目录

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

```
//Save student's information
struct Student {
    int number;
    string name;
    string gender;
    int age;
    string position;
};
```

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

```
//Linklist class
class LinkList {
private:
    Node *head;//Create a head node for the link list

    int length; //Save the length of the link list
```

```
    int boy_amount;//Save total number of boys
    int girl_amount;//Save total number of girls
public:
    //Default constructor
    LinkList() {
        head = new Node("head");
        length = 0;
        boy_amount = 0;
        girl_amount = 0;
    };

    //Add node at the list tail
    void AddNode(Student newStudent);
    //Insert node at a precise position
    void InsertNode(Student newStudent, int position);
    //Display node information
    void ShowNode(Node *node) ;
    //Display the Linklist
    void ShowAll();
    //Delete node by key words
    void DeleteNode(int number);
    //Search student by key words
    Node* SearchNode(int number) ;
    //Modify the information
    void ModifyNode(int number,Student &newMessage);
    //Analyse the information
    void AnalyseInfo();
};
```

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一一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：

```
cout << "首先请建立考生信息系统！" << endl;
cout << "请输入考生人数:";
cin >> amount;
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;
for (int i = 0; i < amount; i++) {
  Student temp;
  cin >> temp.number >> temp.name >> temp.gender >> temp.age >> temp.position;
  students.AddNode(temp);
}
```

AddNode函数部分：

```
//Add node at the list tail
void AddNode(Student newStudent) {
  Node *node = new Node(newStudent);
  Node *temp = head;
  if (newStudent.gender == "男")
    boy_amount++;
  else
    girl_amount++;
  //find the tail node
  while (temp->pNext!=NULL){
    temp = temp->pNext;
  }
  temp->pNext = node;
  length++;
}
```

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

```
//Display the Linklist
void ShowAll() {
  Node *temp = head;
  cout <<endl<< "考号" << "\t" << "姓名" << "\t" << "性别"<<"\t"<< "年龄"<<"\t"<< "报考类别" <<
endl;
  for (int i = 0; i < length; i++) {
    temp = temp->pNext;
    ShowNode(temp);
  }
}
```

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

```
//Search student by key words
    Node* SearchNode(int number) {
        Node *temp = head;
        while (temp->pNext != NULL) {
            if (temp->pNext->stu.number == number)
                break;
            temp = temp->pNext;
        }
        if (temp->pNext == NULL) {
            return NULL;
        }
        else
            return temp->pNext;
```

```
    }
```

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

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

```
//Insert node at a precise position
void InsertNode(Student newStudent, int position) {
  Node *temp = head;
  Node *node = new Node(newStudent);
  if (newStudent.gender == "男")
    boy_amount++;
  else
    girl_amount++;
  for (int i = 0; i < position-1; i++) {
    if (i <= length)
      temp = temp->pNext;
  }
  node->pNext = temp->pNext;
  temp->pNext = node;
  length++;
}
```

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：

```
cout << "请输入你要修改的考生考号：" << endl;
cin >> number;
temp3 = students.SearchNode(number);
if (temp3) {
  int modify_key;
  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;
  cin >> modify_key;
```

```
  cout << "请输入新的信息: " << endl;
  switch (modify_key) {
    case 1:
      cin >> temp3->stu.number;
      break;
    case 2:
      cin >> temp3->stu.name;
      break;
    case 3:
      cin >> temp3->stu.gender;
      break;
    case 4:
      cin >> temp3->stu.age;
      break;
    case 5:
      cin >> temp3->stu.position;
      break;
    default:
      break;
  }
}
```

ModifyNode部分:

```
//Modify the information
void ModifyNode(int number,Student &newMessage) {
  Node *temp = SearchNode(number);
  if (temp) {
    temp->stu = newMessage;
  }
}
```

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

```
//Delete node by key words
void DeleteNode(int number) {
  Node *del = SearchNode(number);
  if (del != NULL) {
    cout << "你删除的考生信息是: ";
    ShowNode(del);
    Node *temp = head;
    while (temp->pNext != NULL) {
      if (temp->pNext->stu.number == number)
        break;
      temp = temp->pNext;
    }
    temp->pNext = temp->pNext->pNext;
    if (del->stu.gender == "男")

      boy_amount--;
```

```
    else
      girl_amount--;
    delete del;
    length--;
  }
}
```

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

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

函数接口说明

返回值类型	成员函数名	参数	属性	功能
\	LinkList	\	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.使用方法及函数功能演示

运行`student_info_system.exe`，打开考生信息系统

输入考生信息

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

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”

```
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