**综合专题学期实训**

题 目： 小区物业管理系统

班 级： 1701

学 号： 07，24,26

姓 名： 李春菊，周琛，陶逸辉

提交时间：

《综合专题学期实训**I**》评分表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 组号 | | | | | 7 | | | 题目 | | | 小区物业管理系统 | | | | | | | | |
| 1 | 学号 | | | | 2017115010107 | | | | | | 姓名 | | | | 李春菊 | | | | |
| 2 | 学号 | | | | 2017115010124 | | | | | | 姓名 | | | | 周琛 | | | | |
| 3 | 学号 | | | | 2017115010126 | | | | | | 姓名 | | | | 陶逸辉 | | | | |
| 共性评分 | | | | | | | | | | | | | | | | | | | |
| I | | | 评分内容 | | | 满分20分 | | | 评分内容 | | | | 满分50分 | | | 评分内容 | | 满分30分 | |
|  | | | 报告  规范 | | |  | | | 报告  内容 | | | |  | | | 报告  独创性 | |  | |
| II | | | 评分内容 | | | 满分50分 | | | 评分内容 | | | | 满分20分 | | | 评分内容 | | 满分30分 | |
|  | | | 运行正常，  功能正确 | | |  | | | 界面友好、操作方便 | | | |  | | | 扩展功能 | |  | |
| 个性评分 | | | | | | | | | | | | | | | | | | | |
| III | | | | | | | IV | | | | | | | | | | | | |
| 编号 | | 评分  内容 | | 满分  100分 | | | 评分  内容 | | | 满分  30分 | | 评分  内容 | | 满分  30分 | | | 评分  内容 | | 满分  40分 |
| 1 | | 实训  考勤 | |  | | | 编程  规范度 | | |  | | 团队  贡献 | |  | | | 答辩  情况 | |  |
| 2 | | 实训  考勤 | |  | | | 编程  规范度 | | |  | | 团队  贡献 | |  | | | 答辩  情况 | |  |
| 3 | | 实训  考勤 | |  | | | 编程  规范度 | | |  | | 团队  贡献 | |  | | | 答辩  情况 | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 教师签字 |  | 批阅时间 |  |

小区物业管理系统

1．题目要求

（1）居民信息管理：建立住户信息库，实现住户档案数据的计算机化管理，使居民的登记与查询简易而快速，对居民的信息查询应有一定的权限检测，按权限进行查询，分级管理，同是应有更新功能。

（2）物业收费管理：对纳入社区收费管理的收费项目进行电子收费管理（主要是指对煤气表、电表、水表的自动抄表与电子收费），提供应缴费用的查询、定期催缴、收费项目和计费方式的变更维护、定期统计收费情况等功能，同时将其他各子系统相应的收费信息递交收费管理中进行统一结算。

（3）来客访问管理：对出入小区的外来人员进行人员信息、出入信息的登记、汇总、监督管理，并提供查询功能。

（4）设备报修管理：对日常设备的报修信息进行登记，对设备报修，及其处理情况有查询、考核与统计等管理功能，定期产生考核情况明细，报修信息的录入具有多条途径，相应的费用通过收费管理功能进行统一的结算。

2．需求分析

住户信息、物业收费信息要用文件储存（来客访问管理、设备报修管理原理是一样的）因而提供文件的输入/输出操作；另外，还要提供键盘选择菜单以实现功能选择。

首先充分了解到用户的各个方面需求,包括现有的以及将来可能增加的需求，通过对洛阳周边小区进行详细的实地考察以及利用网络等多种渠道，了解到洛阳本地在小区物业管理方面的一-些现状以及业主、物业管理人员对小区物业管理系统的各种要求，业主可以查看缴费信息，查看公告信息,进而获得自己所需要知道的物业管理信息,因此查看这两种信息不能对其进行数据的修改、编辑;业主可以参与报修，将所需要维修的设备信息保存。

3.系统分析

该程序具有菜单栏，信息输入，信息搜索，信息查找，信息修改，信息删除，读取文件，录入文件等功能。

1. 菜单栏模块：使用printf，switch case语句编写菜单。
2. 信息输入模块：主要运用for语句，printf语句，scanf语句。
3. 信息搜索模块：主要运用if和scanf语句，可以通过编号来搜索信息。
4. 信息查找模块：主要运用if和scanf语句，可以通过名称来查找信息。
5. 信息修改模块：主要运用if，scanf，printf语句，可以通过输入编号修改信息。
6. 信息删除模块：主要运用if，for循环语句寻找需要删除的采购信息，然后使用scanf语句删除。
7. 读取文件模块：主要功能是读取外部文件数据，实现输入功能。
8. 录入文件模块：主要功能是将输入的信息存入外部文件。

4.主函数流程分析

启动主函数后，界面会首先出现菜单画面，然后根据提示选择所需的功能，通过switch语句调用相应的功能。

开始

显示菜单

输入a

输入错误

XinXi\_Menu（）

a=1

Tongji\_Menu（）

a=2

a=3

结束

a<0||a>3

主要功能函数分析

（1）Xinxi\_Menu（）

开始

输入b

输入错误

b<0||b>5

YeZhu\_Menu()

Init\_save\_yezhu()

b=1

WuYe\_Menu()

b=2

Foreign\_personnel()

b=3

Equipment\_record()

b=4

b=5

返回上一级

（2）Tongji\_Menu（）

开始

输入c

输入错误

c<0||c>3

myfind()

c=1

Arrears\_List()

c=2

c=3

返回上一级

（3）YeZhu\_Menu（）

开始

输入d

输入错误

d<0||d>6

MyAdd1\_Owner

()

d=1

MyReplace1\_Owner()

d=2

MyDelete1\_Owner ()

d=3

Display()

d=4

d=6

返回上一级

Save\_yezhu()

d=5

（4）WuYe\_Menu（）

开始

输入e

输入错误

e<0||e>5

MyAdd2\_Property ()

e=1

MyReplace\_Property()

e=2

MyDelete2\_Property()

e=3

Display2()

e=4

e=5

返回上一级

（5）Doreign\_personnel()

开始

输入 f

输入错误

f<0||f>5

MyAdd1\_Foreign\_personnel()

f=1

MyReplacel\_Foreign\_personnel()

f=2

MyDeletel\_Foreign\_personnel()

f=3

Display\_Foreign\_personnel()

f=4

f=5

返回上一级

（6）Equipment\_record()

开始

输入 g

输入错误

g<0||g>5

MyAdd1\_Equipment\_record()

g=1

MyReplacel\_Equipment\_record()

g=2

MyDeletel\_Equipment\_record()

g=3

Display\_Equipment\_record ()

g=4

g=5

返回上一级

。

5．功能模块图

**主 函 数**

物业信息管理 统计查询系统

业主信息管理 收费信息管理 来客访问系统 设备报修管理

信 信 信 信 信 查 欠

息 息 息 息 息 寻 费

录 修 删 浏 保 缴 名

入 改 除 览 存 费 单

6．数据结构

可以把住户信息、物业收费信息、来客访问管理、设备报修管理分别放在4个结构体内。

住户信息结构体：

struct yezhu

{

//业主结构体

char id[100];

char name[20];

int age;

int family\_num;

int floor\_num;//所在楼号

char room\_num[100];//单元房间号

float lost;

int area;

char rz\_time[20];

struct yezhu \*next;

};

物业收费结构体：

struct wuye

{ //物业信息结构体

char ID[100];

int last\_year;

int last\_month;

float money;

int to\_time;

struct wuye \*link;

};

来客访问管理结构体：

struct Foreign\_personnel

{

//来客访问管理结构体

char id[100];

char name[20];

int age;

char car\_License\_plate[100];//车牌号

int Parking\_space\_num;//停车位号

char come\_time[20];

char leave\_time[20];

struct Foreign\_personnel \*next;

};

设备报修管理结构体：

struct Equipment\_record

{

//设备报修管理结构体

char name[20];

char room\_num[100];

int floor\_num;

char time[20];//维修时间

char Whether\_deal\_with[100];//是否进行处理

struct Equipment\_record \*next;

};

7．程序运行结果

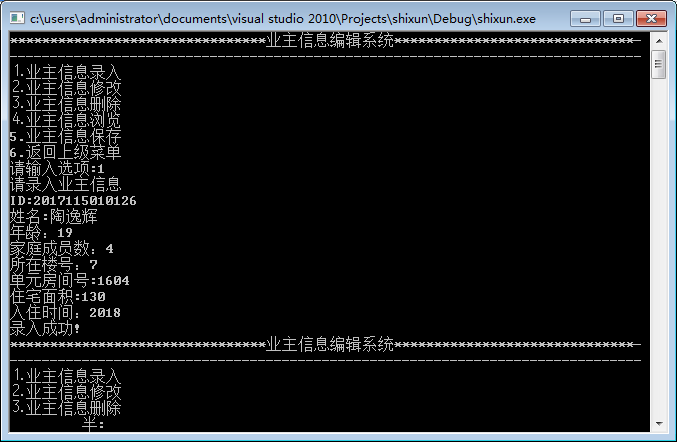


图1 业主信息录入

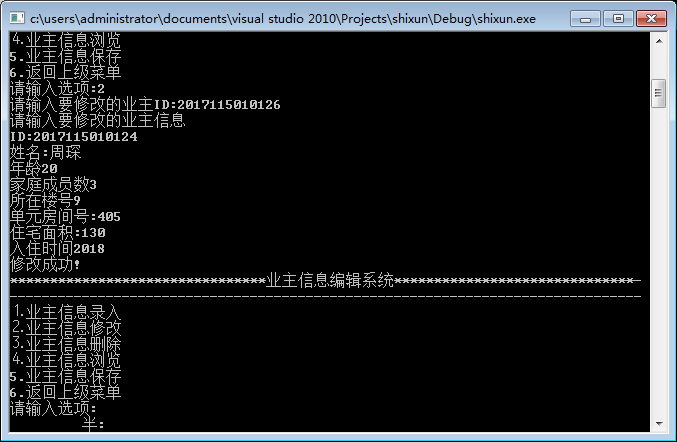


图2 业主信息修改

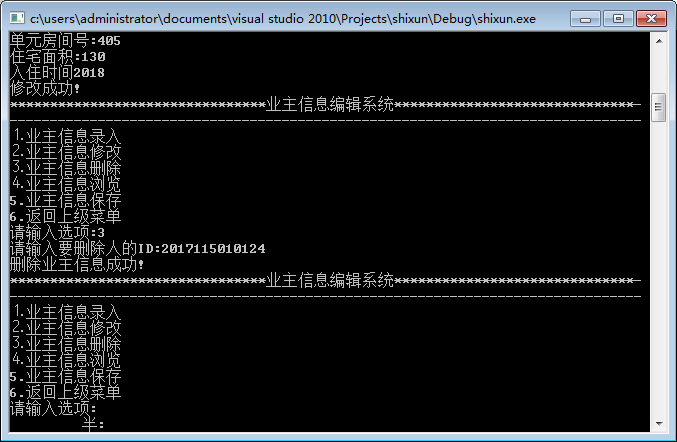


图3 业主信息删除

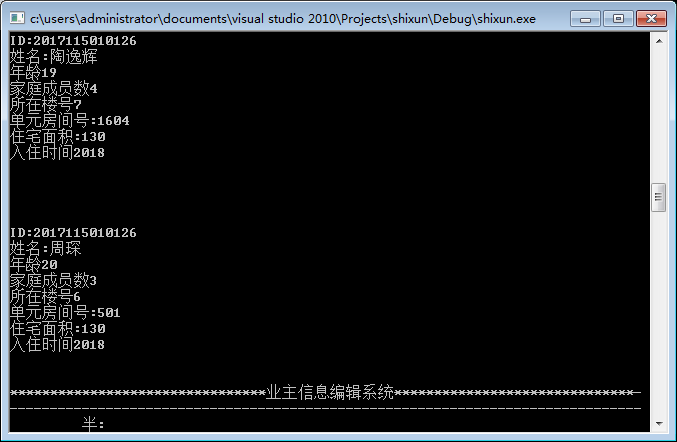


图4 业主信息浏览

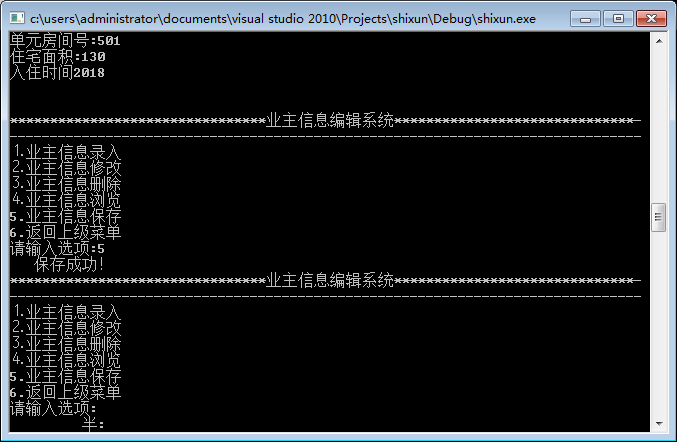


图5 业主信息保存

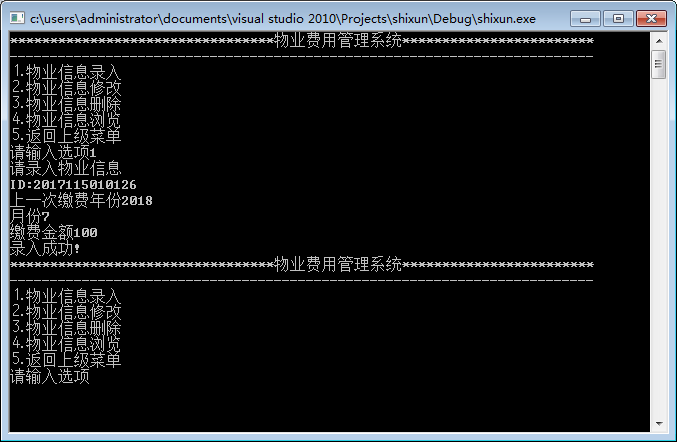


图6 物业信息录入

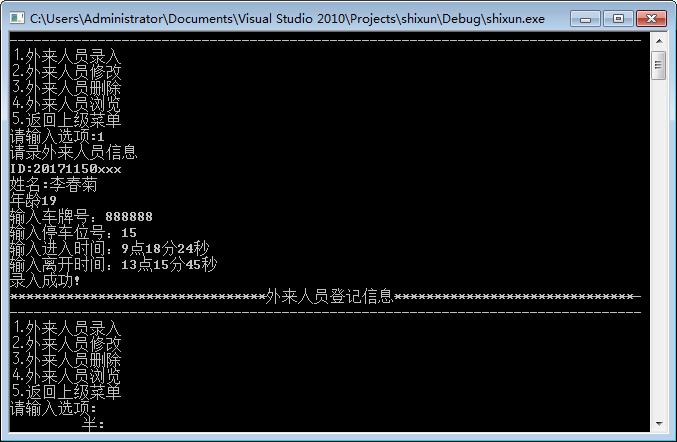


图7 外来人员信息录入

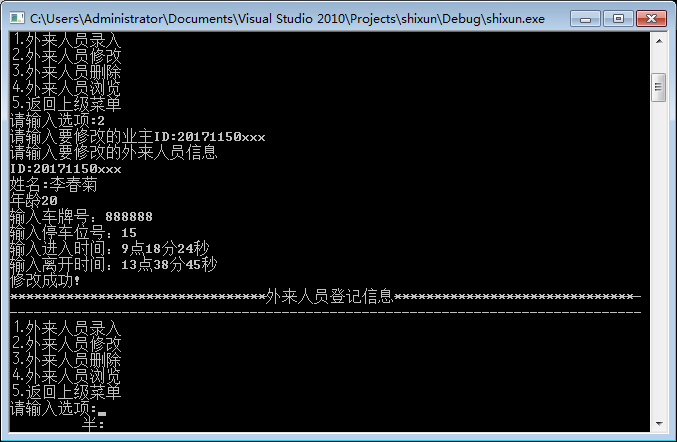
、

图8 外来人员信息修改

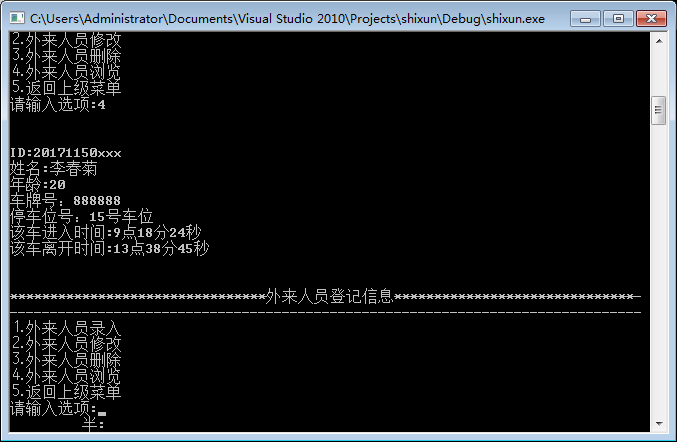


图9 外来人员信息浏览

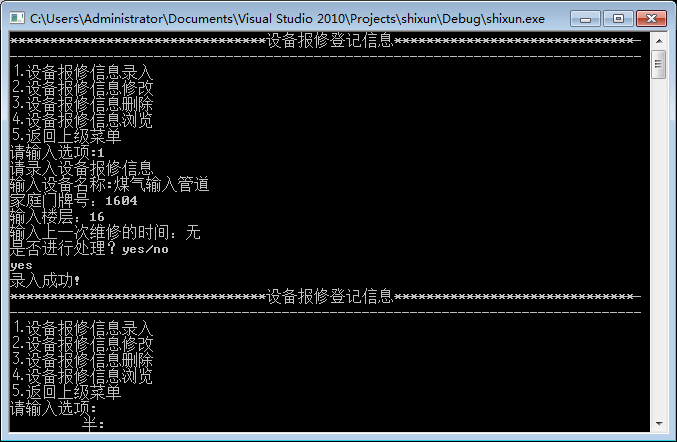


图10 设备保修信息录入

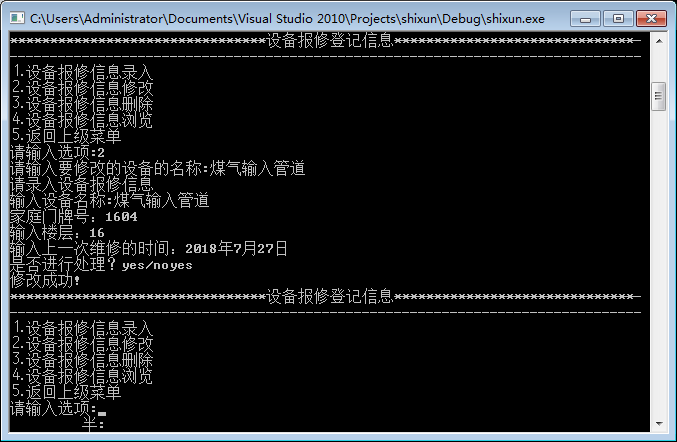


图11 设备保修信息修改

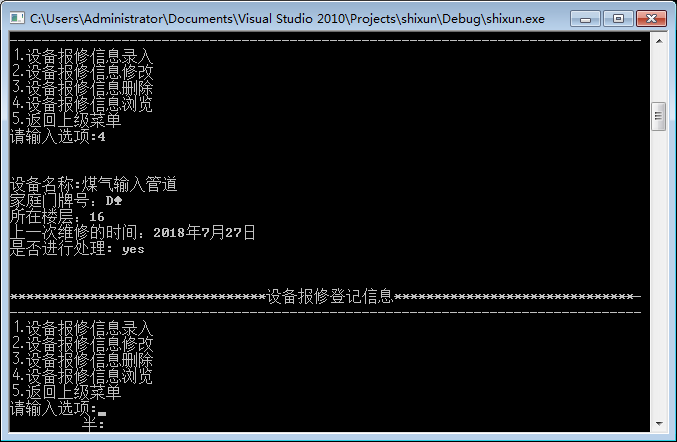


图12 设备保修信息浏览

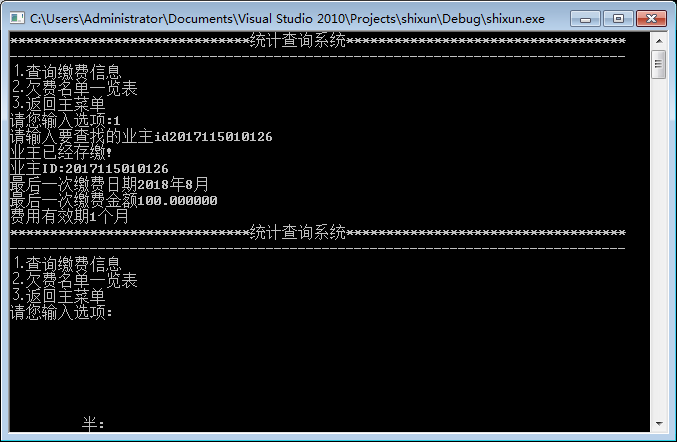


图13 查询缴费信息

8．总结

这次课程设计为期约3星期，虽然时间不长，但感受很深。此次实训是分为3个人一组来进行对C语言数据结构的学习与巩固，与之前的实训方式有所不同，对每个人的能力来完成相应的编码，实现分组完成一个完整的功能实现。无论是在学习知识，还是在解决问题的能力上都有了深刻的认识。 首先就是对理论知识的进一步巩固，再一次接触C语言的数据类型、语句、数组、指针结构等等一系列与c语言有关的东西，通过“庖丁解牛”，我开始懂得不管一个程序多么复杂，其实它都是由最基本的函数组成，只要我们弄清结构、一切都会迎刃而解。

编程是一项考验综合能力的事情，仅仅有课本知识还不够，必须要有自己的知识框架才能通过所学理论解决实际问题。尽管只有八天，但我却碰到许多问题，诸如无法抽象出问题模型，不知道用哪一个或哪几个知识点来解决问题等等，并且一开始设计的程序冗长复杂，效率低，通过与同学的交流和在老师的帮助下，让程序成功地运行，虽然不是自己全部编写的。说实话，我还没达到完全靠自己编写的能力。通过在网上找答案并进行修改调试，还是坚持了下来并取得了成功。此次实习很有意义，它将教会我不断的提高自己的时间能力，挖掘自己解决问题的能力。

9．附录

（源代码）

以下是head.h的源代码：

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <malloc.h>

#include <conio.h> //函数声明

struct yezhu

{

//业主结构体

char id[100];

char name[20];

int age;

int family\_num;

int floor\_num;//所在楼号

char room\_num[100];//单元房间号

float lost;

int area;

char rz\_time[20];

struct yezhu \*next;

};

struct wuye

{ //物业信息结构体

char ID[100];

int last\_year;

int last\_month;

float money;

int to\_time;

struct wuye \*link;

};

struct Foreign\_personnel

{

//外来人员结构体

char id[100];

char name[20];

int age;

char car\_License\_plate[100];//车牌号

int Parking\_space\_num;//停车位号

char come\_time[20];

char leave\_time[20];

struct Foreign\_personnel \*next;

};

struct Equipment\_record

{

//设备报修结构体

char name[20];

char room\_num[100];

int floor\_num;

char time[20];//维修时间

char Whether\_deal\_with[100];//是否进行处理

struct Equipment\_record \*next;

};

struct yezhu \*phead; //业主头结点

struct wuye \*pHead; //物业头结点

struct Foreign\_personnel \*phead\_Foreign; //外来人员头结点

struct Equipment\_record \*phead\_Equipment\_record; //设备维修管理头结点

以下是Array.h的源代码：

int Main\_Menu(); //主菜单

void XinXi\_Menu(); //物业信息菜单

void TongJi\_Menua(); //统计查询菜单

void Arrears\_List(); //欠费名单

void YeZhu\_Menu(); //业主菜单

void WuYe\_Menu(); //物业菜单

void Foreign\_personnel();//外来人员登记菜单

void Equipment\_record();//设备保修管理

void MyAdd1\_Owner(); //录入业主信息

void MyAdd2\_Property(); //录入物业信息

void MyAdd1\_Foreign\_personnel(); //录入外来人员信息

void MyAdd1\_Equipment\_record();//录入设备报修信息

void MyDelete1\_Owner(); //删除业主信息

void MyDelete2\_Property(); //删除物业信息

void MyDelete1\_Foreign\_personnel(); //删除外来人员信息

void MyDelete1\_Equipment\_record();

void MyReplace1\_Owner(); //修改业主信息

void MyReplace2\_Property(); //修改物业信息

void MyReplace1\_Foreign\_personnel();//修改外来人员信息

void MyReplace1\_Equipment\_record();

void Display1(); //浏览业主信息

void Display2(); //浏览物业信息

void Display\_Foreign\_personnel();//浏览外来人员信息

void Display\_Equipment\_record();

void myfind(); //查询缴费

void Display(struct wuye \*p);

void Read1();

void Read2();

void save1(struct yezhu \*p); //保存业主信息至文件

void save2(struct wuye \*p); //保存物业信息至文件

struct yezhu \*find1(char \*id); //查找业主信息

struct wuye \*find2(char \*id); //查找物业信息`

struct yezhu \*myAdd3(struct yezhu \*p); //添加业主信息

struct wuye \*myAdd4(struct wuye \*p);

struct Foreign\_personnel \*myAdd\_Foreign\_personnel(struct Foreign\_personnel \*p);

struct Equipment\_record \*myAdd\_Equipment\_record(struct Foreign\_personnel \*p);

void save\_yezhu();

void init\_save\_yezhu(); //初始化文件中业主的信息到链表中

以下是main.c的源代码：

#include"head.h"

#include"Array.h"

void main()

{

int m;

do

{

m = Main\_Menu();

switch (m)

{

case 1:

XinXi\_Menu();

break;

case 2:

TongJi\_Menua();

break;

case 3:

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*谢谢使用再见!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

default:

printf("输入错误请重新输入!\n\n");

exit(0);

}

} while (1);

}

以下是Array.c的源代码：

#include"head.h"

#include"Array.h"

//主菜单

int Main\_Menu()

{

int n;

system("cls");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*欢迎进入小区物业管理系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("-----------------------------------------------------------------------------\n");

printf("⒈物业信息系统 \n");

printf("⒉统计查询系统 \n");

printf("⒊退出程序\n");

printf("请您输入选项:");

scanf("%d",&n);

return n;

}

//物业信息菜单显示

void XinXi\_Menu()

{

int choose;

system("cls");

do

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*物业信息系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("-----------------------------------------------------------------------------\n");

printf("1.居民信息管理\n");

printf("2.物业收费管理\n");

printf("3.来客访问管理\n");

printf("4.设备报修管理\n");

printf("5.返回主菜单\n");

printf("请您输入选项:");

scanf("%d",&choose);

switch(choose)

{

case 1:

init\_save\_yezhu();

YeZhu\_Menu();

break;

case 2:

WuYe\_Menu();

break;

case 3:

Foreign\_personnel();

break;

case 4:

Equipment\_record();

break;

case 5:

return;

default:

printf("输入错误请重新输入\n");

}

} while (1);

}

//统计查询菜单显示

void TongJi\_Menua()

{

int choose;

system("cls");

do

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*统计查询系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("-----------------------------------------------------------------------------\n");

printf("⒈查询缴费信息\n");

printf("⒉欠费名单一览表\n");

printf("⒊返回主菜单 \n");

printf("请您输入选项:");

scanf("%d",&choose);

switch(choose)

{

case 1:

myfind();

break;

case 2:

Arrears\_List();

break;

case 3:

return;

default:

printf("输入错误请重新输入!\n\n");

}

} while (1);

}

//业主菜单函数

void YeZhu\_Menu()

{

int choose;

system("cls");

do

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*业主信息编辑系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-\n");

printf("-------------------------------------------------------------------------------\n");

printf("⒈业主信息录入 \n");

printf("⒉业主信息修改 \n");

printf("⒊业主信息删除\n");

printf("⒋业主信息浏览 \n");

printf("5.业主信息保存\n");

printf("6.返回上级菜单\n");

printf("请输入选项:");

scanf("%d",&choose);

//getchar();

switch(choose)

{

case 1:

MyAdd1\_Owner();

break;

case 2:

MyReplace1\_Owner();

break;

case 3:

MyDelete1\_Owner();

break;

case 4:

Display1();

break;

case 5:

save\_yezhu();

break;

case 6:

return;

default:

printf("输入错误请重新输入\n");

}

} while (1);

}

void init\_save\_yezhu()

{

FILE \*fp;

int n=1;

struct yezhu \*p,\*q;

// pHead=NULL;

fp=fopen("yezhu.txt","r");

if(fp==NULL)

{

return;

}

else //若文件不为空

{

while(!feof(fp)) //遍历文件

{

{

if(n==1)//作用是：首节点读入数据。

{

p=(struct yezhu\*)malloc(sizeof(struct yezhu));//开辟一个链表节点。

phead=p;

q=p; // 即pHead,p,q指向同一个节点，此节点作为首节点。

q->next=NULL;

// p=(nodelist\*)malloc(sizeof(nodelist)); //新链表节点 先开辟空间

fscanf(fp,"%s\n",(p->id));

fscanf(fp,"%s\n",(p->name));

fscanf(fp,"%d\n",&(p->age));

fscanf(fp,"%d\n",&(p->family\_num));

fscanf(fp,"%d\n",&(p->floor\_num));

fscanf(fp,"%s\n",(p->room\_num));

fscanf(fp,"%d\n",&(p->area));

fscanf(fp,"%s\n",(p->rz\_time));

//p->result=getchar();

n=2;

/\*fscanf(fp,"%d\n%s\n%s\n%s\n%s\n%s\n%c",&(p->id),p->question,p->option1,

p->option2,p->option3,p->option4,&(p->result));//扫描出文件中数据，用p接受

/\*q->next=p;

q=q->next;

pHead=p;\*/

}

else if(n==2)//创建单链表

{

p=(struct yezhu\*)malloc(sizeof(struct yezhu)); //新链表节点 先开辟空间

fscanf(fp,"%s\n",(p->id));

fscanf(fp,"%s\n",(p->name));

fscanf(fp,"%d\n",&(p->age));

fscanf(fp,"%d\n",&(p->family\_num));

fscanf(fp,"%d\n",&(p->floor\_num));

fscanf(fp,"%s\n",(p->room\_num));

fscanf(fp,"%d\n",&(p->area));

fscanf(fp,"%s\n",(p->rz\_time));

/\*fscanf(fp,"%d\n%s\n%s\n%s\n%s\n%s\n%c",&(p->id),p->question,p->option1,

p->option2,p->option3,p->option4,&(p->result));\*///扫描出文件中数据，用p接受

q->next=p;

q=q->next;

}

}

q->next=NULL;

}

}

fclose(fp);

}

//录入业主信息

void MyAdd1\_Owner()

{

struct yezhu \*p;

p = (struct yezhu \*)malloc(sizeof(struct yezhu));

printf("请录入业主信息\n");

printf("ID:");

scanf("%s", p->id);

printf("姓名:");

scanf("%s", p->name);

printf("年龄：");

scanf("%d", &p->age);

printf("家庭成员数：");

scanf("%d", &p->family\_num);

printf("所在楼号：");

scanf("%d", &p -> floor\_num);

printf("单元房间号:");

scanf("%s", p->room\_num);

printf("住宅面积:");

scanf("%d", &p->area);

printf("入住时间：");

scanf("%s", p->rz\_time);

myAdd3(p);

printf("录入成功!\n");

}

//添加业主信息结点

struct yezhu \*myAdd3(struct yezhu \*pNew)

{

struct yezhu \*p,\*q;

if (phead == NULL)

{

phead = pNew;

pNew->next = NULL;

}

else

{

//将地址为pNew的节点插入到首地址为pHead的链表的尾部

/\*p1->next=pNew;

p1=pNew;

pNew->next=NULL;\*/

p=phead;

//q=(nodelist \*)malloc(sizeof(nodelist));

if(p->next!=0) //文件不为空，即pHead后面有数据

{

while(p->next!=0)

{

q=p->next;

p=q;

}

p->next=pNew;

pNew->next=NULL;

}

else if (p->next==0) //只有头结点，即pHead后面无数据

{

//将地址为pNew的节点插入到首地址为pHead的链表的尾部

p->next=pNew;

p=pNew;

pNew->next=NULL;

}

}

}

//修改业主信息

void MyReplace1\_Owner()

{

struct yezhu \*k;

char id[100];

printf("请输入要修改的业主ID:");

scanf("%s", id);

k = find1(id);

if (k != NULL)

{

printf("请输入要修改的业主信息\n");

printf("ID:");

scanf("%s", k->id);

printf("姓名:");

scanf("%s", k->name);

printf("年龄");

scanf("%d", &k->age);

printf("家庭成员数");

scanf("%d", &k->family\_num);

printf("所在楼号");

scanf("%d", &k->floor\_num);

printf("单元房间号:");

scanf("%s", k->room\_num);

printf("住宅面积:");

scanf("%d", &k->area);

printf("入住时间");

scanf("%s", k->rz\_time);

printf("修改成功!\n");

}

else

{

printf("无此业主ID!\n");

return;

}

}

//通过ID查找业主信息

struct yezhu \*find1(char \*id)

{

struct yezhu \*p;

p = phead;

while (p != NULL)

{

if (strcmp(p->id, id) == 0)

return p;

p = p->next;

}

return NULL;

}

struct Foreign\_personnel \*Foreign\_personnel\_find(char \*id)

{

struct Foreign\_personnel \*p;

p = phead\_Foreign;

while (p != NULL)

{

if (strcmp(p->id, id) == 0)

return p;

p = p->next;

}

return NULL;

}

struct Equipment\_record \*Equipment\_record\_find(char \*name)

{

struct Equipment\_record \*p;

p = phead\_Equipment\_record;

while (p != NULL)

{

if (strcmp(p->name, name) == 0)

return p;

p = p->next;

}

return NULL;

}

void Display1()

{

struct yezhu \*p;

p = phead;

while (p != NULL)

{

printf("\n\n");

printf("ID:%s\n", p->id);

printf("姓名:%s\n", p -> name);

printf("年龄%d\n", p->age);

printf("家庭成员数%d\n", p->family\_num);

printf("所在楼号%d\n", p->floor\_num);

printf("单元房间号:%s\n", p->room\_num);

printf("住宅面积:%d\n", p->area);

printf("入住时间%s\n", p->rz\_time);

printf("\n\n");

p = p->next;

}

}

void MyDelete1\_Owner()

{

struct yezhu \*p, \*pp;

struct wuye \*k, \*kk;

char id[100];

pp = phead;

kk = pHead;

printf("请输入要删除人的ID:");

scanf("%s", id);

p = find1(id);

k = find2(id);

if (p != NULL)

{

if (p == phead)

{

phead = p->next;

free(p);

}

else

{

while (pp->next != p)

pp = pp->next;

pp->next = p->next;

free(p);

}

}

else

{

printf("无此业主ID!\n");

return;

}

printf("删除业主信息成功!\n");

if (k != NULL)

{

if (k == pHead)

{

pHead = k->link;

free(k);

}

else

{

while (kk->link != k)

kk = kk->link;

kk->link = k->link;

free(k);

}

}

save2(k);

}

void save\_yezhu()

{

struct yezhu \*p=phead; //第一道试题（结构体）的地址

//重定向，标准的输出stdout不是显示器，而是指定的文件。

//以后的printf语句会输出到指定的文件中

freopen("yezhu.txt","w",stdout);

while(p!=NULL)

{

printf("%s\n", p->id);

printf("%s\n", p ->name);

printf("%d\n", p->age);

printf("%d\n", p->family\_num);

printf("%d\n", p->floor\_num);

printf("%s\n", p->room\_num);

printf("%d\n", p->area);

printf("%s\n", p->rz\_time);

p=p->next;

}

fclose(stdout);

freopen( "CON", "w", stdout );

printf(" 保存成功！\n");

/\*nodelist \*p=pHead;

FILE \*fp;

if ((fp = fopen("Questions.txt", "ab+")) == NULL)

{

printf("无法打开 Questions.txt!\n");

}

while (p != NULL)

{

fwrite(p, sizeof(nodelist), 1, fp);

p = p->next;

}

fclose(fp);\*/

/\*

printf("\n是否返回上一级Y?");

scanf("%c",&choose);

getchar();

if(choose=='Y')

system("cls");

showchildMenu();

\*/

}

void WuYe\_Menu()

{

int choose = 0;

system("cls");

do

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*物业费用管理系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("-------------------------------------------------------------------------\n");

printf("⒈物业信息录入 \n");

printf("⒉物业信息修改 \n");

printf("⒊物业信息删除 \n");

printf("⒋物业信息浏览 \n");

printf("⒌返回上级菜单\n");

printf("请输入选项");

scanf("%d",&choose);

switch(choose)

{

case 1:

MyAdd2\_Property();

break;

case 2:

MyReplace2\_Property();

break;

case 3:

MyDelete2\_Property();

break;

case 4:

Display2();

break;

case 5:

return;

default:

printf("输入错误请重新输入\n");

}

} while (1);

}

void MyAdd2\_Property() //录入物业信息//

{

struct wuye \*p;

struct yezhu \*t;

int x;

p = (struct wuye \*)malloc(sizeof(struct wuye));

printf("请录入物业信息\n");

printf("ID:");

scanf("%s", p->ID);

t = find1(p->ID);

if (t != NULL)

{

printf("上一次缴费年份");

scanf("%d", &p->last\_year);

printf("月份");

scanf("%d", &p->last\_month);

printf("缴费金额");

scanf("%f", &p->money);

x = p->money / (0.4 \* t->area);

p->to\_time = (int)x;

myAdd4(p);

save2(p);

printf("录入成功!\n");

}

else

{

printf("无此业主ID!\n");

return;

}

}

struct wuye \*myAdd4(struct wuye \*p) //添加物业信息结点//

{

struct wuye \*t;

if (pHead == NULL)

{

pHead = p;

p->link = NULL;

}

else

{

t = pHead;

while (t->link != NULL)

t = t->link;

t->link = p;

p->link = NULL;

}

return pHead;

}

struct wuye \*find2(char \*id) //通过姓名查找缴费信息

{

struct wuye \*p;

p = pHead;

while (p != NULL)

{

if (strcmp(p->ID, id) == 0)

return p;

p = p->link;

}

return NULL;

}

void myfind()

{

struct wuye \*p;

char a[100];

printf("请输入要查找的业主id");

scanf("%s",a);

p=find2(a);

if(p!=NULL)

{

printf("业主已经存缴!\n");

}

else

{

printf("业主尚没存缴信息!\n");

return;

}

Display(p);

}

void MyReplace2\_Property()

{

char id[20];

int x;

struct wuye \*k;

struct yezhu \*t;

printf("请输入要修改的业主ID:");

scanf("%s", id);

if (k != NULL)

{

printf("您要修改的信息为\n");

printf("请输入新的信息\n");

printf("ID:");

scanf("%s", k->ID);

printf("上一次缴费年份");

scanf("%d", &k->last\_year);

printf("月份");

scanf("%d", &k->last\_month);

printf("缴费金额");

scanf("%f", &k->money);

t = find1(k->ID);

x = (int)k->money / (0.4 \* t->area);

k->to\_time = (int)x;

myAdd4(k);

save2(k);

printf("修改成功!\n");

}

else

{

printf("无此业主ID!\n");

return;

}

}

void MyDelete2\_Property()

{

struct wuye \*p, \*pp;

char id[100];

pp = pHead;

printf("请输入要删除人的ID:");

scanf("%s", id);

p = find2(id);

if (p != NULL)

{

if (p == pHead)

{

pHead = p->link;

free(p);

}

else

{

while (pp->link != p)

pp = pp->link;

pp->link = p->link;

free(p);

}

}

else

{

printf("无此业主ID!\n");

return;

}

save2(p);

printf("删除成功!\n");

}

void Display2()

{

struct wuye \*p;

p = pHead;

while (p != NULL)

{

printf("ID:%s\n", p->ID);

printf("最后一次缴费日期%d年%d月\n",p->last\_year,p->last\_month);

printf("缴费金额:%f元\n",p->money);

printf("费用有效期%d个月\n",p->to\_time);

printf("\n\n\n");

p=p->link;

}

}

void Display(struct wuye \*p)

{

printf("业主ID:%s\n", p->ID);

printf("最后一次缴费日期%d年%d月\n",p->last\_year,p->last\_month);

printf("最后一次缴费金额%f\n",p->money);

printf("费用有效期%d个月\n",p->to\_time);

}

void Read2()

{

FILE \*fp;

struct wuye \*p;

fp = fopen("wuye.dat", "rb");

while (1)

{

p = (struct wuye \*)malloc(sizeof(struct wuye));

if (feof(fp))

break;

if (fread(p, sizeof(struct wuye), 1, fp))

myAdd4(p);

}

fclose(fp);

}

void save2(struct wuye \*p)

{

FILE \*fp;

fp = fopen("wuye.dat", "wb");

p = pHead;

while (p != NULL)

{

fwrite(p, sizeof(struct wuye), 1, fp);

p = p->link;

}

fclose(fp);

}

void Arrears\_List()

{

int year, month;

struct wuye \*k;

struct yezhu \*l;

k = pHead;

l = phead;

printf("请输入当前年份");

scanf("%d", &year);

printf("请输入当前月份");

scanf("%d", &month);

while (k != NULL)

{

if (year > k->last\_year)

{

l = find1(k->ID);

l->lost = (float)(month + 12 - k->to\_time - k->last\_month) \* (0.4 \* l->area);

if (l->lost > 0)

{

printf("业主ID\*:%s\n", l->id);

printf("业主姓名\*:%s\n", l->name);

printf("最后一次缴费日期%d年%d月\n",k->last\_year,k->last\_month);

printf("欠费金额\*%f\n",l->lost);

printf("\n\n");

}

}

if (year == k->last\_year)

{

l = find1(k->ID);

l->lost = (float)(month - k->to\_time - k->last\_month) \* (0.4 \* l->area);

if (l->lost > 0)

{

printf("业主ID:%s\n", l->id);

printf("业主姓名%s\n", l->name);

printf("最后一次缴费日期%d年%d月\n", k->last\_year, k->last\_month);

printf("欠费金额%f\n", l->lost);

printf("\n\n");

}

}

k = k->link;

}

}

void Foreign\_personnel()

{

int choose;

system("cls");

do

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*外来人员登记信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-\n");

printf("-------------------------------------------------------------------------------\n");

printf("⒈外来人员录入 \n");

printf("⒉外来人员修改 \n");

printf("⒊外来人员删除\n");

printf("⒋外来人员浏览 \n");

printf("⒌返回上级菜单\n");

printf("请输入选项:");

scanf("%d",&choose);

//getchar();

switch(choose)

{

case 1:

MyAdd1\_Foreign\_personnel();

break;

case 2:

MyReplace1\_Foreign\_personnel();

break;

case 3:

MyDelete1\_Foreign\_personnel();

break;

case 4:

Display\_Foreign\_personnel();

break;

case 5:

return;

default:

printf("输入错误请重新输入\n");

}

} while (1);

}

//录入外来人员信息

void MyAdd1\_Foreign\_personnel()

{

struct Foreign\_personnel \*p;

p = (struct Foreign\_personnel\*)malloc(sizeof(struct Foreign\_personnel));

printf("请录外来人员信息\n");

printf("ID:");

scanf("%s", p->id);

printf("姓名:");

scanf("%s", p->name);

printf("年龄");

scanf("%d", &p->age);

printf("输入车牌号：");

scanf("%s", p->car\_License\_plate);

printf("输入停车位号：");

scanf("%d", &p-> Parking\_space\_num);

printf("输入进入时间：");

scanf("%s", p->come\_time);

printf("输入离开时间：");

scanf("%s", p->leave\_time);

myAdd\_Foreign\_personnel(p);

//save1\_Foreign\_personnel(p);

printf("录入成功!\n");

}

//添加外来人员信息结点

struct Foreign\_personnel \*myAdd\_Foreign\_personnel(struct Foreign\_personnel \*p)

{

struct Foreign\_personnel \*t;

if (phead\_Foreign == NULL)

{

phead\_Foreign = p;

p->next = NULL;

}

else

{

t = phead\_Foreign;

while (t->next != NULL)

t = t->next;

t->next = p;

p->next = NULL;

return phead\_Foreign;

}

}

//修改外来人员信息

void MyReplace1\_Foreign\_personnel()

{

struct Foreign\_personnel \*k;

char id[100];

printf("请输入要修改的业主ID:");

scanf("%s", id);

k = Foreign\_personnel\_find(id);

if (k != NULL)

{

printf("请输入要修改的外来人员信息\n");

printf("ID:");

scanf("%s", k->id);

printf("姓名:");

scanf("%s", k->name);

printf("年龄");

scanf("%d", &k->age);

printf("输入车牌号：");

scanf("%s", k->car\_License\_plate);

printf("输入停车位号：");

scanf("%d", &k-> Parking\_space\_num);

printf("输入进入时间：");

scanf("%s", k->come\_time);

printf("输入离开时间：");

scanf("%s", k->leave\_time);

printf("修改成功!\n");

}

else

{

printf("无此业主ID!\n");

return;

}

}

void Display\_Foreign\_personnel()

{

struct Foreign\_personnel \*p;

p = phead\_Foreign;

while (p != NULL)

{

printf("\n\n");

printf("ID:%s\t\n", p->id);

printf("姓名:%s\t\n", p -> name);

printf("年龄:%d\t\n", p->age);

printf("车牌号：%s\t\n",p->car\_License\_plate);

printf("停车位号：%d号车位\t\n",p->Parking\_space\_num);

printf("该车进入时间:%s\t\n",p->come\_time);

printf("该车离开时间:%s\t\n",p->leave\_time);

printf("\n\n");

p = p->next;

}

}

void MyDelete1\_Foreign\_personnel()

{

struct Foreign\_personnel \*p, \*pp;

struct wuye \*k, \*kk;

char id[100];

pp = phead\_Foreign;

kk = pHead;

printf("请输入要删除人的ID:");

scanf("%s", id);

p = Foreign\_personnel\_find(id);

k = find2(id);

if (p != NULL)

{

if (p == phead\_Foreign)

{

phead\_Foreign = p->next;

free(p);

}

else

{

while (pp->next != p)

pp = pp->next;

pp->next = p->next;

free(p);

}

}

else

{

printf("无此外来人员ID!\n");

return;

}

printf("删除此外来人员信息成功!\n");

if (k != NULL)

{

if (k == pHead)

{

pHead = k->link;

free(k);

}

else

{

while (kk->link != k)

kk = kk->link;

kk->link = k->link;

free(k);

}

}

}

void Equipment\_record()

{

int choose;

system("cls");

do

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*设备报修登记信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-\n");

printf("-------------------------------------------------------------------------------\n");

printf("⒈设备报修信息录入 \n");

printf("⒉设备报修信息修改 \n");

printf("⒊设备报修信息删除\n");

printf("⒋设备报修信息浏览\n");

printf("⒌返回上级菜单\n");

printf("请输入选项:");

scanf("%d",&choose);

//getchar();

switch(choose)

{

case 1:

MyAdd1\_Equipment\_record();

break;

case 2:

MyReplace1\_Equipment\_record();

break;

case 3:

MyDelete1\_Equipment\_record();

break;

case 4:

Display\_Equipment\_record();

break;

case 5:

return;

default:

printf("输入错误请重新输入\n");

}

} while (1);

}

//录入设备信息

void MyAdd1\_Equipment\_record()

{

struct Equipment\_record \*p;

p = (struct Equipment\_record\*)malloc(sizeof(struct Equipment\_record));

printf("请录入设备报修信息\n");

printf("输入设备名称:");

scanf("%s",p->name);

printf("家庭门牌号：");

scanf("%s", p->room\_num);

printf("输入楼层：");

scanf("%d", &p->floor\_num);

printf("输入上一次维修的时间：");

scanf("%s",p->time );

printf("是否进行处理？yes/no \n");

scanf("%s",p->Whether\_deal\_with);

myAdd\_Equipment\_record(p);

//save1\_Foreign\_personnel(p);

printf("录入成功!\n");

}

//添加设备报修信息结点

struct Equipment\_record \*myAdd\_Equipment\_record(struct Equipment\_record \*p)

{

struct Equipment\_record \*t;

if (phead\_Equipment\_record == NULL)

{

phead\_Equipment\_record = p;

p->next = NULL;

}

else

{

t = phead\_Equipment\_record;

while (t->next != NULL)

t = t->next;

t->next = p;

p->next = NULL;

return phead\_Equipment\_record;

}

}

//修改设备报修信息

void MyReplace1\_Equipment\_record()

{

struct Equipment\_record \*k;

char name[100];

printf("请输入要修改的设备的名称:");

scanf("%s", name);

k = Equipment\_record\_find(name);

if (k != NULL)

{

printf("请录入设备报修信息\n");

printf("输入设备名称:");

scanf("%s",k->name);

printf("家庭门牌号：");

scanf("%d", k->room\_num);

printf("输入楼层：");

scanf("%d", &k->floor\_num);

printf("输入上一次维修的时间：");

scanf("%s",k->time );

printf("是否进行处理？yes/no");

scanf("%s",k->Whether\_deal\_with);

printf("修改成功!\n");

}

else

{

printf("无此设备名称!\n");

return;

}

}

void Display\_Equipment\_record()

{

struct Equipment\_record \*p;

p = phead\_Equipment\_record;

while (p != NULL)

{

printf("\n\n");

printf("设备名称:%s\t\n", p -> name);

printf("家庭门牌号：%s\t\n",p->room\_num);

printf("所在楼层：%d\t\n",p->floor\_num);

printf("上一次维修的时间：%s\t\n",p->time);

printf("是否进行处理: %s\t\n",p->Whether\_deal\_with);

printf("\n\n");

p = p->next;

}

}

void MyDelete1\_Equipment\_record()

{

struct Equipment\_record \*p, \*pp;

struct wuye \*k, \*kk;

char name[100];

pp = phead\_Equipment\_record;

kk = pHead;

printf("请输入要删除设备的名称:");

scanf("%s", name);

p = Equipment\_record\_find(name);

k = find2(name);

if (p != NULL)

{

if (p == phead\_Equipment\_record)

{

phead\_Equipment\_record = p->next;

free(p);

}

else

{

while (pp->next != p)

pp = pp->next;

pp->next = p->next;

free(p);

}

}

else

{

printf("无此设备ID!\n");

return;

}

printf("删除此设备成功!\n");

if (k != NULL)

{

if (k == pHead)

{

pHead = k->link;

free(k);

}

else

{

while (kk->link != k)

kk = kk->link;

kk->link = k->link;

free(k);

}

}

}