

# 项目说明文档

## 数据结构课程设计

### ——考试报名系统

作者姓名：\_\_\_\_\_沈星宇\_\_\_\_\_

学        号：\_\_\_\_\_1951576\_\_\_\_\_

指导教师：\_\_\_\_\_张颖\_\_\_\_\_

学院、专业：\_\_\_\_\_软件学院 软件工程\_\_\_\_\_

同济大学

Tongji University

## 一、 分析

### (1) 应用背景

考试报名工作涉及考生的一些个人信息的匹配,并且要涉及到对信息的一些基本处理功能。不仅是考试报名工作,类似的信息存储和处理还有很多应用场景。

本项目是对考试报名管理的简单模拟,用控制台选项的方式完成输入考生信息、输出考生信息、查询考生信息、添加考生信息、删除考生信息。

### (2) 项目功能要求

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、报考类别等信息。由于考生信息的数据类型各有不同,又最好能用一个数据结构将其包括在一起,再加上对数据的操作有频繁的增删,所以选用较灵活的链表来实现。

## 二、 设计

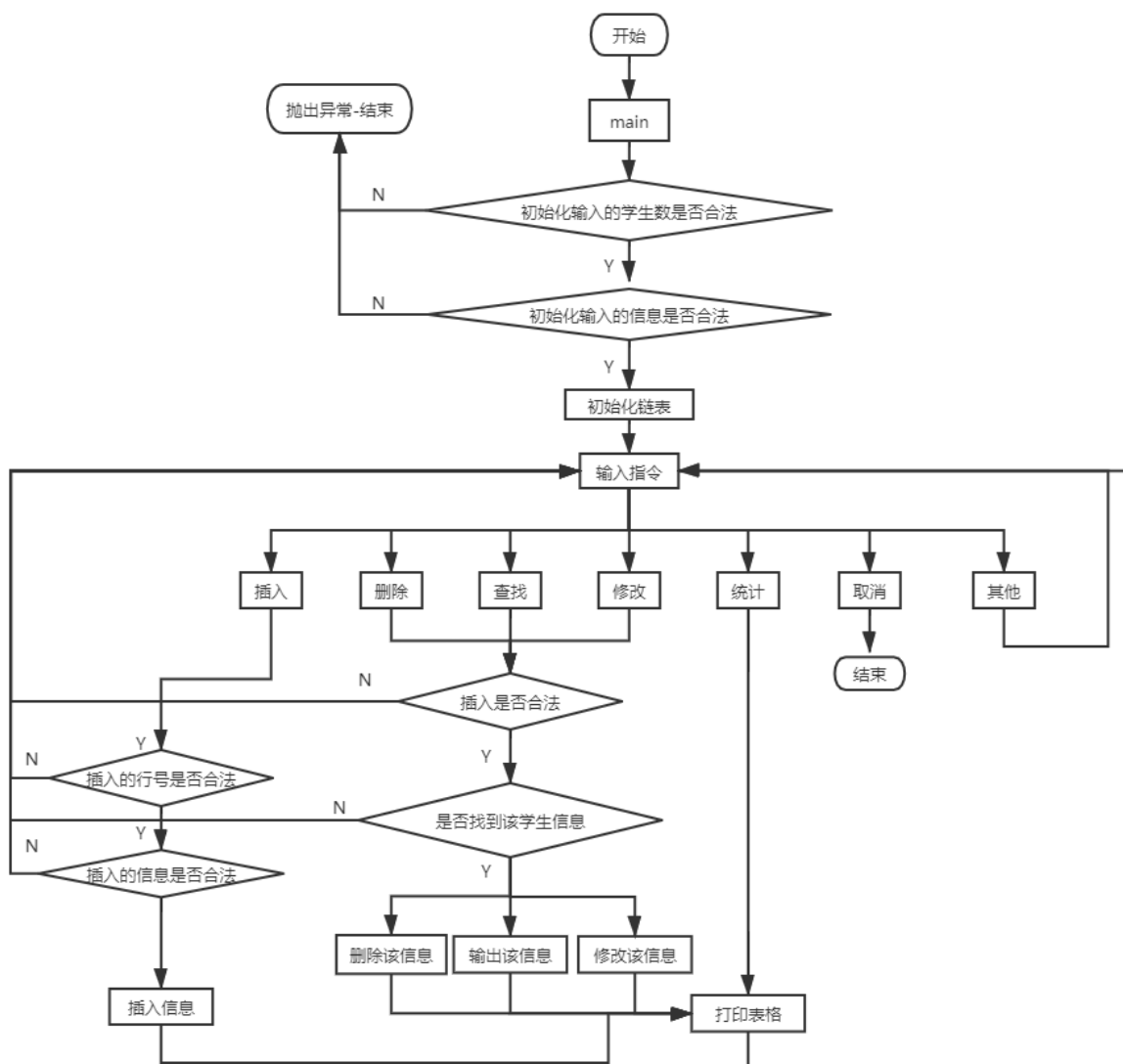
### (1) 数据结构设计

```
//最大长度
#define MAXSIZE_NAME 10
#define MAXSIZE_TYPE 20
#define MAXSIZE_GENDER 5
#define MAXSIZE_NUMBER 20
#define MAXSIZE_AGE 4

//链表
typedef struct listPoint {
    struct listPoint* next;
    unsigned int number;           //学号
    char name[MAXSIZE_NAME];      //姓名
    char gender[MAXSIZE_GENDER];  //性别
    unsigned int age;              //年龄
    char type[MAXSIZE_TYPE];      //报考类型
}listPoint,*linkList;
```

- 1、用宏定义的方式设置数组的最大长度,便于修改,增强代码可读性。
- 2、用 typedef 将 struct listPoint 定义 listPoint 和\*linkList 两个别名,前者强调是链表的结点,后者强调是一个单链表,增强了代码的可读性。

### (2) 程序流程设计



- 1、首先对表格进行初始化，初始输入一些数据，此处对合法性进行判断，如果输入不合法就直接结束程序；输入合法数据后会生成链表的头结点，对链表进行初始化。
- 2、接下来进入循环，直到用户输入取消的指令结束程序。在进入对应的操作时，都会对输入数据的合法性进行检测，如果不合法则需重新输入指令；如果合法，就去搜索对应的学生信息执行相应的操作。
- 3、每一次操作结束之后都会打印最新的表格。

### 三、 实现

#### 1、main()函数的内部逻辑

main()函数会首先对考试报名系统进行初始化，此处要进行合法性检测

- (1) 用 `changeCharToInt()` 来对输入的合法性进行判断，确保如果用户输入负数、字母都能够进行提示，不调用接下来的步骤引发错误
- (2) 对输入的学生数为正数和 0 的情况进行区分
- (3) 创建单链表，对头结点进行初始化置空，并且插入输入的学生信息

```
//-----初始化链表-----
length = 0;
std::cout << "首先请建立考生信息系统! \n" << "请输入考生人数: ";
unsigned int num = 0; //学生人数
char numChar[MAXSIZE_NUM];
memset(numChar, 0, sizeof(numChar));
std::cin >> numChar;
if (!changeCharToInt(numChar, num)) {
    std::cout << "输入的学生人数不合法\n";
    return -1;
}
if (num > 0) {
    std::cout << "请依次输入考生的考号、姓名、性别、年龄及报考类别! (用空格隔开)\n";
}
linkList L; //一个指向单链表的指针
init(L);
initInsert(L, num);
```

用 switch-case 来判断要执行的命令

### (1) 插入操作

- 1、首先用 changeCharToInt()检查用户的输入是否合法，确保如果用户输入负数、字母都能够进行提示，不调用接下来的步骤引发错误
- 2、然后对用户输入的位置进行合法性检查，例如表内已有 n 行信息时，不允许在 n+2 行及以后进行插入操作
- 3、然后对输入的学号和年龄进行合法性检查，通过调用函数 changeCharToInt()来确保学号和年龄不是字符串
- 4、通过一切合法性检测就调用插入函数 insertInfo()

```
//-----插入信息-----
case COMMAND_INSERT:
    std::cout << "请输入您要插入的考生的位置:";
    std::cin >> locationChar;
    if (!changeCharToInt(locationChar, location)) {
        std::cout << "输入的位置不合法! \n";
        continue;
    }
    if (location < 1 || location > length + 1) {
        std::cout << "输入的位置不合法! \n";
        continue;
    }
    else {
        std::cout << "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! \n";
        std::cin >> numberChar >> name >> gender >> ageChar >> type;
        if (!changeCharToInt(numberChar, number) || !changeCharToInt(ageChar, age)) {
            std::cout << "输入的信息不合法\n";
        }
        else {
            insertInfo(L, location, number, name, gender, age, type);
        }
        sumInfo(L);
    }
    break;
```

### (2) 删除/查找/修改操作

- 1、首先对输入考号的合法性进行检测，确保如果用户输入负数、字母都能够进行提示，不调用接下来的步骤引发错误
- 2、然后调用函数 deleteInfo()/findInfo()/rewriteInfo()

```

//-----删除信息-----
case COMMAND_DELETE:
    std::cout << "请输入您要删除的考生的考号: ";
    std::cin >> numberChar;
    if (!changeCharToInt(numberChar, number)) {
        std::cout << "输入的考号不合法! \n";
        continue;
    }
    deleteInfo(L, number);
    sumInfo(L);
    break;

```

## 2、changeCharToInt()的实现

用来判断输入的学号和年龄是否是纯数字，如果是则将其转化为整型返回。

```

bool changeCharToInt(char* string, unsigned int& num) {
    char* p = string;
    int s = 0;
    while (*p != '\0') {
        if (*p >= '0' && *p <= '9') {
            s = s * 10 + *p - '0';
            p++;
        }
        else {
            return false;
        }
    }
    num = s;
    return true;
}

```

## 3、initInsert()的实现

最初初始化插入操作时调用的函数，在用 cin 读取数据的时候，先将所有信息作为字符串读入，然后用 changeCharToInt()将读入的字符串类型的学号和年龄转化为整型，同时进行合法性检测。

```

//-----插入数据-----
for (unsigned int i = 1; i <= num; i++) {
    length++;
    std::cin >> numberChar >> name >> gender >> ageChar >> type;
    if (!changeCharToInt(numberChar, number)) {
        std::cout << "输入的学号不合法\n";
        return false;
    }
    if (!changeCharToInt(ageChar, age)) {
        std::cout << "输入的年龄不合法\n";
        return false;
    }
    insertInfo(L, i, number, name, gender, age, type);
}
return true;

```

## 4、insertInfo()的实现

为链表的每个结点分配内存，并将每一个结点用指针串起

```

bool insertInfo(linkList& L, unsigned int i, unsigned int number, char* name,
               char* gender, unsigned int age, char* type) {
    //-----内存分配-----
    listPoint* p;    //p指向当前扫描到的结点
    unsigned int locate = 0; //p指向的是第几个结点
    p = L;           //p指向头节点

    while (p != NULL && locate < i - 1) { //让p指向i-1个结点
        p = p->next;
        locate++;
    }
    listPoint* temp = (listPoint*)malloc(sizeof(listPoint));
    if (temp == NULL) {
        std::cout << "内存分配失败! \n";
        return false;
    }
    temp->number = number;
    temp->age = age;
    strcpy_s(temp->name, name);
    strcpy_s(temp->gender, gender);
    strcpy_s(temp->type, type);
    temp->next = p->next;
    p->next = temp;
    length++;
    return true;
}

```

## 5、deleteInfo()的实现

在删除之前会按照学号进行查找

```

bool deleteInfo(linkList& L, unsigned int number) {
    listPoint* p = L;
    listPoint* deleteInfo = NULL;
    while (1) {
        deleteInfo = p->next;
        if (deleteInfo == NULL) { //因为p要指向我们要寻找的结点的前一个结点
            std::cout << "没有查找到此人";
            return false;
        }
        else {
            if (deleteInfo->number == number) { //如果找到了
                p->next = deleteInfo->next;
                std::cout << "删除了考号为" << number << "的学生信息\n";
                free(deleteInfo);
                deleteInfo = NULL;
                return true;
            }
            else {
                p = p->next;
            }
        }
    }
}

```

## 6、findInfo()的实现

```

bool findInfo(linkList L, unsigned int number) {
    listPoint* p = L;
    while (1) {
        if (p == NULL) {
            std::cout << "查无此人\n";
            return false;
        }
        else {
            if (p->number == number) {
                std::cout << p->number << " " << p->name << " " << p->gender << " " << p->age << " " << p->type << "\n";
                return true;
            }
            else {
                p = p->next;
            }
        }
    }
}

```

## 7、rewriteInfo()的实现

```

bool rewriteInfo(linkList& L, unsigned int number) {
    listPoint* p = L;
    char name[MAXSIZE_NAME], type[MAXSIZE_TYPE], gender[MAXSIZE_GENDER];
    unsigned int num = 0, age = 0;
    while (1) {
        if (p == NULL) {
            std::cout << "查无此人\n";
            return false;
        }
        else {
            if (p->number == number) {
                std::cout << "请输入该同学新的考号、姓名、性别、年龄及报考类别\n";
                std::cin >> num >> name >> gender >> age >> type;
                p->number = num;
                p->age = age;
                strcpy_s(p->name, name);
                strcpy_s(p->gender, gender);
                strcpy_s(p->type, type);
                return true;
            }
            else {
                p = p->next;
            }
        }
    }
}

```

## 8、sumInfo()的实现

打印当前表格，每次操作后都会执行。

```

void sumInfo(linkList& L) {
    listPoint* p = L;
    std::cout << "\n\n考号 姓名 性别 年龄 报考类别\n";
    while (1) {
        p = p->next;
        if (p == NULL) {
            break;
        }
        std::cout << p->number << " " << p->name << " " << p->gender << " " << p->age << " " << p->type << "\n";
    }
}

```

## 9、destroyInfo()的实现

用于释放内存

```

bool destoryInfo(linkList& L) {
    listPoint* p = L;
    listPoint* freePointer = L;
    while (1) {
        if (p == NULL) {
            return true;
        }
        else {
            freePointer = p;
            p = p->next;
            free(freePointer);
        }
    }
}

```

## 四、 测试

### 1、 初始化表格的合法性检测

- (1) 输入负数、浮点数和字符结束进程
- (2) 输入人数为 0 则直接提示选择操作
- (3) 输入人数为正整数则提示输入学生信息

Microsoft Visual Studio 调试控制台

首先请建立考生信息系统！  
 请输入考生人数：-1  
 输入的学生人数不合法

D:\VS文件\数据结构课程设计\Project1\Debug\Project1.exe (进程 19108) 已退出，代码为 -1。  
 按任意键关闭此窗口。 . . .

Microsoft Visual Studio 调试控制台

首先请建立考生信息系统！  
 请输入考生人数：a  
 输入的学生人数不合法

D:\VS文件\数据结构课程设计\Project1\Debug\Project1.exe (进程 13376) 已退出，代码为 -1。  
 按任意键关闭此窗口。 . . .

Microsoft Visual Studio 调试控制台

首先请建立考生信息系统！  
 请输入考生人数：1.1  
 输入的学生人数不合法

D:\VS文件\数据结构课程设计\Project1\Debug\Project1.exe (进程 20852) 已退出，代码为 -1。  
 按任意键关闭此窗口。 . . .

D:\VS文件\数据结构课程设计\Project1\Debug\Project1.exe

首先请建立考生信息系统！  
 请输入考生人数：0

-----

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）



D:\VS文件\数据结构课程设计\Project1\Debug\Project1.exe

首先请建立考生信息系统！

请输入考生人数：1

请依次输入考生的考号、姓名、性别、年龄及报考类别！（用空格隔开）

1 stu1 女 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、插入操作

（1）输入负数、浮点数和字符回转到输入指令的位置

（2）输入学生信息的学号、年龄非纯数字时，会提示输入信息不合法

（3）输入人数为正整数则提示输入学生信息

（4）当表内有  $n$  条数据的时候，禁止在第  $n+2$  及之后的位置插入数据

1 stu1 女 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）

1  
请输入您要插入的考生的位置：-1  
输入的位置不合法！

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  
请输入您要插入的考生的位置：1.1  
输入的位置不合法！

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  
请输入您要插入的考生的位置：a  
输入的位置不合法！

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  
请输入您要插入的考生的位置：2  
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！  
2 stu2 男 年龄 软件开发师  
输入的信息不合法

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  
请输入您要插入的考生的位置：3  
输入的位置不合法！

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  
请输入您要插入的考生的位置：2  
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！  
2 stu2 男 21 软件开发师

考号	姓名	性别	年龄	报考类别
1	stu1	女	20	软件设计师
2	stu2	男	21	软件开发师

## 3、删除操作

（1）按照考号去查找需要删除的人员，如果没有查找到则打印一个报错

- (2) 同样会对输入数字的合法性进行检测
- (3) 查找到对应考号的学生之后就将他从列表删除

```

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
请输入您要删除的考生的考号：3
没有查找到此人

考号 姓名 性别 年龄 报考类别
1    stu1 女    20  软件设计师
2    stu2 男    21  软件开发师
-----

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
请输入您要删除的考生的考号：1
删除了考号为1的学生信息

考号 姓名 性别 年龄 报考类别
2    stu2 男    21  软件开发师
-----

```

4、查找操作

- (1) 会对输入数字的合法性进行检测
- (2) 如果没有查找到对应学号的同学则打印出“查无此人”的提示

```

1 stu1 女 20 软件设计师
2 stu2 男 21 软件开发师
-----

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
请输入要查找的考生的考号：0
查无此人
-----

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
请输入要查找的考生的考号：a
输入的考号不合法！
-----

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
请输入要查找的考生的考号：1
1    stu1 女    20  软件设计师
-----

```

5、修改操作

- (1) 会对输入数字的合法性进行检测
- (2) 如果没有查找到对应学号的同学则打印出“查无此人”的提示
- (3) 会对输入的新数据的学号和年龄进行合法性检测

```

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
请输入您要修改的考生的考号：5
查无此人

考号 姓名 性别 年龄 报考类别
1    stu1 女    20  软件设计师
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）
4
请输入您要修改的考生的考号：1
请输入该同学新的考号、姓名、性别、年龄及报考类别
1 stu1 女 20 咸鱼

考号 姓名 性别 年龄 报考类别
1    stu1 女    20  咸鱼
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）
4
请输入您要修改的考生的考号：1
请输入该同学新的考号、姓名、性别、年龄及报考类别
啊 啊 啊 啊 啊
输入的学号不合法

```

## 6、统计操作

（1）把目前的表格打印出来

```

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

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

```

## 7、结束输入

```

1 stu1 女 20 软件设计师
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）
0

D:\VS文件\数据结构课程设计\Project1\Debug\Project1.exe (进程 13104) 已退出，代码为 0。
按任意键关闭此窗口。 . . .

```