

东北大学秦皇岛分校

《C++与数据结构项目实训》

课程设计报告

物业基本信息管理系统

学 院	数 学 与 统 计 学 院
专 业	数 据 科 学 与 大 数 据 技 术
班级序号	23
学 号	202115110
姓 名	李 婷
指导教师	陆 杰、王 子 健
开始日期	2022 年 12 月 19 日
结束日期	2022 年 12 月 28 日

教师评阅意见书

一、态度

1. 工作态度 (☐认真、☐较好、☐一般、☐较差)
2. 出勤情况 (☐无缺勤、☐缺勤不超过1/3、☐缺勤超过1/3)
3. 上交时间 (☐按时、☐迟交1天、☐迟交1天以上)

二、格式规范

4. 文字部分 (☐符合规范、☐较符合规范、☐一般、☐不符合规范)
5. 图表部分 (☐符合规范、☐较符合规范、☐一般、☐不符合规范)
6. 数学公式 (☐符合规范、☐较符合规范、☐一般、☐不符合规范)
7. 参考文献 (☐符合规范、☐较符合规范、☐一般、☐不符合规范)

三、报告内容

8. 任务量和可行性 (☐合理、☐较为合理、☐一般、☐不合理)
9. 报告结构 (☐合理、☐较合理、☐一般、☐不合理)
10. 文字叙述 (☐清晰流畅、☐较为清晰、☐一般、☐不清晰)
11. 图表准确性 (☐准确、☐较准确、☐一般、☐不准确)

四、综合能力

12. 综合运用知识能力 (☐很强、☐较强、☐一般、☐较弱)
13. 实践与动手能力 (☐很强、☐较强、☐一般、☐较弱)
14. 创新意识 (☐很强、☐较强、☐一般、☐较弱)

综合评价: ☐优秀、☐良好、☐中等、☐及格、☐不及格

评阅教师签字:

日期:



目录

一 引言	2
1.1 研究背景	2
1.2 研究意义	2
二 任务实现过程	2
2.1 进入管理系统	2
2.1.1 登录	3
2.1.2 主页面	3
2.2 系统内部功能	4
2.2.1 添加业主信息	4
2.2.2 修改业主信息	7
2.2.3 删除业主信息	11
2.2.4 查询业主信息	14
2.2.5 保存信息	16
2.2.6 读取历史信息	17
2.2.7 帮助菜单	18
2.2.8 退出管理系统	18
三 系统运行结果及使用说明	18
3.1 登录页面显示	19
3.2 主页面显示	19
3.3 添加业主信息页面显示	19
3.4 修改业主信息页面显示	21
3.5 删除业主信息页面显示	23
3.6 查询业主信息页面显示	25
3.7 保存信息页面显示	27
3.8 读取历史信息页面显示	28
3.9 帮助菜单页面显示	28
3.10 退出管理系统页面显示	29
四 结论	29
4.1 回顾总结	29
4.2 反思与展望	29
五 参考文献	29
六 附录	30



一 引言

1.1 研究背景

现如今,科技的更新迭代下,我们已迈入了智能化信息时代,在生活水平不断提升的同时,相对应的,我们对于大数据信息处理的需求也在不断增加。伴随着社会和科技的发展,如何为用户提供最便捷、最优质的服务,物业信息管理系统是现代化城市建设中不可或缺的部分,发挥着举足轻重的作用,前景一片辽阔^[2]。

1.2 研究意义

本系统通过对小区业主信息进行综合利用,在方便小区日常管理的同时,也便于提高物业人员的工作效率,规范物业管理系统,从整体上提高小区的物业服务水平和人们的居住满意度^[2]。

二 任务实现过程

2.1 进入管理系统

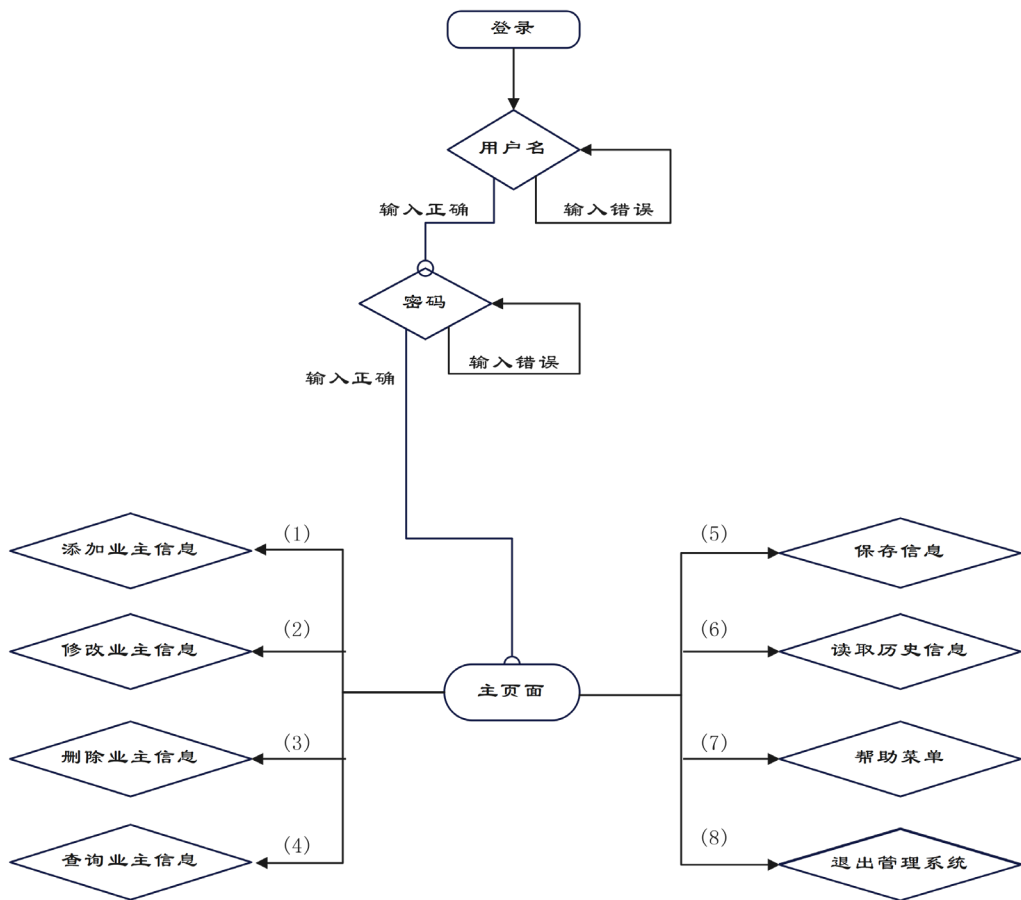


图 1: 登录和主页面流程图

考虑到信息的隐私性，确保不是任何一个人都有权限访问本系统，因此设置了登录页面，只有当使用者输入正确的用户名和密码时，才能进入系统。

关键函数: `enter()`

2.1.2 主页面

表 2: 主页面的程序代码

```
void Show::mainShow()
{
    cout << endl;
    cout << "\t\t\t*****主页面*****" << endl;
}
```



```
cout << endl;
cout << "\t\t\t*          【 (1) 添加业主信息 】          *"<< endl;
cout << "\t\t\t*          【 (2) 修改业主信息 】          *"<< endl;
cout << "\t\t\t*          【 (3) 删除业主信息 】          *"<< endl;
cout << "\t\t\t*          【 (4) 查询业主信息 】          *"<< endl;
cout << "\t\t\t*          【 (5)  保存信息   】          *"<< endl;
cout << "\t\t\t*          【 (6) 读取历史信息 】          *"<< endl;
cout << "\t\t\t*          【 (7)  帮助菜单   】          *"<< endl;
cout << "\t\t\t*          【 (8) 退出管理系统 】          *\n"<< endl;
cout << "\t\t\t*****"<< endl;
cout << endl;
cout << "\t 在此输入您的指令:  " ;
}
void Show::key( )
{
    int button;
    cin >> button;
    switch(button)
    {
        case 1:
            putShow( );
            break;
        case 2:
            reShow( );
            break;
        case 3:
            delShow( );
            break;
        case 4:
            infoShow( );
            break;
        case 5:
            save( );
            break;
        case 6:
            read( );
            break;
        case 7:
            help( );
            break;
        case 8:
            cout << endl;
            cout << "\t-----感谢您使用本系统-----" << endl;
            exit(0);
            break;
    }
}
```

2.2 系统内部功能

2.2.1 添加业主信息

初次进入系统，进行添加操作。添加页面的流程图如下所示：

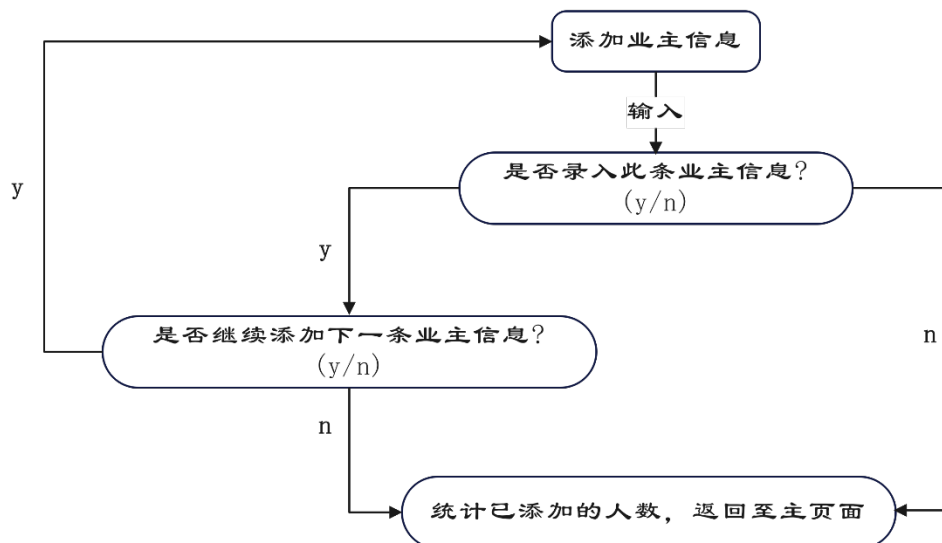


图 2: 添加业主信息的流程图

表 3: 添加业主信息的程序代码

关键函数: putShow()

```

void Show::putShow()
{
    system("cls");
    cout << endl;
    cout << "\t\t\t\t<----- (1) 添加业主信息 ----->\n" << endl;
    char name[20];
    int age;
    char sex[4];
    int year;
    int amount;
    char number[20];
    int door;
    char car[4];
    char marriage[4];
    int size = 0; //用于记录已添加的业主个数
    while(true)
    {
        cout << "【 姓名 】: ";
        cin >> name;
        cout << "【 年龄 】: ";
        cin >> age;
        cout << "【 性别 】: ";
        cin >> sex;
        cout << "【 已居住年限 】: ";
        cin >> year;
        cout << "【 家庭人数 】: ";
        cin >> amount;
        cout << "【 电话号码 】: ";
        cin >> number;
        cout << "【 所在楼栋单元 】: ";
        cin >> door;
    }
}
  
```



```
cout << "【 是否有车 】: " ;
cin >> car;
cout << "【 是否婚配 】: " ;
cin >> marriage;
Owner owner;
owner.setName(name);
owner.setAge(age);
owner.setSex(sex);
owner.setYear(year);
owner.setAmount(amount);
owner.setNumber(number);
owner.setDoor(door);
owner.setCar(car);
owner.setMarriage(marriage);
char ind;
cout << endl;
cout << "请您再次确认, 是否将此条业主信息录入系统? (y/n): ";
cin >> ind;
if( ind == 'y' || ind == 'Y') //实际应用中, 偶尔会把小写 y 打成大写 Y, 所以此
处设置大小写均可
{
    owns.insert(size, owner);
    char ins;
    cout << "【温馨提示】: 是否继续添加下一条业主信息? (y/n): ";
    cin >> ins;
    cout << endl;
    if(ins == 'y' || ins == 'Y')
    {
        cout << "目前已添加 " << size+1 << " 位业主的信息! " << endl; //因
为 s 是从 0 开始加的, 所以统计所添加的业主人数时需要再加 1
        cout << "请继续输入, 添加业主信息! " << endl;
        size++;
        cout << endl;
    }
    else
    {
        cout << "您选择的是'n', 按照您的需求, 不再添加下一条业主信息" << endl;
        cout << endl;
        cout << "已添加 " << size+1 << " 位业主的信息! " << endl;
        cout << "即将返回至主页面! " << endl;
        cout << endl;
        system("pause");
        break;
    }
}
else
{
    cout << "您选择的是'n', 按照您的需求, 不录入此条业主信息" << endl;
    if(size >= 0)
    {
        cout << "已添加 " << size << "位业主的信息! " << endl;
    }
    cout << endl;
    cout << "即将返回至主页面! " << endl;
    cout << endl;
    system("pause");
}
```



```

        break;
    }
}
}

```

2.2.2 修改业主信息

修改业主信息的流程控制图如下：

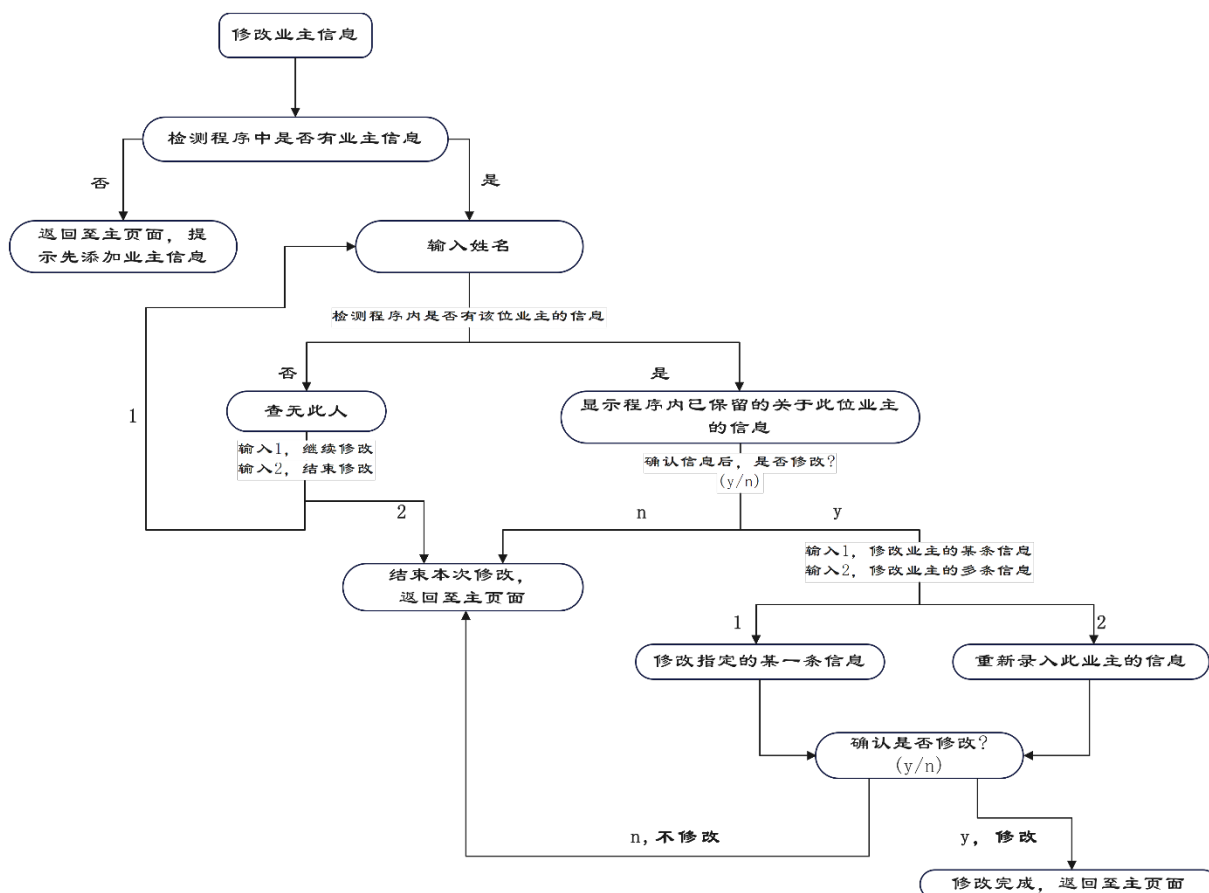


图 3：修改业主信息的流程图

表 4：修改业主信息的程序代码

关键函数：reShow()

```

void Show::reShow( )
{
    system("cls");
    cout << endl;
    cout << "\t\t\t<----- (2) 修改业主信息 ----->\n" << endl;
    int size = owns.length( );
    if(size == 0)
    {
        cout << endl;
        cout << "【温馨提示】：系统还未存储任何业主信息！请返回至主页面添加业主信息！" <<
endl;
    }
}

```



```
cout << "即将返回至主页面!" << endl;
cout << endl;
system("pause");
mainShow( );
}
else
{
    cout << "【温馨提示】：此页面为业主信息修改页面，修改过程中请您仔细操作，以免造成信息录入失误，造成不必要的损失\n" << endl;
    cout << "您修改业主信息时，为了便于定位所要修改的业主信息，请先输入业主的姓名：";
    char ni[20];
    cin >> ni;
    int upd = -1;
    for(int i = 0; i < size ; i++)
    {
        Owner owner;
        owns.getData(i,owner);
        char mi[20];
        owner.getName(mi);
        if(strcmp(ni, mi) == 0)
        {
            cout << endl;
            cout << "-----以下信息为业主 " << ni << " 已录入的信息";
            cout << endl;
            owner.output( );
            upd = i;
            cout << "-----\n";
            cout << endl;
            cout << "【温馨提示】：请您确认好，如上信息是否有需要修改的地方(y/n)：";
            char ind;
            cin >> ind;
            if(ind == 'y' || ind == 'Y')
            {
                Owner owner;
                char name[20];
                int age;
                char sex[4];
                int year;
                int amount;
                char number[20];
                int door;
                char car[4];
                char marriage[4];
                cout << "您总共需要修改业主的几条信息？（若只修改1条，请输入1；若修改多条，请输入2）： ";
                int a;
                cin >> a;
                cout << endl;
                if(a == 1)
                {
                    cout << "您需要修改业主的哪条信息？" << endl;
                    cout << "      _____" << endl;
                    cout << "      | \t a: 修改姓名          b: 修改年龄          c: 修改性别          | " << endl;
                    cout << "      | \t d: 修改已居住年限    e: 修改家庭人数    f: 修改其他信息          | " << endl;
                }
            }
        }
    }
}
```



数学与统计学院课程设计（实习）报告

```
改电话号码      | "<< endl;
                  cout << "  | \t j: 修改所在楼栋单元    h: 修改是否有车    i: 修
改是否婚配      | "<< endl;
                  cout << " _____" << endl;
                  cout << "请输入字段前对应的字母: ";
                  char ch;
                  cin >> ch;
                  switch(ch)
                  {
                      case 'a':
                          cout << "业主" << ni << "的姓名改为: ";
                          cin >> name;
                          owner.setName(name);
                          break;
                      case 'b':
                          cout << "业主" << ni << "的年龄改为: ";
                          cin >> age;
                          owner.setAge(age);
                          break;
                      case 'c':
                          cout << "业主" << ni << "的性别改为: ";
                          cin >> sex;
                          owner.setSex(sex);
                          break;
                      case 'd':
                          cout << "业主" << ni << "的已居住年限改为: ";
                          cin >> year;
                          owner.setYear(year);
                          break;
                      case 'e':
                          cout << "业主" << ni << "的家庭人数改为: ";
                          cin >> amount;
                          owner.setAmount(amount);
                          break;
                      case 'f':
                          cout << "业主" << ni << "的电话号码改为: ";
                          cin >> number;
                          owner.setNumber(number);
                          break;
                      case 'j':
                          cout << "业主" << ni << "所在楼栋单元改为: ";
                          cin >> door;
                          owner.setDoor(door);
                          break;
                      case 'h':
                          cout << "业主" << ni << "是否有车改为: ";
                          cin >> car;
                          owner.setCar(car);
                          break;
                      case 'i':
                          cout << "业主" << ni << "是否婚配改为: ";
                          cin >> marriage;
                          owner.setMarriage(marriage);
                          break;
                  }
            }
        }
    }
else if(a == 2)
```



```
{
    cout << "【温馨提示】：鉴于您需要修改的信息较多，请您重新输入业主的所有信息，系统将保存您修改后的业主信息\n" << endl;
    cout << "业主" << ni << "的姓名改为： ";
    cin >> name;
    owner.setName(name);
    cout << "业主" << ni << "的年龄改为： ";
    cin >> age;
    owner.setAge(age);
    cout << "业主" << ni << "的性别改为： ";
    cin >> sex;
    owner.setSex(sex);
    cout << "业主" << ni << "的已居住年限改为： ";
    cin >> year;
    owner.setYear(year);
    cout << "业主" << ni << "的家庭人数改为： ";
    cin >> amount;
    owner.setAmount(amount);
    cout << "业主" << ni << "的电话号码改为： ";
    cin >> number;
    owner.setNumber(number);
    cout << "业主" << ni << "所在楼栋单元改为： ";
    cin >> door;
    owner.setDoor(door);
    cout << "业主" << ni << "是否有车改为： ";
    cin >> car;
    owner.setCar(car);
    cout << "业主" << ni << "是否婚配改为： ";
    cin >> marriage;
    owner.setMarriage(marriage);
}
cout << endl;
cout << "确认修改？(y/n) " ;
char ins;
cin >> ins;
if(ins == 'y' || ins == 'Y')
{
    owns.setData(upd,owner);
    cout << "修改完毕！" << endl;
    cout << "即将返回至主页面！" << endl;
    cout << endl;
    system("pause");
    break;
    mainShow( );
}
else
{
    cout << endl;
    cout << "您选择的是'n'，按照您的需求，本次不执行修改操作" << endl;
    cout << endl;
    cout << "即将返回至主页面！" << endl;
    cout << endl;
    system("pause");
}
}
```



```
else
{
    cout << endl;
    cout << "您选择的是'n'，按照您的需求，本次不执行修改操作" << endl;
    cout << endl;
    cout << "即将返回至主页面！" << endl;
    cout << endl;
    system("pause");
}
}
if(i == size-1 && upd == -1) //另加的一个条件，若找不到对应的信息，则给出
相应的提示，即“查无此人”
{
    cout << endl;
    cout << "查无此人！\n" << endl;
    cout << "【温馨提示】：若继续修改，请输入 1；若结束本次修改，请输入 2：
";

    int inx;
    cin >> inx;
    if(inx == 1)
    {
        reShow( );
        break;
        mainShow( );
    }
    else if(inx == 2)
    {
        cout << endl;
        cout << "您选择了结束本次修改，即将返回至主页面！" << endl;
        cout << endl;
        system("pause");
        break;
        mainShow( );
    }
    else
    {
        cout << "您输入的不是 1 或 2，输入错误！请重新输入！" << endl;
        system("pause");
        reShow( );
    }
}
}
}
}
```

2.2.3 删除业主信息

删除业主信息的流程控制图如下：

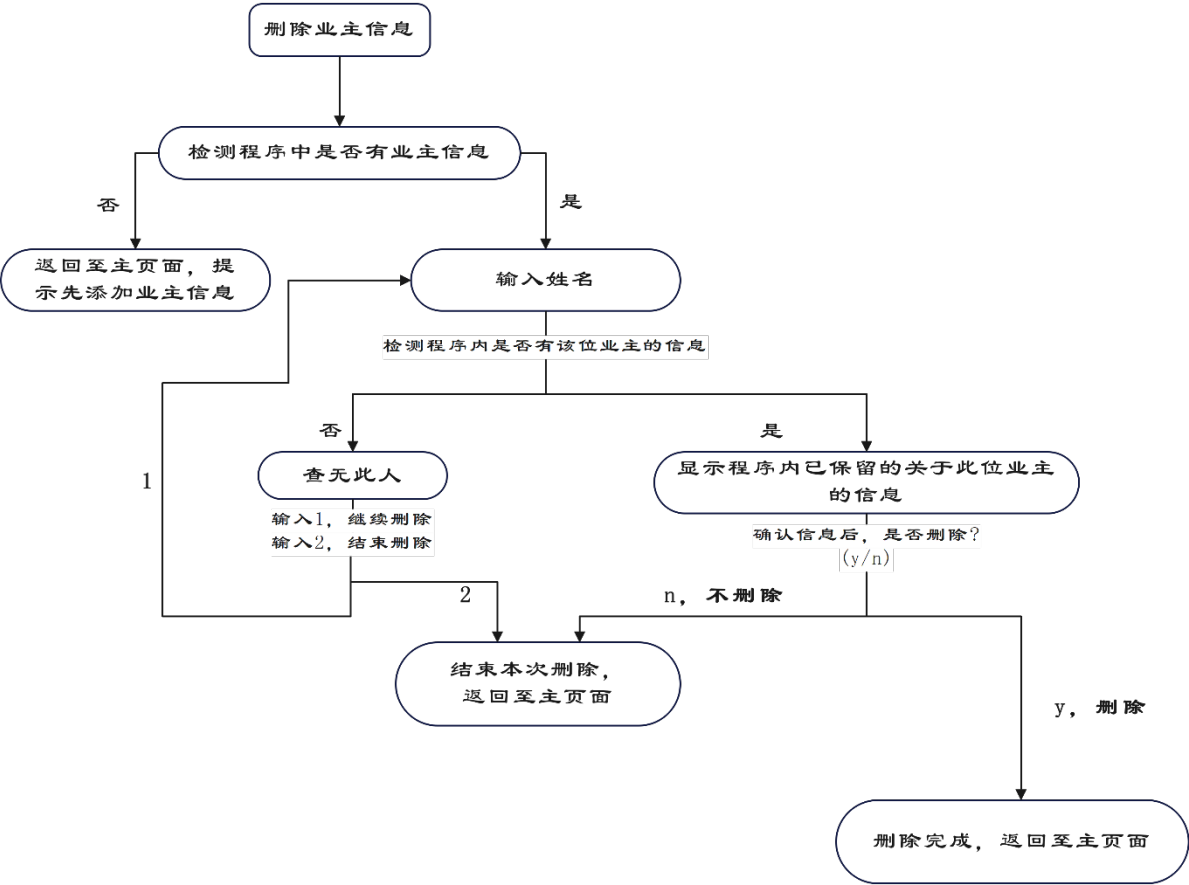


图 4：删除业主信息的流程图

表 5：删除业主信息的程序代码

关键函数：del()
<pre>void Show::delShow() { system("cls"); cout << endl; cout << "\t\t\t<------(3) 删除业主信息----->\n" << endl; cout << endl; int size = owns.length(); if(size == 0) { cout << endl; cout << "【温馨提示】：系统还未存储任何业主信息！请返回至主页面添加业主信息！" << endl; cout << "即将返回至主页面！" << endl; cout << endl; system("pause"); mainShow(); } else { cout << "【温馨提示】：此页面为业主信息删除页面，删除前，请确保需要删除的业主名</pre>



数学与统计学院课程设计（实习）报告

```
单正确，避免误删 " << endl;
cout << endl;
cout << "您删除业主信息时，为了便于定位所要删除的业主信息，请先输入业主的姓名：
";
char ni[20];
cin >> ni;
int del = -1;
for(int i = 0; i < size; i++)
{
    Owner owner;
    owns.getData(i,owner);
    char mi[20];
    owner.getName(mi);
    if(strcmp(ni, mi) == 0)
    {
        cout << endl;
        cout << "-----以下信息为业主" << ni << "已录入的信息-----"
<< endl;

        owner.output( );
        del = i;
        cout << "-----\n" << endl;
        cout << "【温馨提示】：请您确认好，上面的业主信息是否要删除（y/n）" ;
        char ins;
        cin >> ins;
        if(ins == 'y' || ins == 'Y')
        {
            Owner owner;
            owns.remove(del,owner);
            cout << endl;
            cout << "删除完成！" << endl;
            cout << "即将返回至主页面！" << endl;
            cout << endl;
            system("pause");
        }
        else
        {
            cout << endl;
            cout << "您选择的是‘n’，按照您的需求，本次不执行删除操作" << endl;
            cout << endl;
            cout << "即将返回至主页面！" << endl;
            cout << endl;
            system("pause");
        }
    }
    if(i == size-1 && del == -1) //另加的一个条件，若找不到对应的业主信息，则
提示“查无此人”
    {
        cout << endl;
        cout << "查无此人！\n" << endl;
        cout << "【温馨提示】：若继续删除，请输入 1；若结束本次删除，请输入 2：
";

        int inx;
        cin >> inx;
        if(inx == 1)
        {
            delShow( );
            break;

```




```
---->\n" << endl;
int size = owns.length( );
if(size == 0)
{
    cout << endl;
    cout << "【温馨提示】：系统还未存储任何业主信息！请返回至主页面添加业主信息！" <<
endl;
    cout << "即将返回至主页面！" << endl;
    cout << endl;
    system("pause");
    mainShow( );
}
else
{
    cout << "如果需要查询系统内所有业主的信息，请输入 1；如果需要查询单个业主的信息，
请输入 2： " ;
    int ind;
    cin >> ind;
    if(ind == 1)
    {
        cout << endl;
        cout << "***** √查询所有业主信息 *****\n" << endl;
        for(int i = 0; i < size; i++)
        {
            Owner owner;
            owns.getData(i, owner);
            owner.output( );
            cout << endl;
        }
        cout << endl;
        cout << "以上为所有的业主信息！\n" << endl;
        cout << endl;
        cout << "查询完毕！" << endl;
        cout << "即将返回至主页面！" << endl;
        cout << endl;
        system("pause");
    }
    else if(ind == 2)
    {
        cout << endl;
        cout << "***** √查询单个业主信息 *****\n" << endl;
        cout << "您查询业主信息时，为了便于定位所要查询的业主信息，请先输入业主的姓
名： ";
        char ni[20];
        cin >> ni;
        int j;
        for(j = 0; j < size; j++)
        {
            Owner owner;
            owns.getData(j, owner);
            char mi[20];
            owner.getName(mi);
            if(strcmp(ni, mi) == 0)
            {
                cout << endl;
                owner.output( );
                cout << endl;
            }
        }
    }
}
```



```
        cout << "业主" << ni << "的信息如上\n" << endl;
        cout << "查询完毕!" << endl;
        cout << "即将返回至主页面!" << endl;
        cout << endl;
        system("pause");
        break;
        mainShow( );
    }
    if(j == size-1)
    {
        if(strcmp(ni, mi) == 0)
        {
            return;
        }
        else
        {
            cout << endl;
            cout << "查无此人! \n" << endl;
            cout << "【温馨提示】: 若继续查询, 请输入 1; 若结束本次查询, 请
输入 2:  " ;

            int inx;
            cin >> inx;
            if(inx == 1)
            {
                infoShow( );
                break;
                mainShow( );
            }
            else
            {
                cout << endl;
                cout << "您选择了结束本次查询, 即将返回至主页面!" << endl;
                cout << endl;
                system("pause");
                break;
                mainShow( );
            }
        }
    }
}
else
{
    cout << "您输入的不是 1 或 2, 输入错误! 请重新输入!" << endl;
    system("pause");
    infoShow( );
}
}
```

2.2.5 保存信息

将此次录入的信息保存在“物业基本信息管理系统 - 业主信息.txt”文件中, 可永久保存。



表 7：保存信息的程序代码

关键函数：save()

```
void Show::save( ) //保存文件记录
{
    system("cls");
    cout << endl;
    cout << "\t\t\t\t------(5) 保存信息 ----->\n" << endl;
    ofstream out;
    out.open("物业基本信息管理系统 - 业主信息.txt", ios::app);
    if (!out.is_open( )) {
        cout << "文件无法打开！" << endl;
        system("pause");
        return;
    }
    int size = owns.length( );
    char name[20];
    int age;
    char sex[4];
    int year;
    int amount;
    char number[20];
    int door;
    char car[4];
    char marriage[4];
    for(int i = 0; i < size; i++)
    {
        Owner owner;
        owns.getData(i , owner);
        owner.getName(name);
        out << "【 姓名 】: " << name << endl;
        out << "【 年龄 】: " << owner.getAge( ) << endl;
        owner.getSex(sex);
        out << "【 性别 】: " << sex << endl;
        out << "【 已居住年限 】: " << owner.getYear( ) << endl;
        out << "【 家庭人数 】: " << owner.getAmount( ) << endl;
        owner.getNumber(number);
        out << "【 电话号码 】: " << number << endl;
        out << "【 所在楼栋单元 】: " << owner.getDoor( ) << endl;
        owner.getCar(car);
        out << "【 是否有车 】: " << car << endl;
        owner.getMarriage(marriage);
        out << "【 是否婚配 】: " << marriage << endl;
        out << endl;
    }
    out.close();
    cout << endl;
    cout << "-----已成功保存所有业主信息! ----->\n" << endl;
    system("pause");
}
```

2.2.6 读取历史信息

读取“物业基本信息管理系统 - 业主信息.txt”文件，浏览历史记录。



表 8: 读取信息的程序代码

关键函数: read()
<pre>void Show::read() //读取文件记录 { system("cls"); cout << endl; cout << "\t\t\t\t<----- (6) 读取历史信息 ----->\n" << endl; ifstream in; in.open("物业基本信息管理系统 - 业主信息.txt", ios::app); if (!in.is_open()) { cout << "文件无法打开! " << endl; return; } char se[300]; while(!in.eof()) { in.getline(se,100); cout << se << endl; } in.close(); cout << "-----已成功读取所有业主信息! -----\n" << endl; system("pause"); }</pre>

2.2.7 帮助菜单

对程序内的信息录入进行简要说明，提醒使用者。

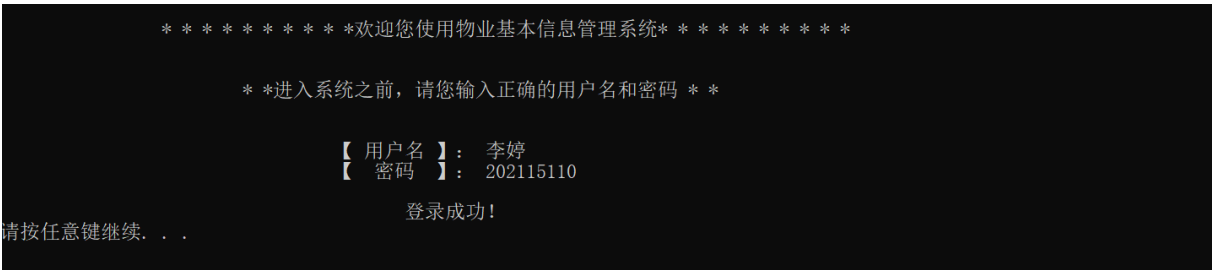
2.2.8 退出管理系统

完成本次所有操作，退出程序。

三 系统运行结果及使用说明

3.1 登录页面显示

正确的用户名和密码：





数学与统计学院课程设计（实习）报告

用户名输入错误时（示例）：

```
*****欢迎您使用物业基本信息管理系统*****

**进入系统之前，请您输入正确的用户名和密码**

【 用户名 】： 李四
此用户名不能访问本系统！请输入正确的用户名！

【 用户名 】：
```

密码输入错误时（示例）：

```
*****欢迎您使用物业基本信息管理系统*****

**进入系统之前，请您输入正确的用户名和密码**

【 用户名 】： 李四
此用户名不能访问本系统！请输入正确的用户名！

【 用户名 】： 李婷
【 密码 】： 123456
此密码不匹配输入的用户名！请输入正确的密码！

【 密码 】：
```

3.2 主页面显示

```
*****主页面*****

*      【 (1) 添加业主信息 】      *
*      【 (2) 修改业主信息 】      *
*      【 (3) 删除业主信息 】      *
*      【 (4) 查询业主信息 】      *
*      【 (5) 保存信息 】          *
*      【 (6) 读取历史信息 】      *
*      【 (7) 帮助菜单 】          *
*      【 (8) 退出管理系统 】      *

*****

在此输入您的指令：
```

3.3 添加业主信息页面显示

成功添加一条业主信息：



数学与统计学院课程设计(实习)报告

```
<----- (1) 添加业主信息 ----->

【 姓名 】： 李唐
【 年龄 】： 45
【 性别 】： 男
【 已居住年限 】： 32
【 家庭人数 】： 4
【 电话号码 】： 16738297633
【 所在楼栋单元 】： 130201
【 是否有车 】： 是
【 是否婚配 】： 是

请您再次确认，是否将此条业主信息录入系统？ (y/n)： y
【温馨提示】： 是否继续添加下一条业主信息？ (y/n)： n

您选择的是 'n'，按照您的需求，不再添加下一条业主信息

已添加 1 位业主的信息！
即将返回至主页面！
```

添加多条业主信息时：

```
<----- (1) 添加业主信息 ----->

【 姓名 】： 张伶
【 年龄 】： 28
【 性别 】： 女
【 已居住年限 】： 12
【 家庭人数 】： 1
【 电话号码 】： 17382298013
【 所在楼栋单元 】： 030201
【 是否有车 】： 否
【 是否婚配 】： 否

请您再次确认，是否将此条业主信息录入系统？ (y/n)： y
【温馨提示】： 是否继续添加下一条业主信息？ (y/n)： y

目前已添加 1 位业主的信息！
请继续输入，添加业主信息！

【 姓名 】： 王三强
【 年龄 】： 46
【 性别 】： 男
【 已居住年限 】： 34
【 家庭人数 】： 3
【 电话号码 】： 18432920441
【 所在楼栋单元 】： 051301
【 是否有车 】： 是
【 是否婚配 】： 是

请您再次确认，是否将此条业主信息录入系统？ (y/n)： y
【温馨提示】： 是否继续添加下一条业主信息？ (y/n)： y

目前已添加 2 位业主的信息！
请继续输入，添加业主信息！

【 姓名 】： 李莉
【 年龄 】： 35
【 性别 】： 女
【 已居住年限 】： 22
【 家庭人数 】： 2
【 电话号码 】： 13275849301
【 所在楼栋单元 】： 081101
【 是否有车 】： 是
【 是否婚配 】： 是

请您再次确认，是否将此条业主信息录入系统？ (y/n)： y
【温馨提示】： 是否继续添加下一条业主信息？ (y/n)： n

您选择的是 'n'，按照您的需求，不再添加下一条业主信息

已添加 3 位业主的信息！
即将返回至主页面！

请按任意键继续. . .
```



数学与统计学院课程设计（实习）报告

选择不添加业主信息时：

```
<----- (1) 添加业主信息 ----->

【 姓名      】： 林三
【 年龄      】： 23
【 性别      】： 男
【 已居住年限 】： 1
【 家庭人数  】： 1
【 电话号码  】： 18498749301
【 所在楼栋单元 】： 020102
【 是否有车  】： 否
【 是否婚配  】： 否

请您再次确认，是否将此条业主信息录入系统？（y/n）： n
您选择的是‘n’，按照您的需求，不录入此条业主信息
已添加0位业主的信息！
```

3.4 修改业主信息页面显示

程序未存储业主信息时：

```
<----- (2) 修改业主信息 ----->

【温馨提示】：系统还未存储任何业主信息！请返回至主页面添加业主信息！
即将返回至主页面！

请按任意键继续. . .
```

姓名输入错误时：

```
<----- (2) 修改业主信息 ----->

【温馨提示】：此页面为业主信息修改页面，修改过程中请您仔细操作，以免造成信息录入失误，造成不必要的损失
您修改业主信息时，为了便于定位所要修改的业主信息，请先输入业主的姓名： 张三
查无此人！

【温馨提示】：若继续修改，请输入1；若结束本次修改，请输入2：
```

输入 1 时，返回至上一步，重新输入：

```
<----- (2) 修改业主信息 ----->

【温馨提示】：此页面为业主信息修改页面，修改过程中请您仔细操作，以免造成信息录入失误，造成不必要的损失
您修改业主信息时，为了便于定位所要修改的业主信息，请先输入业主的姓名：
```

输入 2 时，返回至主页面：

```
【温馨提示】：若继续修改，请输入1；若结束本次修改，请输入2： 2
您选择了结束本次修改，即将返回至主页面！
```

输入的不是 1 或 2 时，返回至上一步，重新输入：



```
【温馨提示】：若继续修改，请输入1；若结束本次修改，请输入2： 10
您输入的不是1或2，输入错误！请重新输入！
请按任意键继续. . .
```

姓名输入正确时（以程序中已经存在的业主信息为例）：

```
<----- (2) 修改业主信息 ----->

【温馨提示】：此页面为业主信息修改页面，修改过程中请您仔细操作，以免造成信息录入失误，造成不必要的损失
您修改业主信息时，为了便于定位所要修改的业主信息，请先输入业主的姓名： 李唐

. . . . . 以下信息为业主 李唐 已录入的信息 . . . . .
【业主的姓名】： 李唐
【李唐的年龄】： 45
【李唐的性别】： 男
【李唐的已居住年限】： 32
【李唐的家庭人数】： 4
【李唐的电话号码】： 16738297633
【李唐所在的楼栋单元】： 130201
【李唐家是否有车】： 是
【李唐是否婚配】： 是
. . . . .

【温馨提示】：请您确认好，如上信息是否有需要修改的地方（y/n）： _
```

若不修改：

```
<----- (2) 修改业主信息 ----->

【温馨提示】：此页面为业主信息修改页面，修改过程中请您仔细操作，以免造成信息录入失误，造成不必要的损失
您修改业主信息时，为了便于定位所要修改的业主信息，请先输入业主的姓名： 李唐

. . . . . 以下信息为业主 李唐 已录入的信息 . . . . .
【业主的姓名】： 李唐
【李唐的年龄】： 45
【李唐的性别】： 男
【李唐的已居住年限】： 32
【李唐的家庭人数】： 4
【李唐的电话号码】： 16738297633
【李唐所在的楼栋单元】： 130201
【李唐家是否有车】： 是
【李唐是否婚配】： 是
. . . . .

【温馨提示】：请您确认好，如上信息是否有需要修改的地方（y/n）： n

您选择的是‘n’，按照您的需求，本次不执行修改操作
即将返回至主页面！
请按任意键继续. . .
```

若继续修改（只修改单个的一条时）（示例）：



数学与统计学院课程设计（实习）报告

```
【温馨提示】：请您确认好，如上信息是否有需要修改的地方（y/n）： y
您总共需要修改业主的几条信息？（若只修改1条，请输入1；若修改多条，请输入2）： 1
您需要修改业主的哪条信息？

a: 修改姓名      b: 修改年龄      c: 修改性别
d: 修改已居住年限  e: 修改家庭人数  f: 修改电话号码
j: 修改所在楼栋单元 h: 修改是否有车  i: 修改是否婚配

请输入字段前对应的字母： h
业主李唐是否有车改为： 否

确认修改？（y/n） y
修改完毕！
即将返回至主页面！

请按任意键继续. . .
```

若继续修改（修改多条时）（示例）：

```
【温馨提示】：请您确认好，如上信息是否有需要修改的地方（y/n）： y
您总共需要修改业主的几条信息？（若只修改1条，请输入1；若修改多条，请输入2）： 2

【温馨提示】：鉴于您需要修改的信息较多，请您重新输入业主的所有信息，系统将保存您修改后的业主信息

业主李唐的姓名改为： 李唐
业主李唐的年龄改为： 46
业主李唐的性别改为： 男
业主李唐的已居住年限改为： 33
业主李唐的家庭人数改为： 4
业主李唐的电话号码改为： 17385957281
业主李唐所在楼栋单元改为： 130201
业主李唐是否有车改为： 是
业主李唐是否婚配改为： 是

确认修改？（y/n） y
修改完毕！
即将返回至主页面！

请按任意键继续. . .
```

修改的最后一步时，若决定不再修改，则返回至主页面：

```
确认修改？（y/n） n

您选择的是‘n’，按照您的需求，本次不执行修改操作

即将返回至主页面！
```

3.5 删除业主信息页面显示

程序未存储业主信息时：

```
<----- (3) 删除业主信息 ----->

【温馨提示】：系统还未存储任何业主信息！请返回至主页面添加业主信息！
即将返回至主页面！

请按任意键继续. . .
```



数学与统计学院课程设计(实习)报告

姓名输入错误时，提示：

```
<----- (3) 删除业主信息 ----->

【温馨提示】：此页面为业主信息删除页面，删除前，请确保需要删除的业主名单正确，避免误删
您删除业主信息时，为了便于定位所要删除的业主信息，请先输入业主的姓名： 王五
查无此人！
【温馨提示】：若继续删除，请输入1；若结束本次删除，请输入2：
```

输入 1 时，返回至上一步，重新输入：

```
<----- (3) 删除业主信息 ----->

【温馨提示】：此页面为业主信息删除页面，删除前，请确保需要删除的业主名单正确，避免误删
您删除业主信息时，为了便于定位所要删除的业主信息，请先输入业主的姓名：
```

输入 2 时，返回至主页面：

```
【温馨提示】：若继续删除，请输入1；若结束本次删除，请输入2： 2
您选择了结束本次删除，即将返回至主页面！
请按任意键继续. . .
```

输入的不是 1 或 2 时，返回至上一步，重新输入：

```
【温馨提示】：若继续删除，请输入1；若结束本次删除，请输入2： 5
您输入的不是1或2，输入错误！请重新输入！
请按任意键继续. . .
```

姓名输入正确时（以程序中已经存在的业主信息为例）：

```
<----- (3) 删除业主信息 ----->

【温馨提示】：此页面为业主信息删除页面，删除前，请确保需要删除的业主名单正确，避免误删
您删除业主信息时，为了便于定位所要删除的业主信息，请先输入业主的姓名： 李唐
- . - . - . - . - . 以下信息为业主李唐已录入的信息 - . - . - . - . - .
【业主的姓名】： 李唐
【李唐的年龄】： 46
【李唐的性别】： 男
【李唐的已居住年限】： 33
【李唐的家庭人数】： 4
【李唐的电话号码】： 17385957281
【李唐所在的楼栋单元】： 130201
【李唐家是否有车】： 是
【李唐是否婚配】： 是
- . - . - . - . - .

【温馨提示】：请您确认好，上面的业主信息是否要删除 (y/n)
```

若不删除，返回至主页面：



数学与统计学院课程设计（实习）报告

```
【温馨提示】：请您确认好，上面的业主信息是否要删除 (y/n) n
您选择的是 'n'，按照您的需求，本次不执行删除操作
即将返回至主页面！
请按任意键继续. . .
```

若删除：

```
【温馨提示】：请您确认好，上面的业主信息是否要删除 (y/n) y
删除完成！
即将返回至主页面！
请按任意键继续. . .
```

3.6 查询业主信息页面显示

程序未存储业主信息时：

```
<----- (4) 查询业主信息 ----->

【温馨提示】：系统还未存储任何业主信息！请返回至主页面添加业主信息！
即将返回至主页面！
请按任意键继续. . .
```

查询程序内所有业主信息：

```
<----- (4) 查询业主信息 ----->

如果需要查询系统内所有业主的信息，请输入1；如果需要查询单个业主的信息，请输入2： 1
*****  √ 查询所有业主信息  *****

【业主的姓名】： 张伶
【张伶的年龄】： 28
【张伶的性别】： 女
【张伶的已居住年限】： 12
【张伶的家庭人数】： 1
【张伶的电话号码】： 17382298013
【张伶所在的楼栋单元】： 30201
【张伶家是否有车】： 否
【张伶是否婚配】： 否

【业主的姓名】： 王三强
【王三强的年龄】： 46
【王三强的性别】： 男
【王三强的已居住年限】： 34
【王三强的家庭人数】： 3
【王三强的电话号码】： 18432920441
【王三强所在的楼栋单元】： 51301
【王三强家是否有车】： 是
【王三强是否婚配】： 是

【业主的姓名】： 李莉
【李莉的年龄】： 35
【李莉的性别】： 女
【李莉的已居住年限】： 22
【李莉的家庭人数】： 2
【李莉的电话号码】： 13275849301
【李莉所在的楼栋单元】： 81101
【李莉家是否有车】： 是
【李莉是否婚配】： 是

以上为所有的业主信息！

查询完毕！
即将返回至主页面！
请按任意键继续. . .
```



查询单个业主信息:

```
<----- (4) 查询业主信息 ----->
如果需要查询系统内所有业主的信息, 请输入1; 如果需要查询单个业主的信息, 请输入2: 2
***** √ 查询单个业主信息 *****
您查询业主信息时, 为了便于定位所要查询的业主信息, 请先输入业主的姓名:
```

姓名不正确时:

```
<----- (4) 查询业主信息 ----->
如果需要查询系统内所有业主的信息, 请输入1; 如果需要查询单个业主的信息, 请输入2: 2
***** √ 查询单个业主信息 *****
您查询业主信息时, 为了便于定位所要查询的业主信息, 请先输入业主的姓名: 张三
查无此人!
【温馨提示】: 若继续查询, 请输入1; 若结束本次查询, 请输入2:
```

输入 1, 继续查询:

```
<----- (4) 查询业主信息 ----->
如果需要查询系统内所有业主的信息, 请输入1; 如果需要查询单个业主的信息, 请输入2:
```

输入 2, 结束本次查询, 返回至主页面:

```
【温馨提示】: 若继续查询, 请输入1; 若结束本次查询, 请输入2: 2
您选择了结束本次查询, 即将返回至主页面!
请按任意键继续. . .
```

姓名正确时 (以查询程序内的某一位业主信息为例):



数学与统计学院课程设计（实习）报告

```
<----- (4) 查询业主信息 ----->

如果需要查询系统内所有业主的信息，请输入1；如果需要查询单个业主的信息，请输入2： 2

***** √ 查询单个业主信息 *****

您查询业主信息时，为了便于定位所要查询的业主信息，请先输入业主的姓名： 李莉

【业主的姓名】： 李莉
【李莉的年龄】： 35
【李莉的性别】： 女
【李莉的已居住年限】： 22
【李莉的家庭人数】： 2
【李莉的电话号码】： 13275849301
【李莉所在的楼栋单元】： 81101
【李莉家是否有车】： 是
【李莉是否婚配】： 是

业主李莉的信息如上

查询完毕！
即将返回至主页面！

请按任意键继续. . .
```

3.7 保存信息页面显示

程序内显示为：

```
<----- (5) 保存信息 ----->

. . . . . 已成功保存所有业主信息！ . . . . .

请按任意键继续. . .
```

保存为 txt 文件，并命名为“物业基本信息管理系统 - 业主信息”。以下为 txt 文件中的内容：

```
*物业基本信息管理系统 - 业主信息.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

【 姓名 】： 张伶
【 年龄 】： 28
【 性别 】： 女
【 已居住年限 】： 12
【 家庭人数 】： 1
【 电话号码 】： 17382298013
【 所在楼栋单元 】： 30201
【 是否有车 】： 否
【 是否婚配 】： 否

【 姓名 】： 王三强
【 年龄 】： 46
【 性别 】： 男
【 已居住年限 】： 34
【 家庭人数 】： 3
【 电话号码 】： 18432920441
【 所在楼栋单元 】： 51301
【 是否有车 】： 是
【 是否婚配 】： 是

【 姓名 】： 李莉
【 年龄 】： 35
【 性别 】： 女
【 已居住年限 】： 22
【 家庭人数 】： 2
【 电话号码 】： 13275849301
【 所在楼栋单元 】： 81101
【 是否有车 】： 是
【 是否婚配 】： 是

第 30 行, 第 1 列 100% Windows (CRL) 册 英力, 简
```



3.8 读取历史信息页面显示



3.9 帮助菜单页面显示

对程序的相关输入进行解释：





3.10 退出管理系统页面显示



四 结论

4.1 回顾总结

本程序有效地完成了对业主信息的添加、修改、删除和查询等功能，在此基本上，通过设置登录页面，可以在一定程度上确保信息的安全性；在实现增删改查时，每一步都设置了“安全操作”，防止使用者在对信息进行修正时，会产生偏差，减少了不必要的麻烦；最后，成功将所要保存的业主信息记录在 txt 文件中，实现业主信息的永久保存，同时也方便下次再使用本程序时，能查看以往的记录。

4.2 反思与展望

本程序的功能不是十分完美的，目前程序存储的信息量不大，运行时的速度较快，未存在卡顿现象。若要运用至实际生活中，除了要收集更丰富的业主信息之外，也需要加快程序读取信息等各方面的能力；同时，也有更多的功能值得完善。

五 参考文献

- [1] 崔亚楠, 小区物业信息管理系统 V1.0. 广西壮族自治区, 柳州工学院, 2019-09-04.
- [2] 徐连霞, 傅伟. 小区物业管理系统的设计与实现 [J]. 现代营销 (信息版), 2019(10):189.



六 附 录

头文件

```
#include<iostream>
#include<cstring>
#include<fstream>
using namespace std;
```

单链表

```
//-----定义一个链式表-----
template <class T>
struct LinkNode
{
    LinkNode<T> * next = NULL; //定义指针域, 并初始化为空指针
    T data; //定义数据域
};
template <class T>
class Link
{
private:
    LinkNode<T> * head; //定义链表头指针
public:
    Link( );
    ~Link( );
    int length( );
    int search(T & x);
    bool isEmpty( );
    bool isFull();
    void makeEmpty( );
    bool getData(int index, T & x);
    bool setData(int index, T & x);
    bool insert(int index, T & x);
    bool remove(int index, T & x);
    void output();
};
template <class T>
Link<T>::Link( )
{
    head = new LinkNode<T>; // 构造函数
}
template <class T>
Link<T>::~~Link()
{
    //析构函数
    makeEmpty( ); //置空链表
    delete head; //删除头结点
}
template <class T>
int Link<T>::length( )
{
    LinkNode<T> * n = head->next;
    int count = 0;
    while(n != NULL)
```




```
{
    n = n->next;
    count++;
}
return count;    //返回链表的长度
}
template <class T>
int Link<T>::search(T & x)    //搜索含数据 x 的元素
{
    LinkNode<T> * n = head->next;
    int i = 0;
    while(n != NULL){
        if(n->data == x)
            return i;    //搜索成功，返回 x 对应的元素的索引
        n = n->next;
        i++;
    }
    return -1;    //搜索失败，返回-1
}
template <class T>
bool Link<T>::isEmpty( )
{
    return head->next == NULL ? true : false;    //判断链表是否满了，如果满了，则返回 true；如果不满，则返回 false
}
template <class T>
bool Link<T>::isFull( )
{
    return false;
}
template <class T>
void Link<T>::makeEmpty( )
{    // 置空链表
    LinkNode<T> * n;
    while(head->next != NULL)
    {    //判断链表是否为空，如果不为空，则删除所有结点
        n = head->next;
        head->next = n->next;    //保存被删结点，从链表里面删除结点
        delete n;    // 删除全部结点
    }
}
template <class T>
bool Link<T>::getData(int index, T & x)
{    //取出下标为 index 的元素的值
    LinkNode<T> * n = head->next;
    for(int i = 0; i <= index; i++)
    {
        if(n == NULL)
            return false;
        if(i == index)
        {
            x = n->data;
            return true;
        }
        n = n->next;
    }
    return false;
}
```



```
}
template <class T>
bool Link<T>::setData(int index, T & x)
{ //用 x 代替下标为 index 的元素的值
  LinkNode<T> * n = head->next;
  for(int i = 0; i <= index; i++)
  {
    if(n == NULL)
      return false;
    if(i == index)
    {
      n->data = x;
      return true;
    }
    n = n->next;
  }
  return false;
}

template <class T>
bool Link<T>::insert(int index, T & x)
{ //在索引为 index 的元素后面插入新元素 x
  LinkNode<T> * n = head;
  for(int i = 0; i <= index; i++)
  {
    if(n == NULL)
      return false;
    if(i == index)
    {
      LinkNode<T> * m = new LinkNode<T>;
      m->data = x;
      m->next = n->next;
      n->next = m;
      return true;
    }
    n = n->next;
  }
  return false;
}

template <class T>
bool Link<T>::remove(int index, T & x)
{ //删除索引为 index 的元素, x 保存该元素的值
  LinkNode<T> * n = head;
  for(int i = 0; i <= index; i++)
  {
    if(n == NULL)
    {
      return false;
    }
    if(i == index)
    {
      LinkNode<T> * m = n->next;
      x = m->data;
      n->next = m->next;
      delete m;
      m = NULL;
      return true;
    }
    n = n->next;
  }
```



```
    }
    return false;
}
template <class T>
void Link<T>::output( )
{ //输出链表
    LinkNode<T> * n = head->next;
    while(n != NULL)
    {
        cout << n->data << " ";
        n = n->next;
    }
    cout << endl;
}
```

Owner 类

```
//-----业主的基本信息-----
class Owner
{
private:
    char name[20]; //业主的姓名
    int age; //业主的年龄
    char sex[4]; //业主的性别
    int year; //业主已居住的年限
    int amount; //业主家居住的人数
    char number[20]; //业主的电话号码
    int door; //业主的所在的楼栋
    char car[4]; //是否有车
    char marriage[4]; //婚姻状况，是否结婚
public:
    void setName(char name[]);
    char getName(char name[]);
    void setAge(int age);
    int getAge( );
    void setSex(char sex[]);
    char getSex(char sex[]);
    void setYear(int year);
    int getYear( );
    void setAmount(int amount);
    int getAmount( );
    void setNumber(char number[]);
    char getNumber(char number[]);
    void setDoor(int door);
    int getDoor( );
    void setCar(char car[]);
    char getCar(char car[]);
    void setMarriage(char marriage[]);
    char getMarriage(char marriage[]);
    void output(); //输出结果
};
void Owner::setName(char name[])
{
    strcpy(this->name, name);
}
```



```
}
char Owner::getName(char name[])
{
    strcpy(name, this->name);
}
void Owner::setAge(int age)
{
    this->age = age;
}
int Owner::getAge( )
{
    return age;
}
void Owner::setSex(char sex[])
{
    strcpy(this->sex, sex);
}
char Owner::getSex(char sex[])
{
    strcpy(sex, this->sex);
}
void Owner::setYear(int year)
{
    this->year = year;
}
int Owner::getYear( )
{
    return year;
}
void Owner::setAmount(int amount)
{
    this->amount = amount;
}
int Owner::getAmount( )
{
    return amount;
}
void Owner::setNumber(char number[])
{
    strcpy(this->number, number);
}
char Owner::getNumber(char number[])
{
    strcpy(number, this->number);
}
void Owner::setDoor(int door)
{
    this->door = door;
}
int Owner::getDoor( )
{
    return door;
}
void Owner::setCar(char car[])
{
    strcpy(this->car, car);
}
char Owner::getCar(char car[])
```



```
{
    strcpy(car, this->car);
}
void Owner::setMarriage(char marriage[])
{
    strcpy(this->marriage, marriage);
}
char Owner::getMarriage(char marriage[])
{
    strcpy(marriage, this->marriage);
}
void Owner::output( )
{
    cout << "【业主的姓名】： " << name << endl;
    cout << "【" << name << "的年龄】： " << age << endl;
    cout << "【" << name << "的性别】： " << sex << endl;
    cout << "【" << name << "的已居住年限】： " << year << endl;
    cout << "【" << name << "的家庭人数】： " << amount << endl;
    cout << "【" << name << "的电话号码】： " << number << endl;
    cout << "【" << name << "所在的楼栋单元】： " << door << endl;
    cout << "【" << name << "家是否有车】： " << car << endl;
    cout << "【" << name << "是否婚配】： " << marriage << endl;
}
```

页面展示 Show 类（此处未展示的函数实现代码见正文）

```
//-----总页面显示-----
class Show
{
private:
    Link <Owner> owns;
public:
    Show( );
    void mainShow( ); //主页面
    void key( ); //主页面上的选项跳转
    void infoShow( ); //展示业主信息
    void putShow( ); //添加业主信息
    void reShow( ); //修改业主信息
    void delShow( ); //删除业主信息
    void help( ); //对主页面下（1）~（5）相关服务进行一个解释
    void save( ); //保存业主的信息
    void read( ); //读取业主的信息
};
Show::Show( )
{
    mainShow( );
}
void Show::help( )
{
    system("cls");
    cout << endl;
    cout << "\t\t\t<------(7) 帮助菜单 ----->\n" << endl;
```



数学与统计学院课程设计(实习)报告

```
cout << "【温馨提示】：此页面为解释页面，如果您有任何的疑惑，可以从此页面下查询相关信息，解答您的疑惑，祝您使用愉快！\n" << endl;
cout << "【使用前提】：" << endl;
cout << "打开主页面前，需要输入正确的用户名和密码，为了确保信息的隐私，本系统设置的用户名和密码是唯一的哦。\\n" << endl;
cout << "【注意事项】：\\n" << endl;
cout << "1、您输入“年龄”、“已居住年限”的时候，请输入整数，默认单位为年(year)；\\n" << endl;
cout << "2、电话号码可以填业主的手机号码，也可以填业主家中的座机号码；\\n" << endl;
cout << "3、填“所在楼栋单元”时，为了统一格式，方便检查，请输入类似于‘021101’的形式；" << endl;
cout << "（其中，02表示业主所在的楼栋，表示第几栋；11表示业主家的楼层，表示第几层；01表示同层内业主所在的房间，表示第几个）\\n" << endl;
cout << "4、填“是否有车”和“是否婚配”时，请填‘是’或‘否’，不要填其他的词；\\n" << endl;
cout << "以上为【帮助菜单】内的全部内容。\\n" << endl;
cout << "即将返回至主页面！" << endl;
cout << endl;
system("pause");
}
```

main()函数

```
void mainPoint()
{
    enter();
    Show show;
    while(true)
    {
        system("cls");
        show.mainShow();
        show.key();
    }
}
int main()
{
    cout << endl;
    cout << "\\t\\t* * * * * * * * *欢迎您使用物业基本信息管理系统* * * * * * * * * *\\n" << endl;
    cout << endl;
    cout << "\\t \\t \\t* *进入系统之前，请您输入正确的用户名和密码 * *" << endl;
    cout << endl;
    mainPoint();
    return 0;
}
```