

---

# 数据结构课程设计

## ——考试报名系统

作者姓名：\_\_\_\_\_郑超\_\_\_\_\_

学        号：\_\_\_\_\_1751747\_\_\_\_\_

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

<b>1 项目功能分析</b>	<b>2</b>
<b>2 程序设计及实现</b>	<b>2</b>
2.1 总述	2
2.2 操作引导函数	3
2.3 考试报名系统类等数据结构	3
2.4 考试报名系统类相关操作函数	4
<b>3 各功能运行测试</b>	<b>8</b>
3.1 测试数据	8
3.2 测试结果	8

## 1 项目功能分析



考试报名系统功能分析结构图

考试报名系统应实现以下几个基本操作：（一）系统操作；（二）增删查改操作；（三）查看报名信息操作。系统操作包括进、出系统两个部分；用户操作包括增删查改考生信息，以及查看报名信息两个操作类型。

## 2 程序设计及实现

### 2.1 总述

程序包括三个部分：

- （一）操作指引的函数，提示输入对应的字母进行不同的操作
- （二）储存考试报名系统信息的实例类。
- （三）实例类实现对数据的增删查改等相关操作。

## 2.2 操作指引函数

```
94  +int menu() { ... }
```

```
7  enum
8  {
9      CANCEL,    //取消操作
10     INSERT,    //插入
11     DELETE,    //删除
12     SEARCH,    //查找
13     MODIFY,    //修改
14     CALCULATE  //统计
15 };
```

```
while (flag)
{
    int Construction = menu();
    switch (Construction)
    {
        case(CANCEL):
            { ... }
        case(INSERT):
            { ... }
        case(DELETE):
            { ... }
        case(SEARCH):
            { ... }
        case(CALCULATE):
            { ... }
        case(MODIFY):
            { ... }
    }
}
```

menu()主要功能是展现字母和操作的对应关系，提示使用者输入某字母以选择想要的功能，并将该值作为函数的返回值。

利用枚举 enum 实现数字和操作的对应关系，便于查看。

## 2.3 考试报名系统类等数据结构

```
struct Member
{
    int number;
    string name;
    string sex;
    int age;
    string occupation;
    Member* next;

    Member() { ... }
    Member(int _number, str
```

```
31 class ExamSystem
32 {
33 public:
34     ExamSystem();
35     ~ExamSystem();
36
37     //insert member
38     void Insert(Member* newNode, int pos);
39
40     //delete member
41     void Delete(int number);
42     void Delete(string name);
43
44     //find member
45     Member* Find(int candidate_number);
46     Member* Find(string name);
47
48     //change member information
49     void Modify(int candidate_number, Member* newNode);
50
51     //print student's infomation
52     void print_stu_ifo(Member* p);
53     void print_all();
54
55 private:
56     Member* first; //表头节点
57     int total = 0;
58
59     Member* find(int pos);
60     Member* find_pre_withNAME(string name);
61     Member* find_pre_withNUMBER(int number);
62
63 };
```

Member 是一个结构体，用于表示某个考生的信息。包括学号、姓名、性别、年龄、职位。本程序设计中主要使用链表结构来保存考生信息，因此 Member 还需带有一个 Member\* next 指针，指向下一个存储下一个考生信息的结构体。Member 设置了两个构造函数，一个是默认的不带参数的构造函数，另一个可以直接依次输入考生信息，构造出一个 Member 的结构体。

ExamSystem 类包括两个基本数据，其一是 Member 的表头节点 Member\* first；其二是 int 类型的 total，用于保存当前报名系统中的考生人数。

## 2.4 考试报名系统类相关操作函数

```

37      //insert member
38      void Insert(Member* newNode, int pos);
39
40      //delete member
41      void Delete(int number);
42      void Delete(string name);
43
44      //find member
45      Member* Find(int candidate_number);
46      Member* Find(string name);
47
48      //change member information
49      void Modify(int candidate_number, Member* newNode);
50
51      //print student's infomation
52      void print_stu_ifo(Member* p);
53      void print_all();
    
```

考试报名系统主要实现的功能有：（一）增删查改操作；（二）查看报名信息操作。

**插入功能：**

```
void ExamSystem::Insert(Member* newNode, int pos)
```

```

27 void ExamSystem::Insert(Member* newNode, int pos)
28 {
29     //如果信息输入错误 newNode为空
30     if (!newNode) return;
31     //考生的考号不允许重复
32     //apointed position is out of range
33     if (pos > total || pos < 0) { ... }
34     //插入
35     //如果在第一个元素的位置插入
36     if (pos == 0) { ... }
37     //在末尾位置插入
38     else if (pos == total) { ... }
39     //在中间插入
40     else { ... }
41     total++; //change the total number
42     return;
43 }

```

插入时先进行错误判断：（一）检查 newNode 是否成功初始化；（二）检查考生考号是否唯一；（三）检查考生插入位置是否合理。

如果判断错误，则直接返回，不进行其他操作；反之若均正确，则根据链表插入的操作，将 newNode 插入到存储考生信息的链表中，并修改 total 的值。

### 删除操作：

```
void ExamSystem::Delete(int candidate_number)
```

```
void ExamSystem::Delete(string name)
```

```

65 void ExamSystem::Delete(int candidate_number)
66 {
67     Member* preDel = find_pre_withNUMBER(candidate_number);
68     Member* del = preDel->next;
69     preDel->next = del->next;
70
71     //CARVEAT!!!
72     cout << "你删除的考生信息是：";
73     print_stu_ifo(del);
74
75     delete del;
76
77     //change the total number
78     total--;
79     return;
80 }

```

```

82  void ExamSystem::Delete(string name)
83  {
84      Member* preDel = find_pre_withNAME(name);
85      Member* del = preDel->next;
86      preDel->next = del->next;
87      delete del;
88
89      //change the total number
90      total--;
91      return;
92  }
93

```

删除操作有两个：（一）删除对应考号的考生；（二）删除对应姓名的考生。

删除时，调用 find 函数找到符合要求考生的前一个考生，依据链表的相关操作进行节点的删除，并修改 total 的值。

### 查找操作：

```
Member* ExamSystem::Find(int candidate_number)
```

```
Member* ExamSystem::Find(string name)
```

<pre> 114  Member* ExamSystem::Find(string name) 115  { 116      //point to ZERO 117      Member* iter = first; 118      while (iter-&gt;next) 119      { 120          iter = iter-&gt;next; 121          if (iter-&gt;name == name) 122          { 123              return iter; 124          } 125      } 126      cout &lt;&lt; "错误! 未找到指定名称的考生! "; 127      return nullptr; 128  } </pre>	<pre> 98  Member* ExamSystem::Find(int candidate_number) 99  { 100      //point to ZERO 101      Member* iter = first; 102      while (iter-&gt;next) 103      { 104          iter = iter-&gt;next; 105          if (iter-&gt;number == candidate_number) 106          { 107              return iter; 108          } 109      } 110      cout &lt;&lt; "错误! 未找到指定考号的考生! "; 111      return nullptr; 112  } </pre>
---	--

进行查找操作时，从表头节点开始遍历整个链表，若找到符合要求的考生，则返回对应的 Member 指针。若遍历完整个链表依然没有符合要求的考生，则报错，并返回一个空指针。



## 修改操作

`void ExamSystem::Modify(int candidate_number, Member* newNode)`

```

134 void ExamSystem::Modify(int candidate_number, Member* newNode)
135 {
136     Member* pre = find_pre_withNUMBER(candidate_number);
137     Member* target = pre->next;
138     pre->next = newNode;
139     newNode->next = target->next;
140     delete target;
141
142     return;
143 }
    
```

修改操作的逻辑是，根据需要修改的信息直接创建一个新的考生。通过考生的考号查找到对应考生的前一个考生，把旧的信息用新的信息替换掉，即删除原本的考生，添加一个新的考生。

## 统计操作

`void ExamSystem::print_stu_ifo(Member* p)`

`void ExamSystem::print_all()`

```

148 void ExamSystem::print_stu_ifo(Member* p)
149 {
150     cout << p->number << '\t'
151         << p->name << '\t'
152         << p->sex << '\t'
153         << p->age << '\t'
154         << p->occupation << '\n';
155     return;
156 }
157
158 void ExamSystem::print_all()
159 {
160     cout << "考号\t姓名\t性别\t年龄\t报考类别\t\n";
161
162     Member* iter = first;
163     while (iter->next)
164     {
165         iter = iter->next;
166         print_stu_ifo(iter);
167     }
168
169     return;
170 }
    
```

统计操作实质上就是从头遍历整个 Member 链表，根据格式输出所有考生的信息。



### 3 各功能运行测试

#### 3.1 测试数据

```

3
1 赵 男 10 语文
2 钱 女 11 数学
3 孙 男 12 英语
1
4
4 李 女 13 物理
5
1
3
3 周 男 20 生物
5
2
3
5
4
1
5 吴 女 15 化学
5
0
    
```

#### 3.2 测试结果

初始化：建立考生信息系统

```

首先请建立考生信息系统！
请输入考生人数:3
请依次输入要插入的考生的考号，姓名，性别，年龄，及报考类别！
1 赵 男 10 语文
2 钱 女 11 数学
3 孙 男 12 英语
    
```

尾插入操作、统计操作

```

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）
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
请依次输入要插入的考生的考号，姓名，性别，年龄，及报考类别！
4 李 女 13 物理
    
```

```

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)
请选择您要进行的操作: 5
考号      姓名      性别      年龄      报考类别
1         赵      男      10      语文
2         钱      女      11      数学
3         孙      男      12      英语
4         李      女      13      物理
    
```

### 中间插入操作

```

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)
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
请依次输入要插入的考生的考号, 姓名, 性别, 年龄, 及报考类别!
3 周 男 20 生物

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)
请选择您要进行的操作: 5
考号      姓名      性别      年龄      报考类别
1         赵      男      10      语文
2         钱      女      11      数学
3         周      男      20      生物
3         孙      男      12      英语
4         李      女      13      物理
    
```

### 删除操作

```

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)
请选择您要进行的操作: 2
请输入要删除的考生的考号: 3
你删除的考生信息是: 3 周 男 20 生物

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)
请选择您要进行的操作: 5
考号      姓名      性别      年龄      报考类别
1         赵      男      10      语文
2         钱      女      11      数学
3         孙      男      12      英语
4         李      女      13      物理
    
```

### 修改操作

```

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)
请选择您要进行的操作: 4
请输入要修改的考生的考号: 1
请依次输入要修改的考生的考号, 姓名, 性别, 年龄, 及报考类别!
5 吴 女 15 化学

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)
请选择您要进行的操作: 5
考号      姓名      性别      年龄      报考类别
5         吴      女      15      化学
2         钱      女      11      数学
3         孙      男      12      英语
4         李      女      13      物理
    
```

### 退出操作

```

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)
请选择您要进行的操作: 0
感谢使用本系统! 请按任意键继续. . .
    
```