**项目说明文档**

**数据结构课程设计**

**——考试报名系统**

作者姓名：香宁雨

学号：1954098

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.功能分析**

**2.设计**

**2.1数据结构设计**

**2.2类结构设计**

**2.3类的成员与方法**

2.3.1链表节点类（ListNode）

2.3.2链表类（List）

**2.4测试函数**

**3.实现**

**3.1链表节点类（ListNode）方法实现**

3.1.1重载输入输出运算符

3.1.2PrintNode()方法

**3.2链表类（List）方法实现**

3.2.1构造函数及析构函数

3.2.2Creat()方法

3.2.2.1流程图

3.2.2.2代码实现

3.2.2.3输出测试

3.2.2.3.1错误输入

3.2.2.3.2正确输入

3.2.3Insert()方法

3.2.3.1流程图

3.2.3.2代码实现

3.2.3.3输出测试

3.2.3.3.1错误输入

3.2.3.3.2正确输入

3.2.4Modify()方法

3.2.4.1流程图

3.2.4.2代码实现

3.2.4.3输入测试

3.2.4.3.1错误输入

3.2.4.3.2正确输入

3.2.5Remove()方法

3.2.5.1流程图

3.2.5.2代码实现

3.2.5.3输入测试

3.2.5.3.1错误输入

3.2.5.3.2正确输入

3.2.6SearchCandidateNumber(int num)方法

3.2.6.1流程图

3.2.6.2代码实现

3.2.6.3输入测试

3.2.6.3.1错误输入

3.2.6.3.2正确输入

3.2.7Print()方法

3.2.7.1代码实现

3.2.7.2输出测试

**4.补充**

**1.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。除此之外，还需要实现正确退出考试报名系统的功能。

**2.设计**

**2.1数据结构设计**

因为考试报名系统涉及大量的查找、插入、修改及删除功能，所以我们需要选择一个可以在低复杂度进行这些操作的数据结构。基于这种特点，链表这种可以随意增加和删除的数据结构非常符合我们的需求，所以本次项目使用链表进行数据的储存和修改增删。与此同时，为了实现代码的简洁防止冗余，我们采用空的头节点来减少链表中特殊结点的个数，让所有方法对各节点具有普适性。

**2.2类结构设计**

链表包括两个抽象数据类型，链表结点类和链表类，一般会采用嵌套或者继承等多种关系。本项目采用声明友元函数的方式使二者产生联系，使得链表类可以访问链表结点类中的元素。为便于实现题目中所提到的操作，本项目采用双向链表的方式来储存所有元素，使链表类元素既可以向前遍历也可以向后遍历，简化遍历的难度。

**2.3类的成员和方法**

2.3.1链表节点类（ListNode）



2.3.2链表类（List）



**2.4测试函数**

本项目中采用main函数作为测试函数来验证考生信息系统的正确性，并添加了用户友好的提示，让此系统更贴近用户需求。其中使用system(“pause”)来防止可执行文件的闪退。



**3.实现**

**3.1链表节点类（ListNode）方法实现**

3.1.1重载输入输出运算符

为操作方便，我们对链表节点对象进行输入输出运算符的重载，并进行输入是否正确的判断增加程序的用户友好性及其健壮性。



3.1.2PrintNode()方法

为了充分利用类的特性，即封装性，我们采用PrintNode()方法进行结点的输出，这样可以减少从类的外部对类内成员的调用，转而用类自己的方法对自己的调用，更加充分利用了类的特性。同时，我们对当前结点进行判断，如果当前结点不满足输出要求则输出报错信息。



**3.2链表类（List）方法实现**

3.2.1构造函数及析构函数

因为在代码调用过程中我们使用new进行了内存的分配，所以在结束调用之后需要采用delete进行内存的释放，来确保代码的正确性。

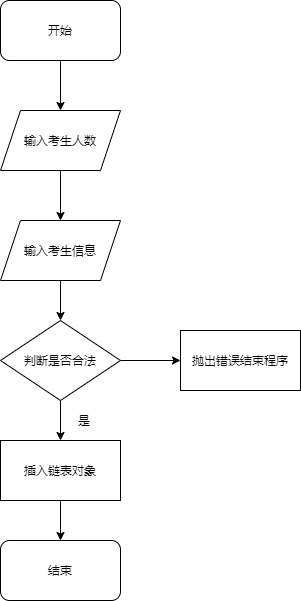




3.2.2Creat()方法

Creat()方法是建立初始考试系统的方法，在代码最开始的时候被调用。

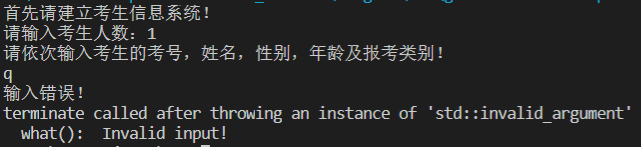
3.2.2.1流程图



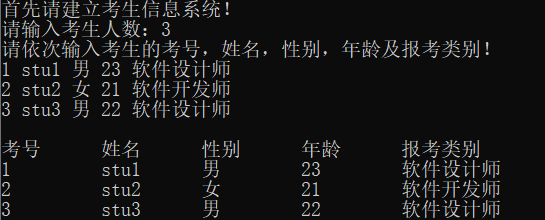
3.2.2.2代码实现

3.2.2.3输出测试

3.2.2.3.1错误输入



3.2.2.3.2正确输入



3.2.3Insert()方法

3.2.3.1流程图

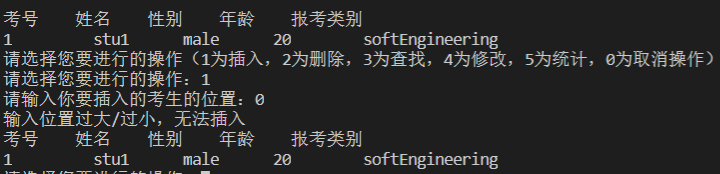


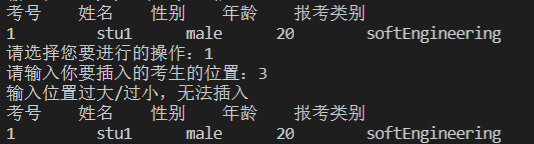
3.2.3.2代码实现



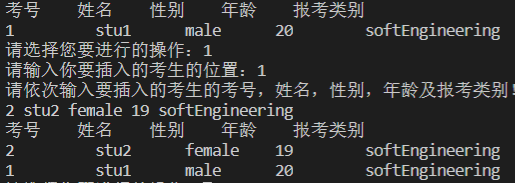
3.2.3.3输出测试

3.2.3.3.1错误输入





3.2.3.3.2正确输入



3.2.4Modify()方法

3.2.4.1流程图

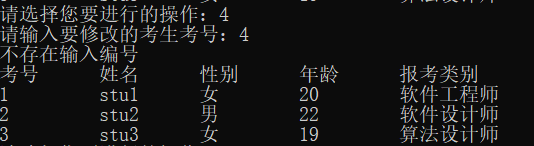


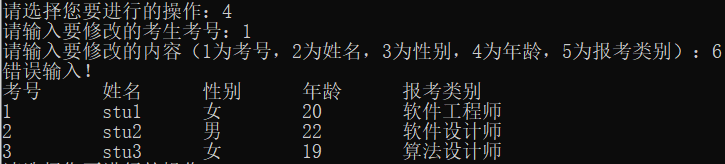
3.2.4.2代码实现



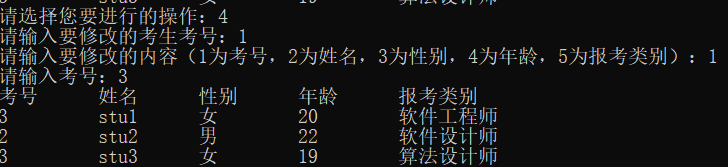
3.2.4.3输入测试

3.2.4.3.1错误输入





3.2.4.3.2正确输入



3.2.5Remove()方法

3.2.5.1流程图

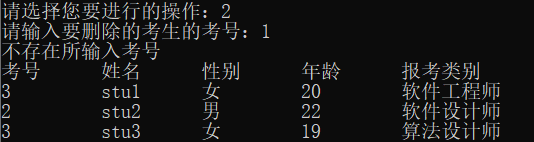


3.2.5.2代码实现

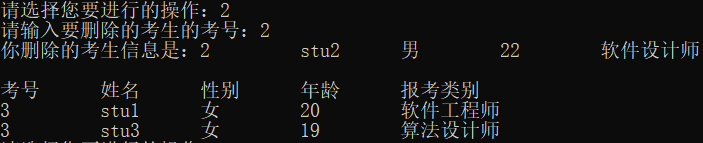


3.2.5.3输入测试

3.2.5.3.1错误输入

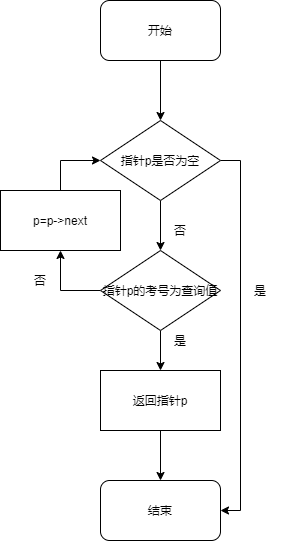


3.2.5.3.2正确输入



3.2.6SearchCandidateNumber(int num)方法

3.2.6.1流程图

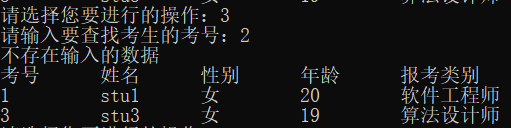


3.2.6.2代码实现

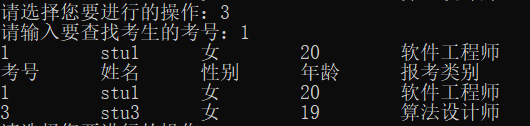


3.2.6.3输入测试

3.2.6.3.1错误输入



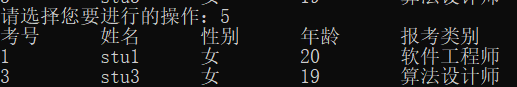
3.2.6.3.2正确输入



3.2.7Print()方法

3.2.7.1代码实现

3.2.7.2输出测试



**4.补充**

程序源代码在Visual Studio2019可以用中文进行提示和输入，在Visual Studio Code终端中可以用中文进行提示和输入，但无法读取输入的中文字符，且生成的exe文件不支持中文提示及中文输入，故在进行文档编写之后为确保exe文件运行正确，项目编写者将源代码中所有中文提示字符更改为英文字符，建议输入也采用英文输入。