

# 目录

## 目录

### 1.项目背景

### 2.需求分析

### 3.设计思路

数据结构

功能实现

### 4.核心代码说明

节点类

链表类

各功能对应函数

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

统计考生信息

函数接口说明

### 5.使用方法

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

统计考生信息

退出程序

## 1.项目背景

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。借助计算机程序，设计一套考试报名系统，可以很大程度上便于这项工作的开展，能够更有效率、更加便捷的管理考生信息

## 2.需求分析

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

# 3.设计思路

## 数据结构

采用单向链表的数据结构。考虑到考试报名系统需要满足无人数上限的自由添加、删除操作，选用链表更为合适。将单个考生的信息作为一个链表节点，以各考生唯一的考号作为键值，来实现查询、修改等操作。

## 功能实现

功能	设计思路
输入考生信息	创建链表，添加节点
输出考生信息	搜索节点+信息输出
查询考生信息	搜索节点
添加考生信息	添加节点
修改考生信息	搜索节点+信息修改
删除考生信息	搜索节点+删除节点
统计考生信息	链表类中建立统计值，增删时根据具体信息改变

# 4.核心代码说明

## 节点类

```
1 //Save student's information
2 struct Student {
3     int number;
4     string name;
5     string gender;
6     int age;
7     string position;
8 };
```

用于存储单个考生信息，包含考号（number），姓名（name），性别（gender），年龄（age），报考的类别（position）

## 链表类

```
1 //Linklist class
2 class LinkList {
3 private:
4     Node *head;//Create a head node for the link list
5     int length; //Save the length of the link list
```

```

6     int boy_amount;//Save total number of boys
7     int girl_amount;//Save total number of girls
8 public:
9     //Default constructor
10    LinkList() {
11        head = new Node("head");
12        length = 0;
13        boy_amount = 0;
14        girl_amount = 0;
15    };
16
17    //Add node at the list tail
18    void AddNode(Student newStudent);
19    //Insert node at a precise position
20    void InsertNode(Student newStudent, int position);
21    //Display node information
22    void ShowNode(Node *node) ;
23    //Display the Linklist
24    void ShowAll();
25    //Delete node by key words
26    void DeleteNode(int number);
27    //Search student by key words
28    Node* SearchNode(int number) ;
29    //Modify the information
30    void ModifyNode(int number, Student &newMessage);
31    //Analyse the information
32    void AnalyseInfo();
33 };

```

采用了带头结点的单向链表，类的成员变量和成员函数的功能均——在代码中通过注释说明

## 各功能对应函数

### 输入考生信息

在main函数中读取考生信息，每有一个考生创建一个新节点存储信息，然后调用链表 `void AddNode(Student newStudent)` 函数，依次将读取到的节点添加如链表

main部分：

```

1  cout << "首先请建立考生信息系统！ " << endl;
2  cout << "请输入考生人数:";
3  cin >> amount;
4  cout << endl << "请依次输入考生的考号，姓名，性别，年龄以及报考类别！ " << endl;
5  for (int i = 0; i < amount; i++) {
6      Student temp;
7      cin >> temp.number >> temp.name >> temp.gender >> temp.age >> temp.position;
8      students.AddNode(temp);
9  }

```

AddNode函数部分：

```

1 //Add node at the list tail
2 void AddNode(Student newStudent) {
3     Node *node = new Node(newStudent);
4     Node *temp = head;
5     if (newStudent.gender == "男")
6         boy_amount++;
7     else
8         girl_amount++;
9     //find the tail node
10    while (temp->pNext!=NULL){
11        temp = temp->pNext;
12    }
13    temp->pNext = node;
14    length++;
15 }

```

## 输出考生信息

main函数内，整个操作在while(1)循环中，除非输入0退出，每一次循环开始，调用链表类的 `void ShowAll()` 函数，格式化输出当前链表中所有考生信息

```

1 //Display the Linklist
2 void ShowAll() {
3     Node *temp = head;
4     cout <<endl<< "考号" << "\t" << "姓名" << "\t" << "性别"<<"\t"<< "年龄"<<"\t"<< "报考类别" << endl;
5     for (int i = 0; i < length; i++) {
6         temp = temp->pNext;
7         ShowNode(temp);
8     }
9 }

```

## 查询考生信息

通过链表类的 `Node* SearchNode(int number)` 和 `void ShowNode(Node *node)` 函数实现

前者通过考生考号对链表进行遍历搜索，发现对应节点后返回该节点地址,后者以节点地址作为参数格式化输出该节点存储的考生信息

```

1 //Search student by key words
2 Node* SearchNode(int number) {
3     Node *temp = head;
4     while (temp->pNext != NULL) {
5         if (temp->pNext->stu.number == number)
6             break;
7         temp = temp->pNext;
8     }
9     if (temp->pNext == NULL) {
10        return NULL;
11    }
12    else
13        return temp->pNext;

```

```
14     }
```

```
1 //Display node information
2 void ShowNode(Node *node) {
3     if (node != NULL) {
4         Student stu = node->stu;
5         cout << stu.number << "\t" << stu.name << "\t" << stu.gender << "\t" << stu.age << "\t"
6         << stu.position << endl;
7     }
8     else
9         cout << "该考生信息不存在" << endl;
10 }
```

## 添加考生信息

在main函数中，将考生信息读入新创建的节点中，同时读取所需要插入的位置，再调用成员函数 `void InsertNode(Student newStudent, int position)` 完成插入操作

```
1 //Insert node at a precise position
2 void InsertNode(Student newStudent, int position) {
3     Node *temp = head;
4     Node *node = new Node(newStudent);
5     if (newStudent.gender == "男")
6         boy_amount++;
7     else
8         girl_amount++;
9     for (int i = 0; i < position-1; i++) {
10         if (i <= length)
11             temp = temp->pNext;
12     }
13     node->pNext = temp->pNext;
14     temp->pNext = node;
15     length++;
16 }
```

## 修改考生信息

在main函数中记录需要修改信息的考生考号以及新信息，建立临时节点来存储更新后的节点信息，调用成员函数 `void ModifyNode(int number, Student& newMessage)` 对信息进行修改，在 `ModifyNode` 中首先通过调用 `SearchNode` 函数查找到对应节点，然后更新值

main部分：

```
1 cout << "请输入你要修改的考生考号：" << endl;
2 cin >> number;
3 temp3 = students.SearchNode(number);
4 if (temp3) {
5     int modify_key;
6     cout << "请选择你要修改的信息：(1为考号，2为姓名，3为性别，4为年龄，5为报考类别)" << endl;
7     cin >> modify_key;
```

```

8      cout << "请输入新的信息: " << endl;
9      switch (modify_key) {
10         case 1:
11             cin >> temp3->stu.number;
12             break;
13         case 2:
14             cin >> temp3->stu.name;
15             break;
16         case 3:
17             cin >> temp3->stu.gender;
18             break;
19         case 4:
20             cin >> temp3->stu.age;
21             break;
22         case 5:
23             cin >> temp3->stu.position;
24             break;
25         default:
26             break;
27     }
28 }

```

ModifyNode部分:

```

1  //Modify the information
2  void ModifyNode(int number, Student &newMessage) {
3      Node *temp = SearchNode(number);
4      if (temp) {
5          temp->stu = newMessage;
6      }
7  }

```

## 删除考生信息

main部分读取到需要删除的考生号后，调用成员函数 `void DeleteNode(int number)`，在函数内，先调用 `SearchNode` 函数搜索到对应节点，然后调用 `ShowNode` 输出该节点信息，再执行链表的删除操作

```

1  //Delete node by key words
2  void DeleteNode(int number) {
3      Node *del = SearchNode(number);
4      if (del != NULL) {
5          cout << "你删除的考生信息是: ";
6          ShowNode(del);
7          Node *temp = head;
8          while (temp->pNext != NULL) {
9              if (temp->pNext->stu.number == number)
10                 break;
11                 temp = temp->pNext;
12             }
13             temp->pNext = temp->pNext->pNext;
14             if (del->stu.gender == "男")
15                 boy_amount--;

```

```
16     else
17         girl_amount--;
18     delete del;
19     length--;
20 }
21 }
```

统计考生信息

在创建链表时，及通过重载的默认构造函数将 boy\_amount 和 girl\_amount 初始化为0，后面的 AddNode InsertNode DeleteNode 函数中，都添加了对所处理节点性别的判断，实现添加删除节点后性别统计的实时变化。通过调用成员函数 AnalyseInfo 输出统计的信息

```
1 //Analyse the information
2 void AnalyseInfo() {
3     cout << "共有" << length << "名考生"<<endl;
4     cout << "其中男生有: " << boy_amount << "名" << "女生有: " << girl_amount << "名" <<
5     endl;
6 }
```

函数接口说明

返回值类型	成员函数名	参数	属性	功能
\	LinkedList	\	public	默认构造函数
void	AddNode	(Student& newStudent)	public	在链表末尾添加节点
void	InsertNode	(Student& newStudent,int position)	public	在链表指定位置添加节点
void	ShowNode	(Node *node)	public	输出节点信息
void	ShowAll	\	public	输出所有节点信息
void	DeleteNode	(int number)	public	删除对应考号节点
Node*	SearchNode	(int number)	public	返回对应考号节点地址，如果未搜索到，则返回NULL
void	ModifyNode	(int number,Student &newMessage)	public	更新对应节点信息
void	AnalyseInfo	\	public	输出统计的信息

5.使用方法

运行 student\_info\_system.exe ，打开考生信息系统

## 输入考生信息

```
G:\coding\Data Structure\数据结构\Data-Structure-Practice\exercise_1\student_info_system.exe
首先请建立考生信息系统！
请输入考生人数：
```

首先会提示建立信息系统，要求用户输入首次需要录入的考生人数及信息

```
G:\coding\Data Structure\数据结构\Data-Structure-Practice\exercise_1\student_info_system.exe
首先请建立考生信息系统！
请输入考生人数:3

请依次输入考生的考号，姓名，性别，年龄以及报考类别！
1 stu1 女 20 软件测试员
2 stu2 男 21 软件开发师
3 stu3 男 20 软件设计师

考号      姓名      性别      年龄      报考类别
1          stu1      女         20         软件测试员
2          stu2      男         21         软件开发师
3          stu3      男         20         软件设计师
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）
```

输入完成后系统会显示出当前存储信息，即存储成功，并提示进行下一步操作

## 输出考生信息

每一次操作完成后会遍历链表，格式化输出所有考生信息。并输出操作提示

```
考号      姓名      性别      年龄      报考类别
1          stu1      女         20         软件测试员
2          stu2      男         21         软件开发师
3          stu3      男         20         软件设计师
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）
```

## 查询考生信息

在功能选择步骤输入3进入查找功能

```
考号      姓名      性别      年龄      报考类别
1          stu1      女         20         软件测试员
2          stu2      男         21         软件开发师
3          stu3      男         20         软件设计师
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）
3
请输入你要查找的考生考号：
2
考号      姓名      性别      年龄      报考类别
2          stu2      男         21         软件开发师
```

随后输入需查询考生考号，系统搜索到后会输出考生信息，如考号有误，系统会提示“该考生信息不存在”



```
考号      姓名      性别      年龄      报考类别
1          stu1      女        20        软件测试员
2          stu2      男        21        软件开发师
3          stu3      男        20        软件设计师
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）
3
请输入你要查找的考生考号：
2
考号      姓名      性别      年龄      报考类别
2          stu2      男        21        软件开发师
```

## 添加考生信息

在功能选择步骤输入1进入插入功能，随后输入插入位置（<表格现有项数）及新考生信息，系统会将该考生信息存储至输入位置，并将该位置后的考生信息顺序后移

```
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）
1
请输入你要插入的考生位置：
2
请依次输入要插入的考生考号，姓名，性别，年龄以及报考类别！
4 stu4 女 19 程序员鼓励师

考号      姓名      性别      年龄      报考类别
1          stu1      女        20        软件测试员
4          stu4      女        19        程序员鼓励师
2          stu2      男        21        软件开发师
3          stu3      男        20        软件设计师
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）
```

## 修改考生信息

在功能选择步骤输入4进入修改功能，随后输入要修改考生的考号，系统若查找到该考生，则会继续提示输入要修改的信息项，并提示输入新值，若未找到，则会提示“该考生信息不存在”

```
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）
4
请输入你要修改的考生考号：
4
请选择你要修改的信息：（1为考号，2为姓名，3为性别，4为年龄，5为报考类别）
5
请输入新的信息：
程序员安慰师
共有4名考生
其中男生有：2名女生有：2名
```

## 删除考生信息

在功能选择步骤输入2进入删除功能，随后输入要删除的考生考号，若系统查找到该考生，则会输出考生信息，并删除该节点，若未找到，则会提示“该考生信息不存在”

```
3      stu3      男      20      软件设计师
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）
2
请输入你要删除的考生考号：
4
你删除的考生信息是： 4      stu4      女      19      程序员安慰师

考号      姓名      性别      年龄      报考类别
1          stu1      女      20      软件测试员
2          stu2      男      21      软件开发师
3          stu3      男      20      软件设计师
```

统计考生信息

在功能选择步骤输入5进入统计功能，统计的项目包括：

- 考生总数
- 男生数量
- 女生数量

```
考号      姓名      性别      年龄      报考类别
1          stu1      女      20      软件测试员
2          stu2      男      21      软件开发师
3          stu3      男      20      软件设计师
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）
5
共有3名考生
其中男生有： 2名女生有： 1名
```

退出程序

在功能选择步骤输入0退出程序，需要注意数据并不会保存