。List的各种类方法实现插入、删除、查找、编辑等功能。

其中，使用双向链表的原因是双向链表的插入和删除操作较单项链表来说比较方便。本项目中需要大量的插入和删除操作，所以考虑使用双向链表。而使用链表来存储数据则很显然是因为链表很灵活，虽然搜索的效率可能不高但是插入删除等操作弥补了这一缺陷。

**数据结构：**

双向链表

**具体实现：**

1. **基本构成**

结构体Student：姓名 name （string）、考号 test\_Number （int）、年龄 age（int）、

性别 sex（string）、类别 category（string）、前指针 Pre （Student\*）、

后指针 Next （Student\*）

链表类 List: 私有类：链表头指针 Head （Student\*）、链表尾指针 End （Student\*）、

考生人数 List\_Size （int）

公有类：

主要函数：

默认构造函数List();

析构函数 ~List();

插入操作 void Insert();

删除操作 void Delete();

编辑操作 void Edit();

统计操作 void Count();

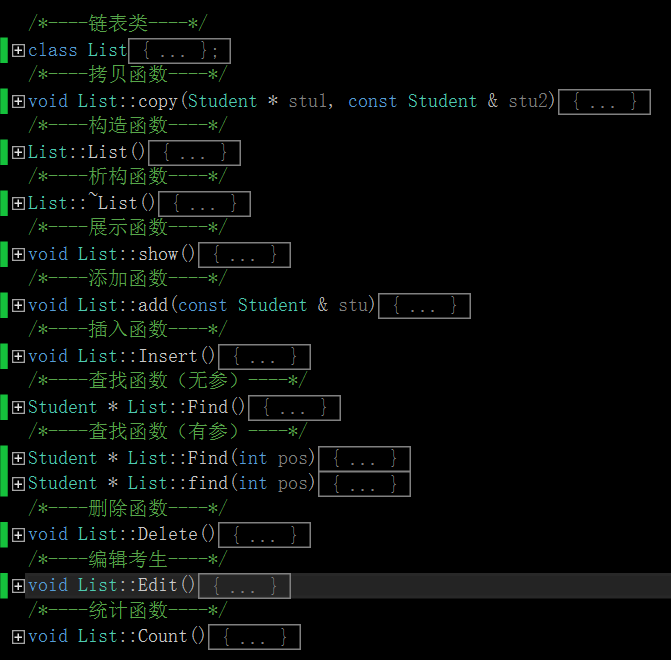
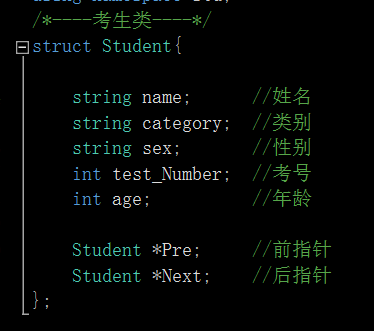
查找操作 void Find();

辅助函数：

展示操作 void show();

拷贝操作 void copy(Student \* stu1, const Student & stu2);

查找操作 Student \* Find(int pos);



1. **方法实现具体说明**

构造函数：从安全角度考虑，将Head & End 设置为NULL，将 List\_Size 设置为 0。

析构函数：创建临时指针 p，将p移动至倒数第二个考生，然后一级一级往前移动，同时释放结构体。

插入函数：要求用户键入插入位置，采用前插入的方式进行插入。插入方式为调用类方法Find(int pos)找到该位置，操作它的前后指针进行插入，同时动态生成需要插入的数据。其中，考虑插入的是第一个，最后一个，中间三种情况，分开来处理。然后List\_Size++。

查找操作：根据用户键入的位置利用临时指针从头指针开始向后移动，找到该考生对应的结构体之后返回该结构体对应的指针。

删除操作：调用方法Find，找到需要删除的结构体之后，对其前后指针进行操作，将其从链表中脱离出来，然后将其内存释放。

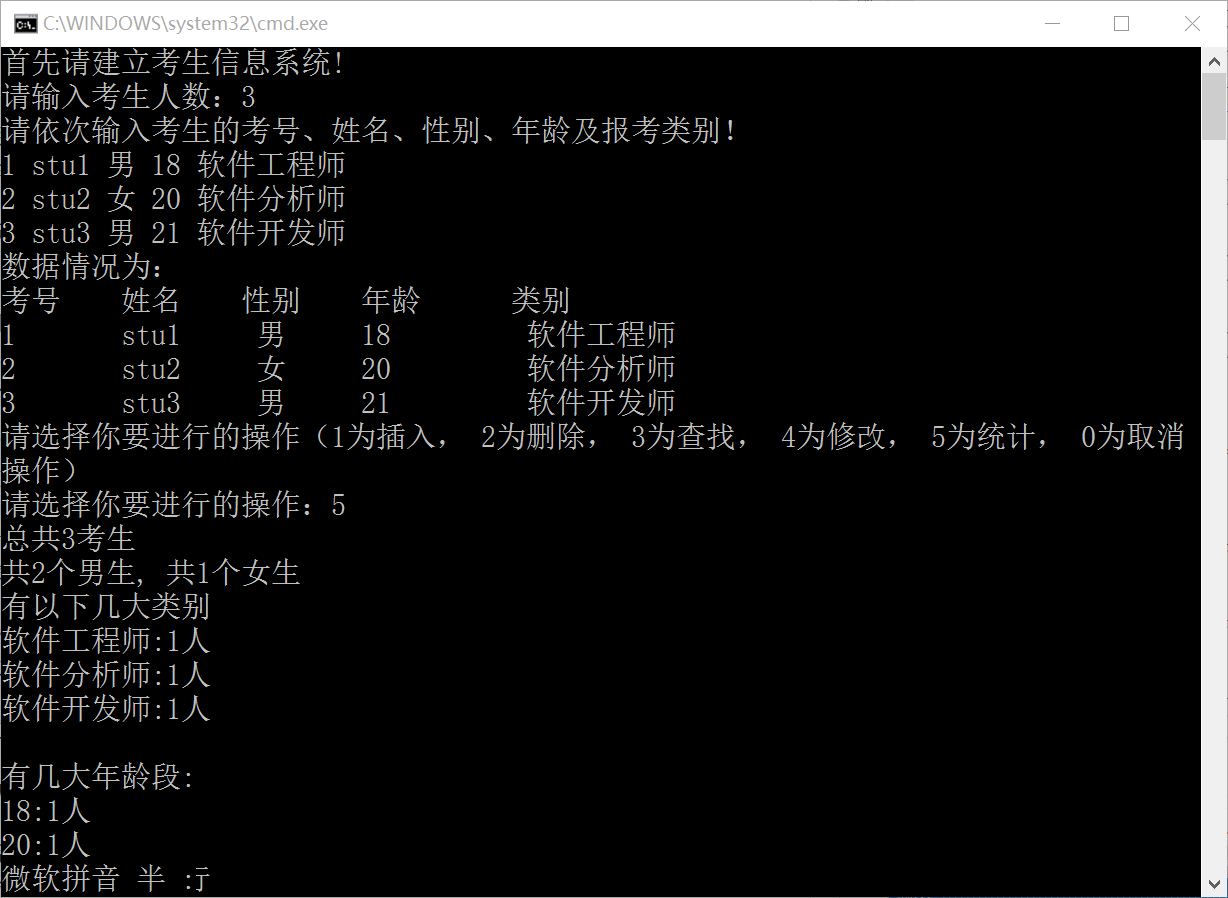
编辑操作：调用方法Find，找到需要编辑的结构体之后，根据用户键入的操作，通过switch进行对对应数据的修改操作。

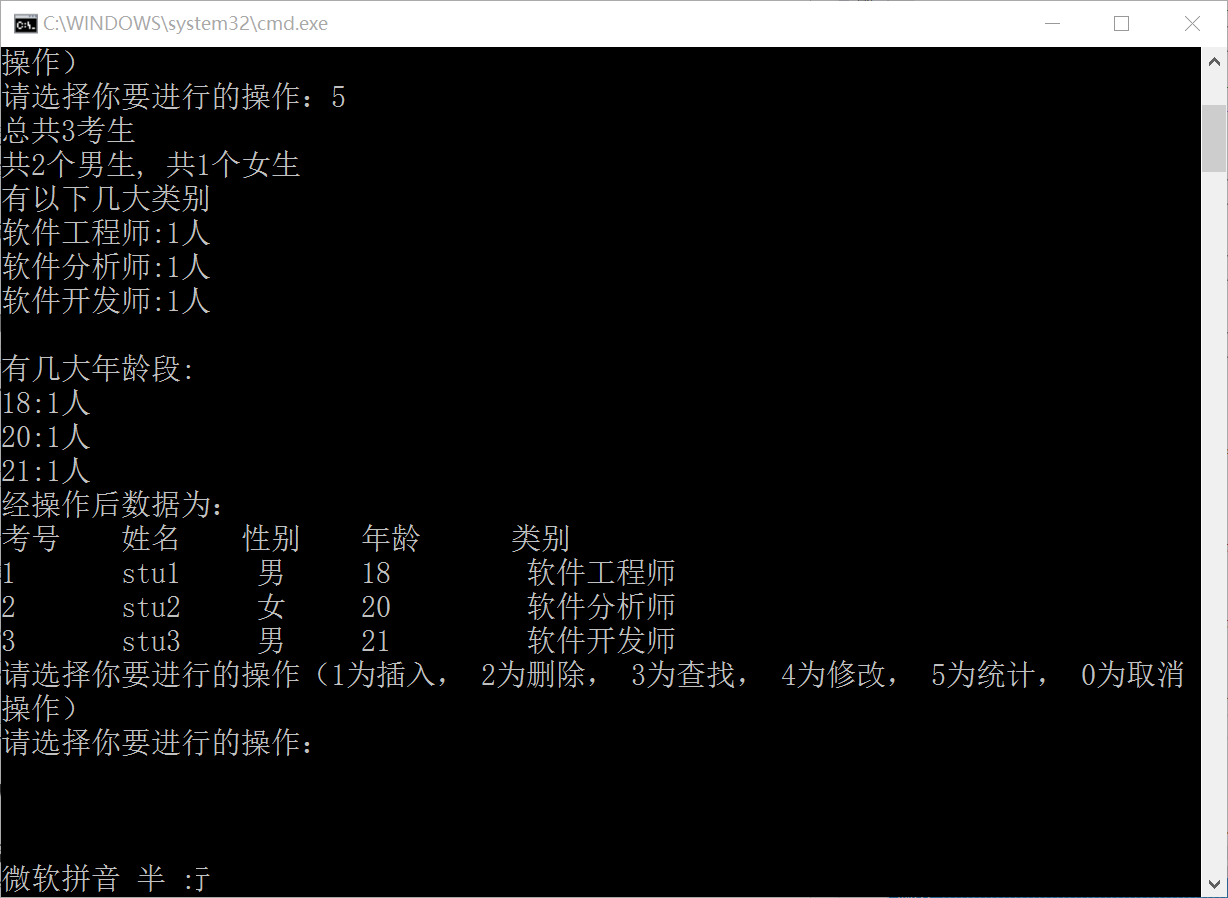
统计操作：对当前链表进行一次遍历，遍历过程中，通过字符串比较，得出男女个数，相同年龄个数，相同类别人数。相同年龄个数和相同类别人数使用了C++ STL中的Vector 和 Pair 容器。最后将统计人数进行输出。

1. **程序运行说明**

程序开始运行之后会出现用户提示信息，要求输入考生人数，之后会要求用户按照考号、姓名、性别、年龄和报考类别的顺序键入考生信息。然后会继续弹出用户界面，用户可以根据相应的数字输入想要进行的操作，每次执行操作之后，都会向用户展示一下修改之后的所有考生信息，以便校对。一旦用户键入“0”，程序将跳出循环，结束运行。

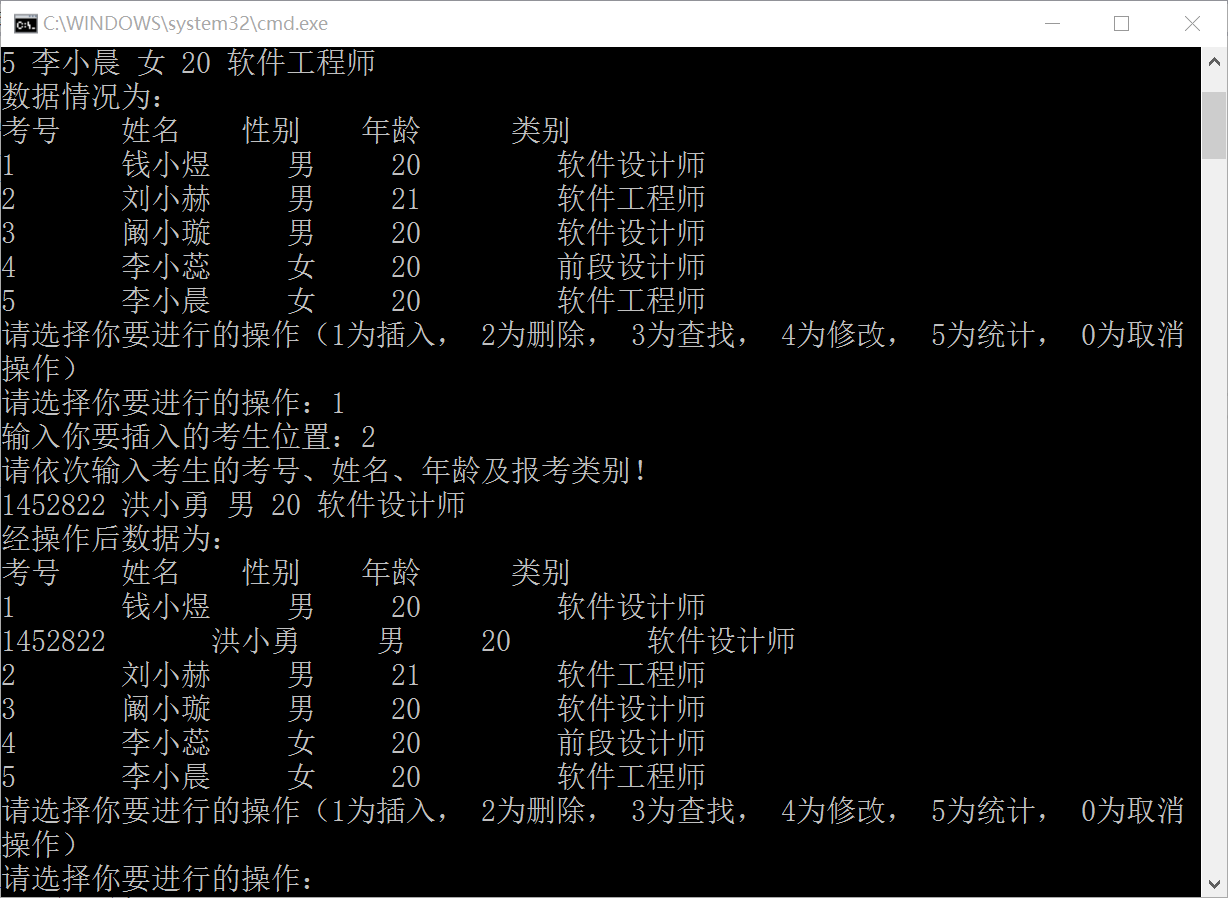
运行图：



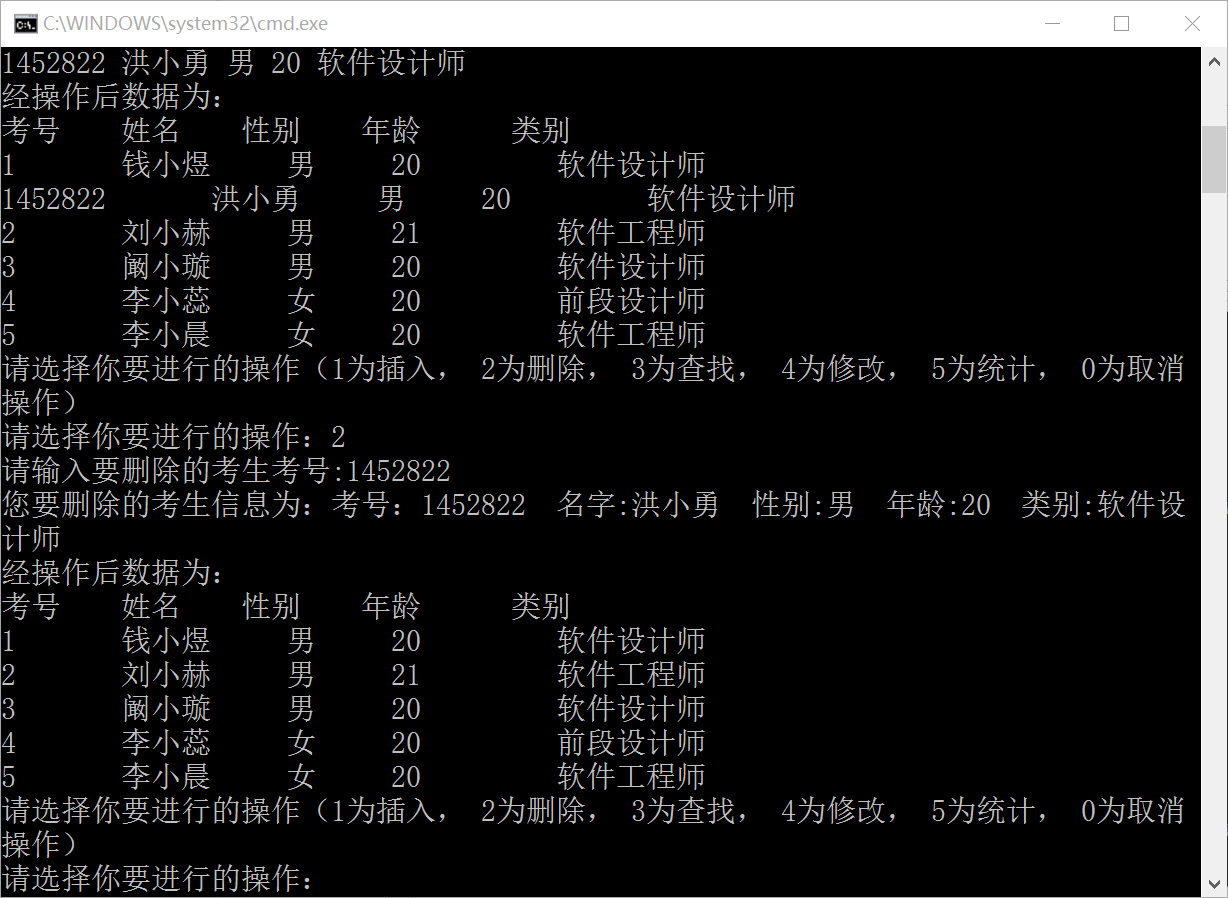


六项功能演示：

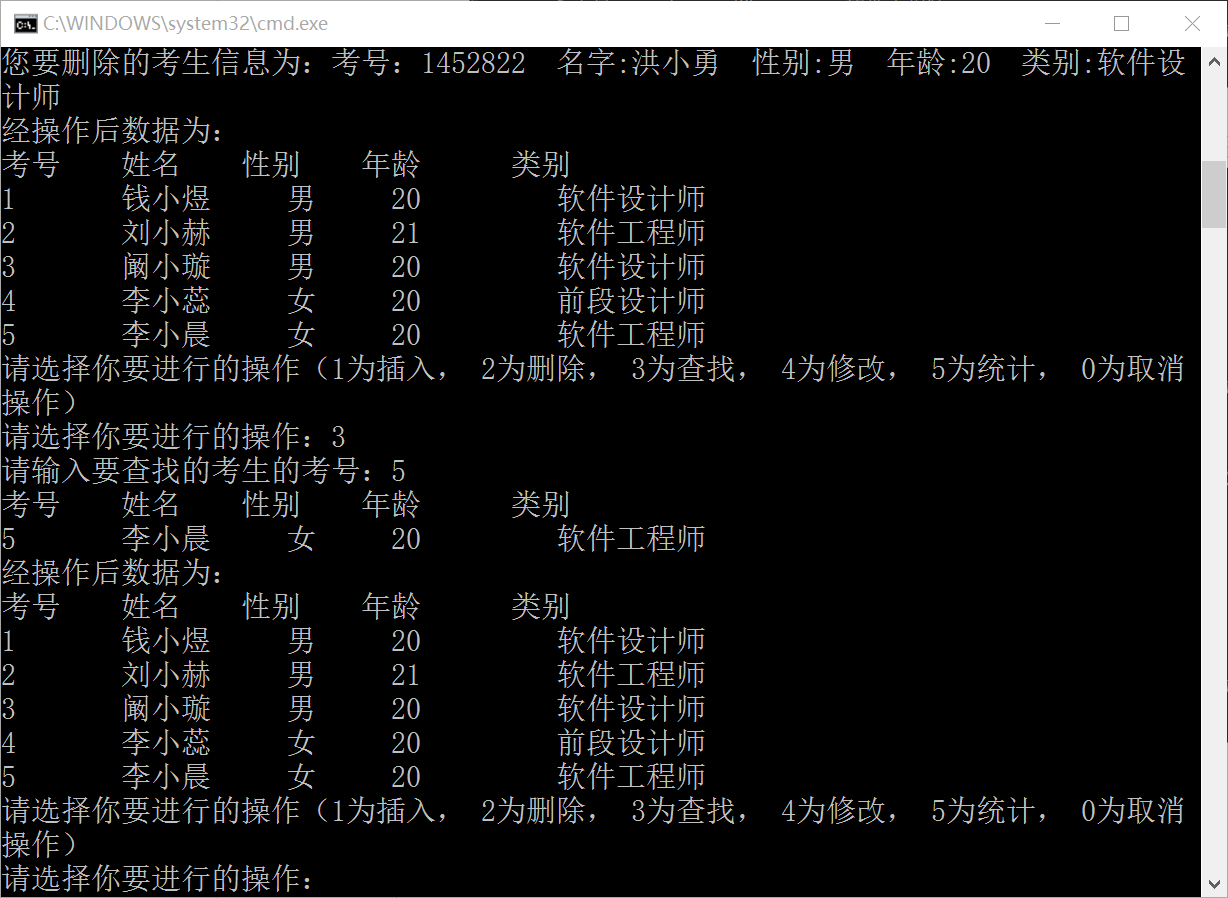
1. 插入演示：



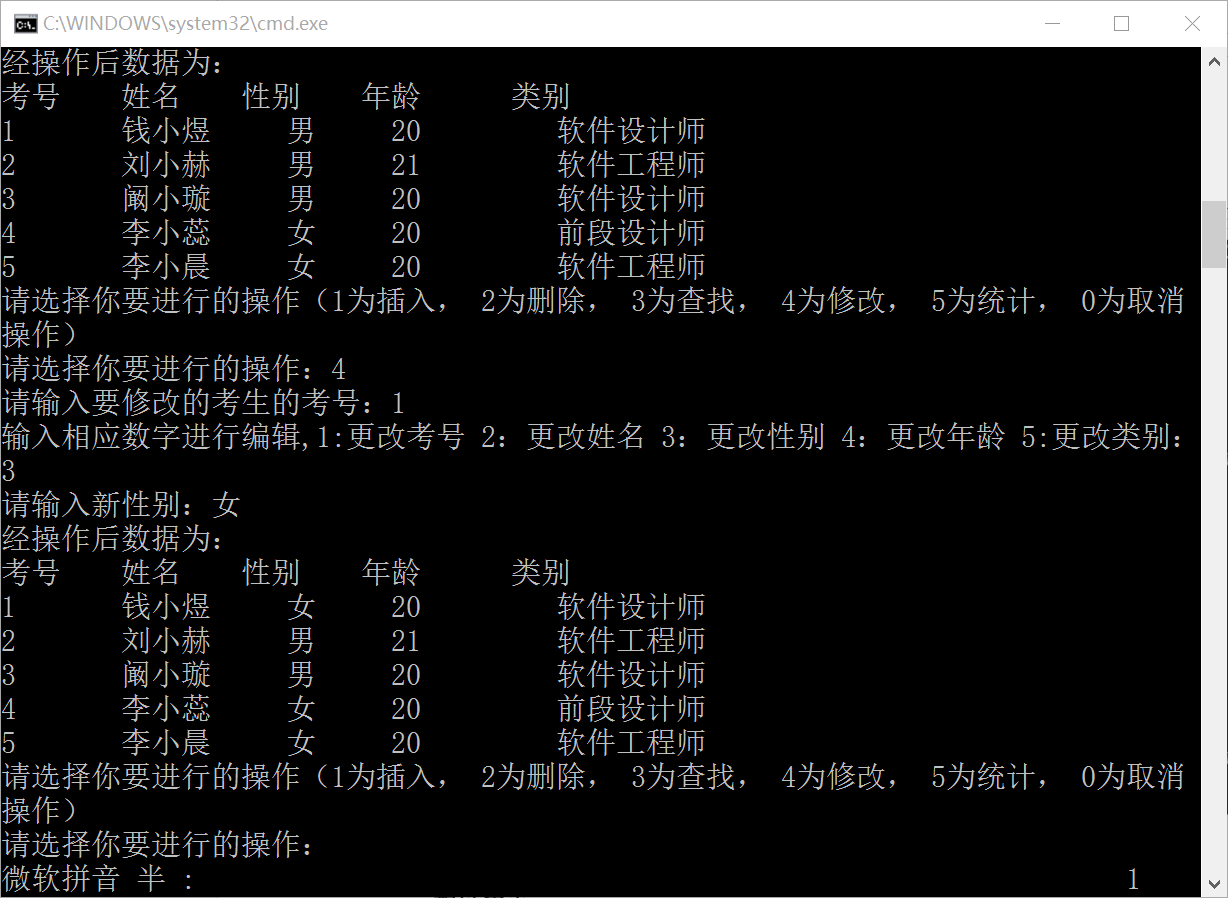
1. 删除演示：



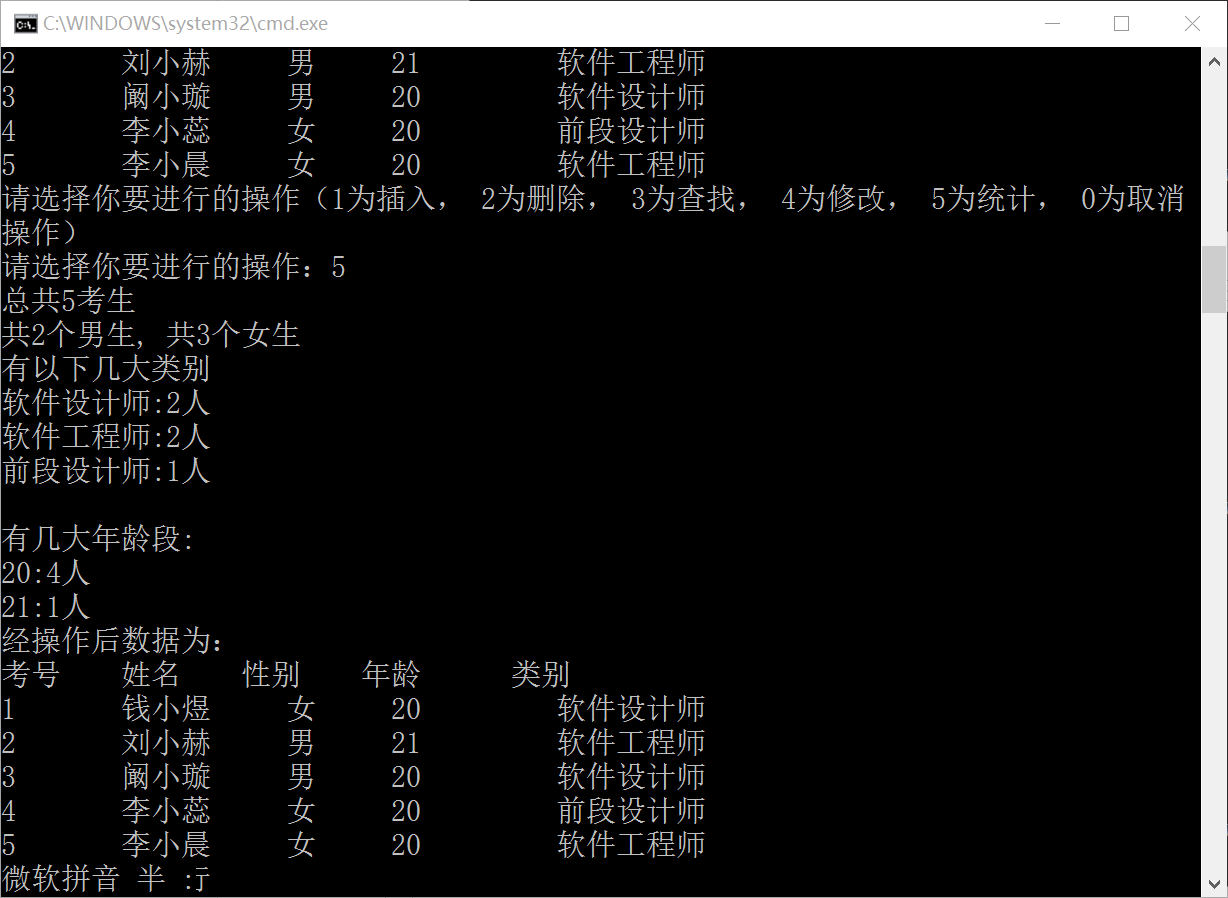
1. 查找演示



1. 修改演示



1. 统计演示



**程序优点：**

1. 在该程序中，我的链表全部都是动态分配的内存，防止了内存泄露的问题，也规避了静态链表浪费内存的问题。对于每一个动态生成的链表，都有对应的delete操作，规避了出现野指针的问题。
2. 函数复用。在实现插入、删除、编辑操作过程中，服用了查找函数，大大减少了代码量。
3. 链表使用了类。将链表进行了封装，体现了C++类的封装性，信息的安全性得到了提高。
4. 功能的强化。在统计功能中，我实现了分类和年龄段的统计，如上运行图所示。灵活运用了C++ STL 中的Vector 容器和 Pair 容器。
5. 容错操作。在该程序中我进行了很多的容错操作来应对用户错误的操作。如用户性别必须为男或女，插入的位置不能不合理等等。如下所示：

while (!temp){

cout << "对不起，没有找到您所要求插入的位置，请重新输入位置, 或者按0取消：";

int pos2;

cin >> pos2;

if (pos2 == 0)

return;

temp = Find(pos2);

}

while (!temp){

cout << "输入有误，请重新输入或者按0取消：";

int pos2;

cin >> pos2;

if (pos2 == 0)

return;

temp = Find(pos2);

}

while (temp.sex != "男" && temp.sex != "女"){

cout << endl << "性别错误， 重新输入性别(男 or 女):";

cin >> temp.sex;

}

while (chose > 5 || chose < 1){

cout << "输入错误，按0取消或输入相应数字进行编辑,1:更改考号 2：更改姓名 3：更改性别 4：更改年龄 5:更改类别：";

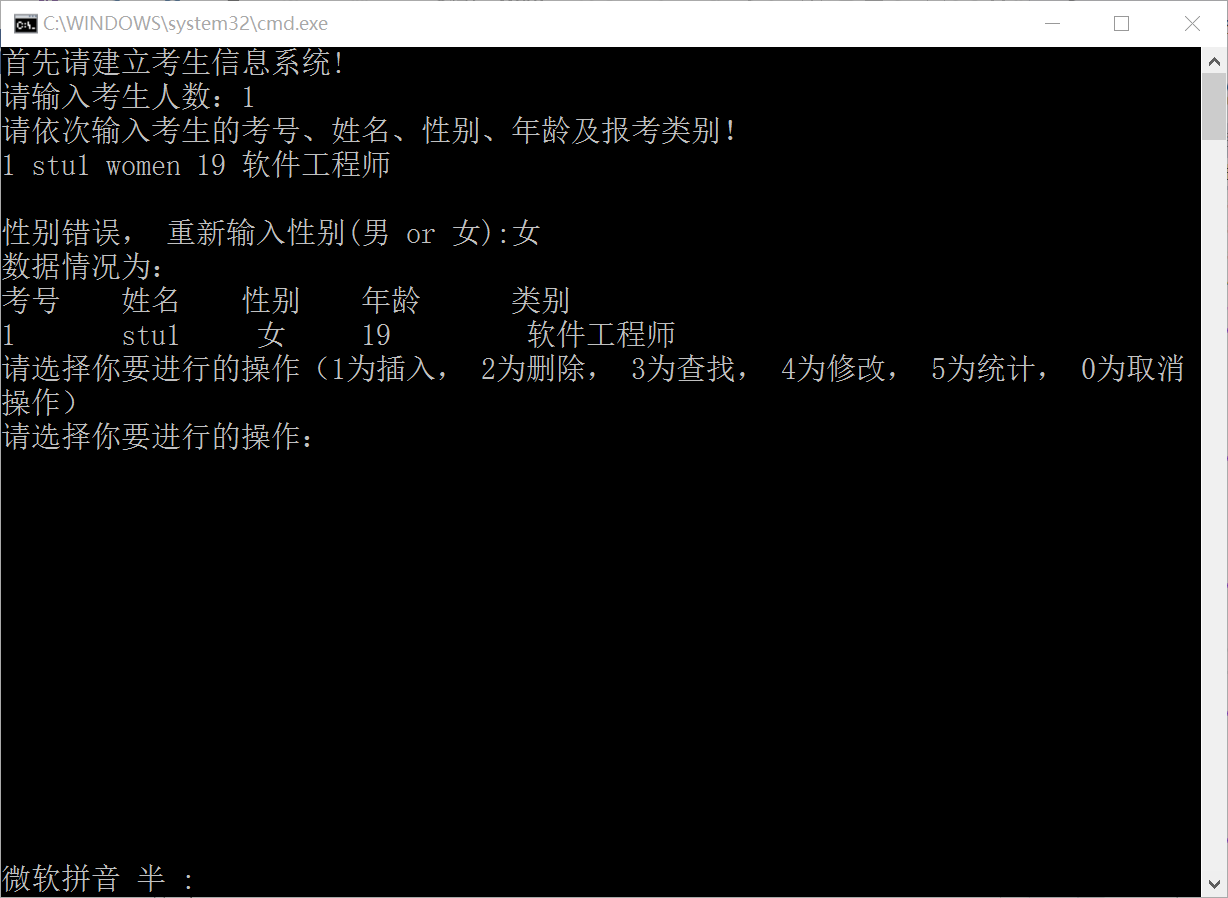
cin >> chose;

if (chose == 0)

return;

}

运行图：



**收获**

该项目的撰写带给我很多收获，比如在代码操练上，在对数据结构的理解上。其中最大的收获在于让我对链表这一常用的数据结构有了更深的认识。

链表有很多种，单向链表、双向链表、循环链表等。这道题考察了我对双向链表的认识。链表尽管查找效率很低，但是在插入和删除数据上很有效，是其他数据结构无法媲美的。

在该项目中我使用了动态内存分配，每个new 对应一个 delete，动了不少脑筋，我认为这是最好的数据内存方式，不浪费也不会造成内存泄漏。

使用类的方法来写是对我C++语言的考验。我认为使用类来写增加安全性，适合考试报名系统。

写双向链表过程中，我的逻辑思维必须非常清晰，稍有不慎就会写错出现野指针。很感谢这样的作业，能让我静下心来去写代码，使我的代码实现能力得到很大的提升。