

Project documentation

Data structure course design

——Examination registration system

Author name: Xuedi Liu

Number: 1752985

instructor: Ying Zhang

College/Major: School of Software Engineering /Software Engineering

1. Analysis
 - 1.1 Project background analysis
 - 1.2 Project function requirements
2. Design
 - 2.1 Data structure design
 - 2.2 Class structure design
 - 2.3 Member and operational design
 1. Node
 2. LinkList
 - 2.4 System design
3. Realization
 - 3.1 Realization of the insert function
 1. Insert function flow chart
 2. Insert function core code
 3. Insert function diagram
 - 3.2 Realization of the delete function
 1. Delete function flow chart
 2. Delete function core code
 3. Delete function diagram
 - 3.3 Realization of the search function
 1. Find function flow chart
 2. Find function core code
 3. Find function diagram
 - 3.4 Realization of the modify function
 1. Modify function flow chart
 2. Modify function core code
 3. Modify function diagram
 - 3.5 Realization of statistical functions.
 1. Statistical function flow chart
 2. Core code of statistical function
 3. Schematic diagram of statistical function
 - 3.6 Realization of exit operation
 1. Exit operation core code
 2. Schematic diagram of exit operation function
 - 3.7 Realization of overall system
 1. Overall system flow chart
 2. Core code for the overall system
 3. Schematic diagram of the overall system
4. Test
 - 4.1 Function testing
 1. Insert function test
 2. Delete function test
 3. Find function test
 4. Modify function test
 5. Statistical function test
 - 4.2 Boundary test
 1. Initialize no input data
 2. Delete header node
 3. After the deletion, the linked list is empty.
 - 4.3 Error test

1. The number of candidates is wrong
2. Opcode error
3. Insert position does not exist
4. Delete test number does not exist
5. Find the test number does not exist
6. Modify the test number does not exist

- **Operating Environment:**

- Unix executables: running on **Unix** platforms
- Linux executables: running on **Linux** platforms
- exe executable file: Windows Console Application, running on 64-bit Windows platform
- **Code hosting platform:** Github

1. Analysis

1.1 Project background analysis

The exam registration system is a system that every school and educational institution needs. It is important for the administrator of the exam and the students who take the exam. However, due to the large number of examinations and the lack of management staff, the examination registration often adds a lot of work to the administrative management department. A quick and easy exam registration system simplifies operations and improves efficiency. This project is a simple simulation of the exam registration system. The following functions are completed by selecting the console option: input candidate information, query candidate information, add candidate information, modify candidate information, and delete candidate information.

1.2 Project function requirements

As a simple exam registration system, we need to implement the functions of establishing, searching, inserting, modifying, deleting, and exiting candidates. Among them, the candidate information includes information such as the test number, name, gender, age, and application category. The project should first determine the data structure of the system, define the class and other member variables and member functions; then implement each member function to complete the corresponding functions of the data operation; finally complete the main function to verify the function of each member function and get the running result .

2. Design

2.1 Data structure design

From the analysis, the examination registration system needs to carry out a large number of additions, deletions, and modifications, and the frequency of updating the registration information table is extremely high. The data structure of the linked list is very convenient for operations such as adding, deleting, and modifying. Therefore, the design of the system considers the data structure of the linked list. At the same time, in order to make the operation easier and safer, the system will attach a head node in front of the node that stores the information, add the tail node after the last node that stores the information, and adopt the mode of the doubly linked list, so that the processing head It is more convenient and safe to insert and delete operations such as the boundary condition of the tail node, which makes the program more concise and more universal.

A linked list is a typical dynamic storage structure. The data is scattered into a series of units called nodes, which are indexed and accessed by pointers. In order to introduce a new node or delete an existing node, it is only necessary to adjust the pointer between a small number of related nodes locally. This means that the use of dynamic storage strategies can greatly reduce the cost of dynamic operations.

2.2 Class structure design

The linked list data structure generally contains two abstract data types - a node class (Node) and a linked list class (LinkedList). Two classes. The node class needs to not only save the corresponding data item, but also record its predecessor and successor location. The linked list class macroscopically manages the update operations of the entire linked list.

2.3 Member and operational design

1.Node

Private members:

```
int _number;//考号
std::string _name;//姓名
std::string _sex;//性别
int _age;//年龄
std::string _type;//报考科目
```

Public members:

```
Node *next; //后继指针
Node *pre; //前驱指针
```

Public operation:

```
Node() = default;
Node(int number, std::string name, std::string sex, int age, std::string type); //初始化该节点
~Node();

Node *insertAsPred(Node *t); //紧接着该节点后面 插入新节点
Node *insertAsNext(Node *t); //紧接着这个节点前面 插入新的
void getInform(); //输出该节点（考生）的信息
void setInform(); //修改该节点（考生）的信息
int getNumber() { return this->_number; } //返回该节点（考生）的学号
```

2. LinkList

Private members:

```
int _size; //链表的规模
Node *header; //头哨兵
Node *trailer; //尾哨兵
```

Public operation:

```
LinkList();
~LinkList();

Node *insertAtPos(int pos); //在某个位置插入
Node *insertAsLast(Node *t); //在链表的最后插入
void remove(int num); //按考号删除考生位置
Node *find(int num); //按学号查找 返回对象
Node *change(int num); //按学号修改内容
void traverse(); //统计
```

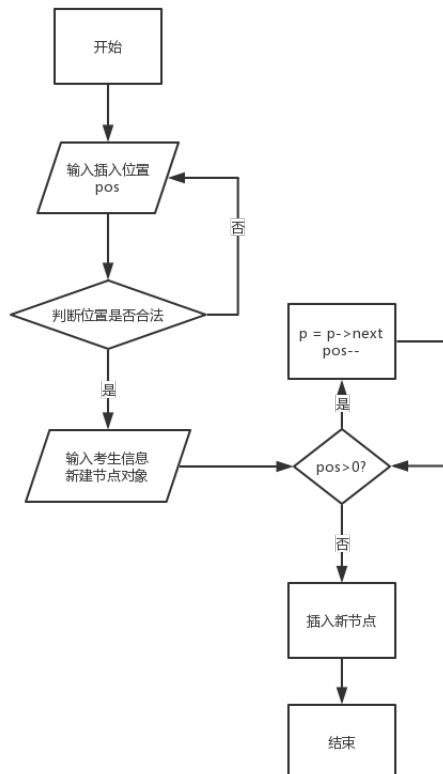
2.4 System design

When the system is first used, the user will be prompted to enter the number of candidates, and then enter the series information of the candidate, and use this information to complete the initialization of the linked list. Then, according to the operation code input by the user, the corresponding operation is implemented, and the prompt when the operation code is not standardized is given.s。

3. Realization

3.1 Realization of the insert function

1. Insert function flow chart



2. Insert function core code

```
int pos;
std::cout << "请输入你要插入的考生的位置: ";
std::cin >> pos;
if (nameList->insertAtPos(pos) != nullptr) {
    nameList->traverse();
}
break;
```

```
Node *LinkList::insertAtPos(int pos) {

    auto temp = this->header; //头哨兵
    while (--pos) {
        temp = temp->next; //现在要在他的后面插入
```

```

        if (temp == this->trailer || nullptr) {
            std::cout << "您输入的位置有误，请重新输入\n";
            return nullptr;
        }
    }

    std::cout << "请依次输入考生的考号、姓名、性别、年龄及报考类别\n";
    int number;
    std::string name;//姓名
    std::string sex;//性别
    int age;//年龄
    std::string type;//报考科目
    std::cin >> number >> name >> sex >> age >> type;
    auto stu = new Node(number, name, sex, age, type);
    _size++;
    temp->insertAsNext(stu);
    return stu;
}

```

3. Insert function diagram

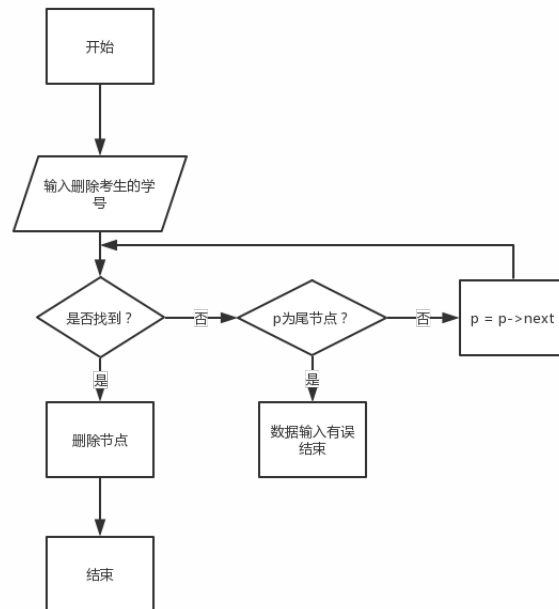
```

首先建立考生信息系统
首先输入考生人数： 3
请依次输入考生的考号、姓名、性别、年龄及报考类别
1 stu1 男 18 科目一
2 stu2 男 22 科目二
3 stu3 女 20 科目三
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，6为退出，0取消操作)：
请选择您要进行的操作： 1
请输入您要插入的考生的位置： 3
请依次输入考生的考号、姓名、性别、年龄及报考类别
4 stu4 女 70 挂科
1 stu1 男 18 科目一
2 stu2 男 22 科目二
4 stu4 女 70 挂科
3 stu3 女 20 科目三
请选择您要进行的操作： 1
请输入您要插入的考生的位置： 7
您输入的位置有误，请重新输入
请选择您要进行的操作：

```


3.2 Realization of the delete function

1. Delete function flow chart



2. Delete function core code

```
cout << "请输入要删除的考生的考号:";
cin >> number;
nameList->remove(number);
nameList->traverse();
```

```
void LinkedList::remove(int num) {
    auto p = this->header->next;
    while (p != this->trailer && p->getNumber() != num) {
        p = p->next;
    }
    if (p->getNumber() == num) {
        std::cout << "您要删除的考生信息是: \n";
        p->getInform();
        std::cout << '\n';
        p->pre->next = p->next;
        p->next->pre = p->pre;
        //delete p;
        _size--;
    } else {
        std::cout << "您的输入有误\n";
    }
}
```

```
}  
}
```

3. Delete function diagram

首先建立考生信息系统

首先输入考生人数： 3

请依次输入考生的考号、姓名、性别、年龄及报考类别

1 stu1 男 18 科目一

2 stu2 男 22 科目二

3 stu3 女 20 科目三

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, 6为退出, 0取消操作) :

请选择您要进行的操作： 2

请输入要删除的考生的考号：2

您要删除的考生信息是：

2 stu2 男 22 科目二

1 stu1 男 18 科目一

3 stu3 女 20 科目三

请选择您要进行的操作： 2

请输入要删除的考生的考号：2

您的输入有误

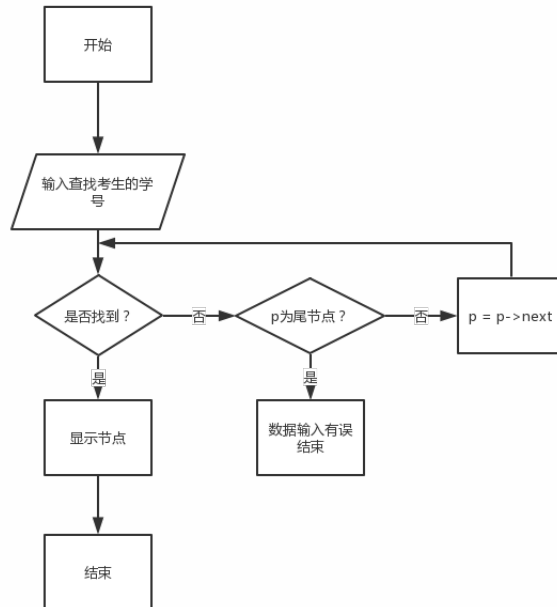
1 stu1 男 18 科目一

3 stu3 女 20 科目三

请选择您要进行的操作：

3.3 Realization of the search function

1. Find function flow chart



2. Find function core code

```
cout << "请输入您要查找的考生的考号: ";
cin >> number;
nameList->find(number);
```

```
Node *LinkList::find(int num) {
    auto p = this->header->next;
    while (p != this->trailer && p->getNumber() != num) {
        p = p->next;
    }
    if (p->getNumber() == num) {
        p->getInform();
        return p;
    } else {
        std::cout << "您输入的信息有误\n";
        return nullptr;
    }
}
```

3. Find function diagram

首先建立考生信息系统

首先输入考生人数： 3

请依次输入考生的考号、姓名、性别、年龄及报考类别

1 stu1 男 18 科目一

2 stu2 男 22 科目二

3 stu3 女 20 科目三

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, 6为退出, 0取消操作) :

请选择您要进行的操作： 3

请输入您要查找的考生的考号： 3

3 stu3 女 20 科目三

请选择您要进行的操作： 3

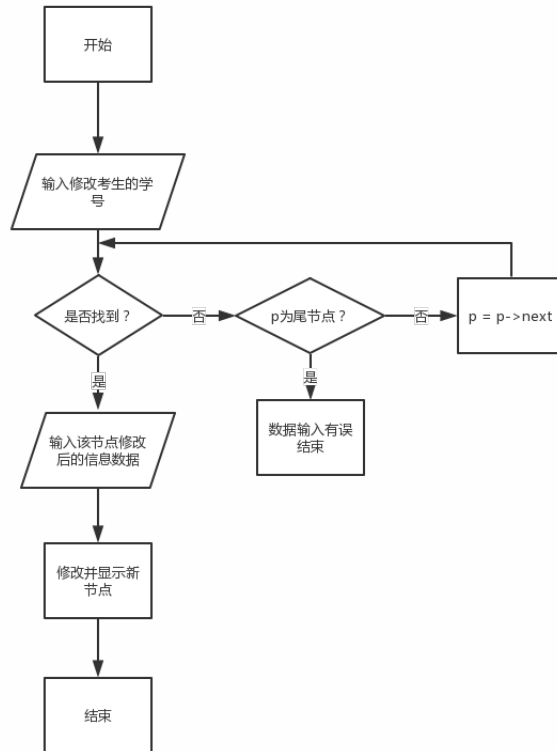
请输入您要查找的考生的考号： 8

您输入的信息有误

请选择您要进行的操作：

3.4 Realization of the modify function

1. Modify function flow chart



2. Modify function core code

```
cout << "请输入您要修改的考生的考号: ";
cin >> number;
if (nameList->change(number) != nullptr) {
    nameList->traverse();
}
```

```
Node *LinkList::change(int num) {
    auto p = this->header->next;
    while (p != this->trailer && p->getNumber() != num) {
        p = p->next;
    }
    if (p->getNumber() == num) {
        p->getInform();
        std::cout << "请依次输入修改之后的学号 姓名 性别 年龄 报考类别: \n";
        p->setInform();
        return p;
    }
}
```

```
    } else {  
        std::cout << "您输入的信息有误\n";  
        return nullptr;  
    }  
}
```

3. Modify function diagram

首先建立考生信息系统

首先输入考生人数： 3

请依次输入考生的考号、姓名、性别、年龄及报考类别

1 stu1 男 18 科目一

2 stu2 男 22 科目二

3 stu3 女 20 科目三

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, 6为退出, 0取消操作) :

请选择您要进行的操作： 4

请输入您要修改的考生的考号： 2

2 stu2 男 22 科目二

请依次输入修改之后的学号 姓名 性别 年龄 报考类别：

4 stu4 女 27 挂科

1 stu1 男 18 科目一

4 stu4 女 27 挂科

3 stu3 女 20 科目三

请选择您要进行的操作： 4

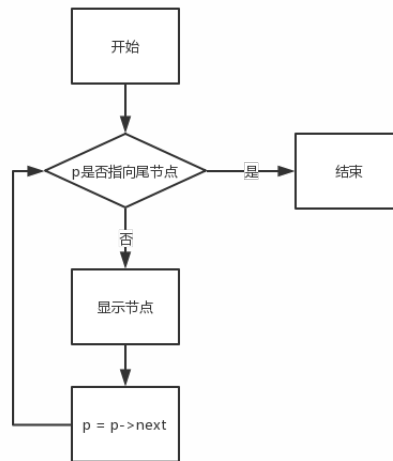
请输入您要修改的考生的考号： 2

您输入的信息有误

请选择您要进行的操作：

3.5 Realization of statistical functions.

1. Statistical function flow chart



2. Core code of statistical function

```
nameList->traverse();
```

```
void LinkedList::traverse() {  
    auto p = this->header->next;  
    for (; p != this->trailer; p = p->next) {  
        p->getInform();  
    }  
}
```

3. Schematic diagram of statistical function

首先建立考生信息系统

首先输入考生人数： 3

请依次输入考生的考号、姓名、性别、年龄及报考类别

1 stu1 男 18 科目一

2 stu2 男 22 科目二

3 stu3 女 20 科目三

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, 6为退出, 0取消操作) :

请选择您要进行的操作： 5

1 stu1 男 18 科目一

2 stu2 男 22 科目二

3 stu3 女 20 科目三

请选择您要进行的操作： 1

请输入你要插入的考生的位置： 3

请依次输入考生的考号、姓名、性别、年龄及报考类别

4 stu4 女 23 科目四

1 stu1 男 18 科目一

2 stu2 男 22 科目二

4 stu4 女 23 科目四

3 stu3 女 20 科目三

请选择您要进行的操作： 5

1 stu1 男 18 科目一

2 stu2 男 22 科目二

4 stu4 女 23 科目四

3 stu3 女 20 科目三

请选择您要进行的操作：

3.6 Realization of exit operation

1. Exit operation core code

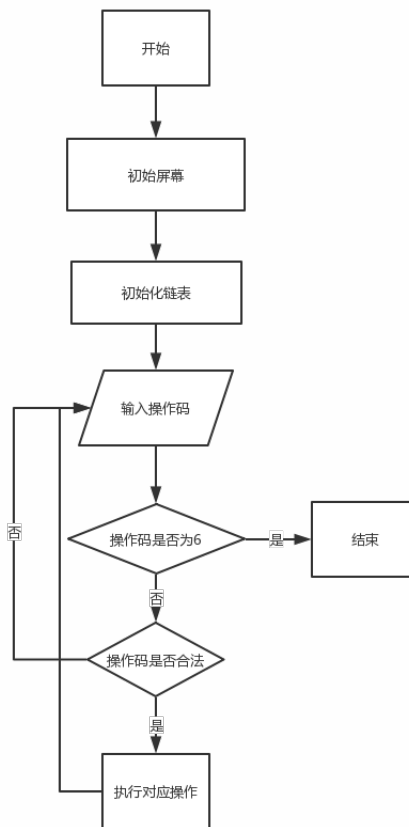
```
if (op == 6) {  
    cout << "您已退出系统。";  
    break;  
}
```

2. Schematic diagram of exit operation function

```
首先建立考生信息系统  
首先输入考生人数： 3  
请依次输入考生的考号、姓名、性别、年龄及报考类别  
1 stu1 男 18 科目一  
2 stu2 男 22 科目二  
3 stu3 女 20 科目三  
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，6为退出，0取消操作）：  
请选择您要进行的操作： 6  
您已退出系统。
```

3.7 Realization of overall system

1. Overall system flow chart



2. Core code for the overall system

```
int op = 0;
while (1) {
    cout << "请选择您要进行的操作： ";
    cin >> op;
    if (op == 6) {
        cout << "您已退出系统。";
        break;
    }
    switch (op) {
        case 0: {
            break;
        }
        case 1: {
            int pos;
            std::cout << "请输入你要插入的考生的位置： ";
```

```

        std::cin >> pos;
        if (nameList->insertAtPos(pos) != nullptr) {
            nameList->traverse();
        }
        break;
    }
    case 2: {
        cout << "请输入要删除的考生的考号:";
        cin >> number;
        nameList->remove(number);
        nameList->traverse();
        break;
    }
    case 3: {
        cout << "请输入您要查找的考生的考号: ";
        cin >> number;
        nameList->find(number);
        break;
    }
    case 4: {
        cout << "请输入您要修改的考生的考号: ";
        cin >> number;
        if (nameList->change(number) != nullptr) {
            nameList->traverse();
        }
        break;
    }
    case 5: {
        nameList->traverse();
        break;
    }
    default: {
        cout << "您输入的数字有误, 请重新输入: ";
        break;
    }
}
}

```

3. Schematic diagram of the overall system

首先建立考生信息系统

首先输入考生人数： 3

请依次输入考生的考号、姓名、性别、年龄及报考类别

1 stu1 男 18 科目一

2 stu2 男 22 科目二

3 stu3 女 19 科目三

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，6为退出，0取消操作）：

请选择您要进行的操作： 1

请输入您要插入的考生的位置： 4

请依次输入考生的考号、姓名、性别、年龄及报考类别

4 stu4 女 25 科目四

1 stu1 男 18 科目一

2 stu2 男 22 科目二

3 stu3 女 19 科目三

4 stu4 女 25 科目四

请选择您要进行的操作： 3

请输入您要查找的考生的考号： 2

2 stu2 男 22 科目二

请选择您要进行的操作： 2

请输入要删除的考生的考号：3

您要删除的考生信息是：

3 stu3 女 19 科目三

1 stu1 男 18 科目一

2 stu2 男 22 科目二

4 stu4 女 25 科目四

请选择您要进行的操作： 4

请输入您要修改的考生的考号： 2

2 stu2 男 22 科目二

请依次输入修改之后的学号 姓名 性别 年龄 报考类别：

3 stu3 女 19 科目三

1 stu1 男 18 科目一

3 stu3 女 19 科目三

4 stu4 女 25 科目四

请选择您要进行的操作： 6

您已退出系统。

4. Test

4.1 Function testing

1. Insert function test

Test case:

3 stu3 男22 网络工程师

Expected outcome:

1 stu1 男20 软件开发师

2 stu2 女21 软件测试员

3 stu3 男22 网络工程师

Experimental result:

```
首先建立考生信息系统
首先输入考生人数： 2
请依次输入考生的考号、姓名、性别、年龄及报考类别
1 stu1 男 20 软件开发师
2 stu2 女 21 软件测试员
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，6为退出，0取消操作)：
请选择您要进行的操作： 1
请输入你要插入的考生的位置： 3
请依次输入考生的考号、姓名、性别、年龄及报考类别
3 stu3 男 22 网络工程师
1 stu1 男 20 软件开发师
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
```

2. Delete function test

Test case:

Remove candidates with a test number of 4

Expected outcome:

1 stu1 男20 软件开发师

2 stu2 女21 软件测试员

3 stu3 男22 网络工程师

Experimental result:

```
首先建立考生信息系统
首先输入考生人数： 4
请依次输入考生的考号、姓名、性别、年龄及报考类别
1 stu1 男 20 软件开发师
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
4 stu4 男 25 软件架构师
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，6为退出，0取消操作)：
请选择您要进行的操作： 2
请输入要删除的考生的考号：4
您要删除的考生信息是：
4 stu4 男 25 软件架构师

1 stu1 男 20 软件开发师
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
```

3. Find function test

Test case:

Find candidates with a test number of 2

Expected outcome:

2 stu2 女21 软件测试员

Experimental result:

```
请依次输入考生的考号、姓名、性别、年龄及报考类别
1 stu1 男 20 软件开发师
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
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，6为退出，0取消操作)：
请选择您要进行的操作： 3
请输入您要查找的考生的考号： 2
2 stu2 女 21 软件测试员
```

4. Modify function test

Test case:

Change the test number 1 to a gender female, age 20, and apply for a mobile developer.

Expected outcome:

1 stu1 女 20 移动开发员

Experimental result:

```
请选择您要进行的操作： 4
请输入您要修改的考生的考号： 1
1 stu1 男 20 软件开发师
请依次输入修改之后的学号 姓名 性别 年龄 报考类别：
1 stu1 女 20 移动开发员
1 stu1 女 20 移动开发员
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
```

5. Statistical function test

Test case:

Statistics current data

Expected outcome:

1 stu1 女 20 移动开发员

2 stu2 女 21 软件测试员

3 stu3 男 22 网络工程师

Experimental result:

```
请选择您要进行的操作： 5
1 stu1 女 20 移动开发员
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
```

4.2 Boundary test

1. Initialize no input data

Test case:

Initialize no input data

Expected outcome:

Give an error message and the program will run normally without crashing.

Experimental result:

```
首先建立考生信息系统
首先输入考生人数： 0
请输入一个正整数： |
```

2. Delete header node

Test case:

Delete header node

Expected outcome:

Give an error message and the program will run normally without crashing.

Experimental result:

```
请依次输入考生的考号、姓名、性别、年龄及报考类别
1 stu1 男 20 软件开发师
2 stu2 女 21 软件测试员
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, 6为退出, 0取消操作) :
请选择您要进行的操作: 2
请输入要删除的考生的考号: 1
您要删除的考生信息是:
1 stu1 男 20 软件开发师

2 stu2 女 21 软件测试员
请选择您要进行的操作:
```

3. After the deletion, the linked list is empty.

Test case:

Delete the front list only one node, delete the linked list is empty

Expected outcome:

Give an error message and the program will run normally without crashing.

Experimental result:

```
2 stu2 女 21 软件测试员
请选择您要进行的操作: 2
请输入要删除的考生的考号: 2
您要删除的考生信息是:
2 stu2 女 21 软件测试员
请选择您要进行的操作:
```

4.3 Error test

1. The number of candidates is wrong

Test case:

The number of candidates entered is negative

Expected outcome:

Give an error message and the program will run normally without crashing.

Experimental result:

```
首先建立考生信息系统
首先输入考生人数： -2
请输入一个正整数：
```

2. Opcode error

Test case:

Wrong input code

Expected outcome:

Give an error message and the program will run normally without crashing.

Experimental result:

```
请依次输入考生的考号、姓名、性别、年龄及报考类别
1 stu1 女 20 移动开发员
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
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, 6为退出, 0取消操作) :
请选择您要进行的操作： 7
您输入的数字有误, 请重新输入： 请选择您要进行的操作： |
```

3. Insert position does not exist

Test case:

Insert information into the 5th position when there are only 3 candidates in the list.

Expected outcome:

Give an error message and the program will run normally without crashing.

Experimental result:

```
1 stu1 女 20 移动开发员
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
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, 6为退出, 0取消操作) :
请选择您要进行的操作: 1
请输入您要插入的考生的位置: 5
您输入的位置有误, 请重新输入
```

4. Delete test number does not exist

Test case:

Delete test number does not exist

Expected outcome:

Give an error message and the program will run normally without crashing.

Experimental result:

```
请选择您要进行的操作: 2
请输入要删除的考生的考号: 5
您的输入有误
1 stu1 女 20 移动开发员
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
请选择您要进行的操作:
```

5. Find the test number does not exist

Test case:

Find the test number does not exist

Expected outcome:

Give an error message and the program will run normally without crashing.

Experimental result:

```
1 stu1 女 20 移动开发员
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
请选择您要进行的操作: 3
请输入您要查找的考生的考号: 6
您输入的信息有误
请选择您要进行的操作:
```

6. Modify the test number does not exist

Test case:

Modify the test number does not exist

Expected outcome:

Give an error message and the program will run normally without crashing.

Experimental result:

```
1 stu1 女 20 移动开发员
2 stu2 女 21 软件测试员
3 stu3 男 22 网络工程师
请选择您要进行的操作： 4
请输入您要修改的考生的考号： 4
您输入的信息有误
请选择您要进行的操作：
```