

南京航空航天大学

《程序设计》课程设计报告

客房管理系统

学号：071830118

姓名：顾尧

日期：2020-05-20

指导老师：郑洪源

目录

| | | |
|----|-----------------------------|-------|
| 一、 | 需求分析 | 3 |
| 二、 | 系统总框图 | 4-5 |
| 三、 | 每个模块设计分析 | 6-14 |
| 四、 | 所有类及定义的函数说明 | 15-19 |
| 五、 | 特色函数算法 | 20-23 |
| 六、 | 存在的不足与对策 | 23 |
| 七、 | 使用说明 | 24-28 |
| 八、 | 程序源代码 | 29-66 |

需求分析

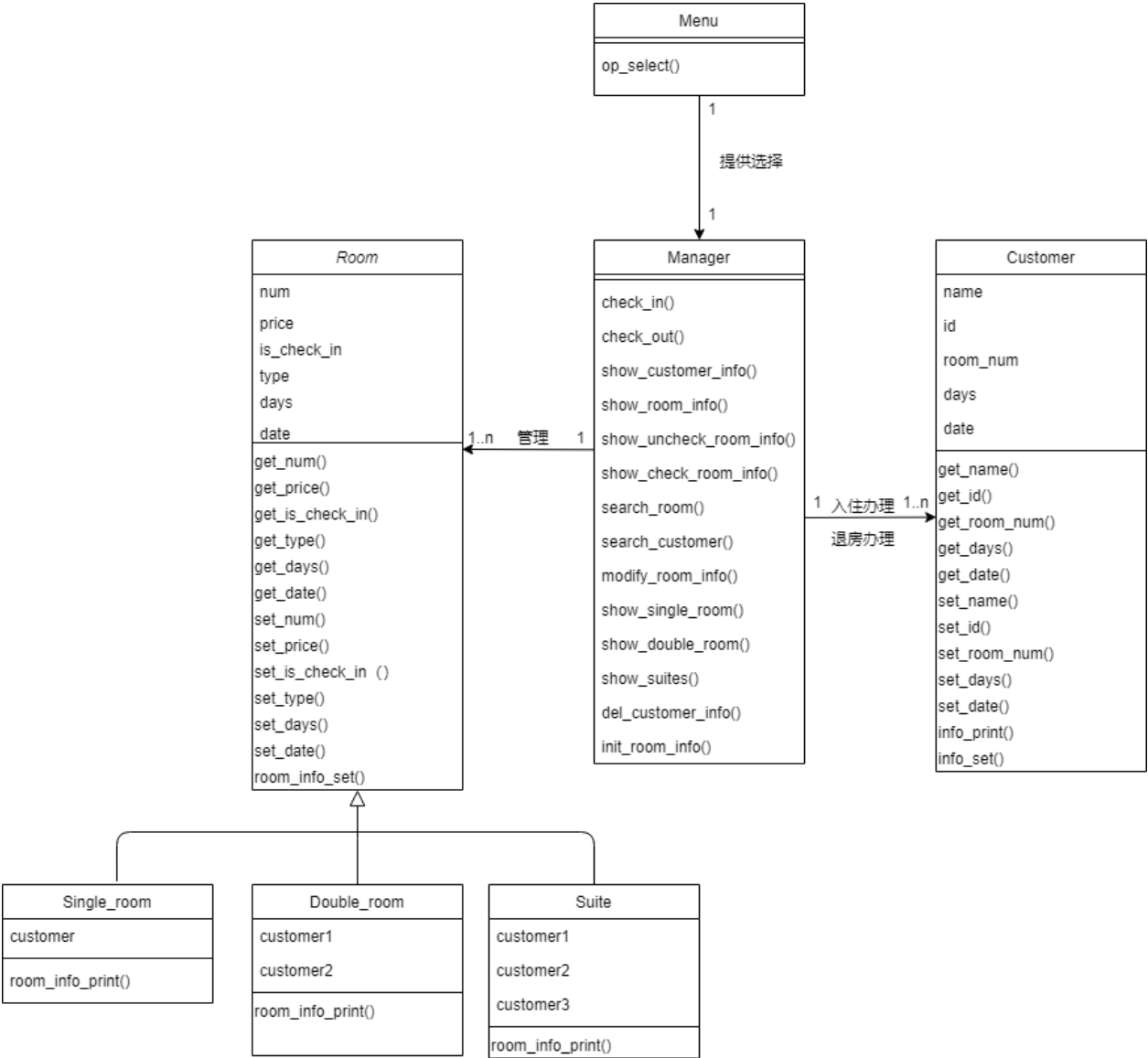
编写一个客房管理系统，该系统提供键盘式的功能选择。功能包括：客人入住办理；客人退房办理；显示所有房间信息；显示所有空闲房间信息；显示所有已入住房间信息；显示所有客人信息；输入房间号和价格区间查询房间信息；输入客人姓名或身份证号查询客人信息；修改房间价格，房间类型，房间号等功能。

其中所有房间信息和顾客信息存入文件中。运行各功能模块时能给出提示信息帮助管理员完成操作，当管理员的输入不合法时给出相应的提示信息，并回到重新输入界面。程序运行出错时能给出相应的提示信息。

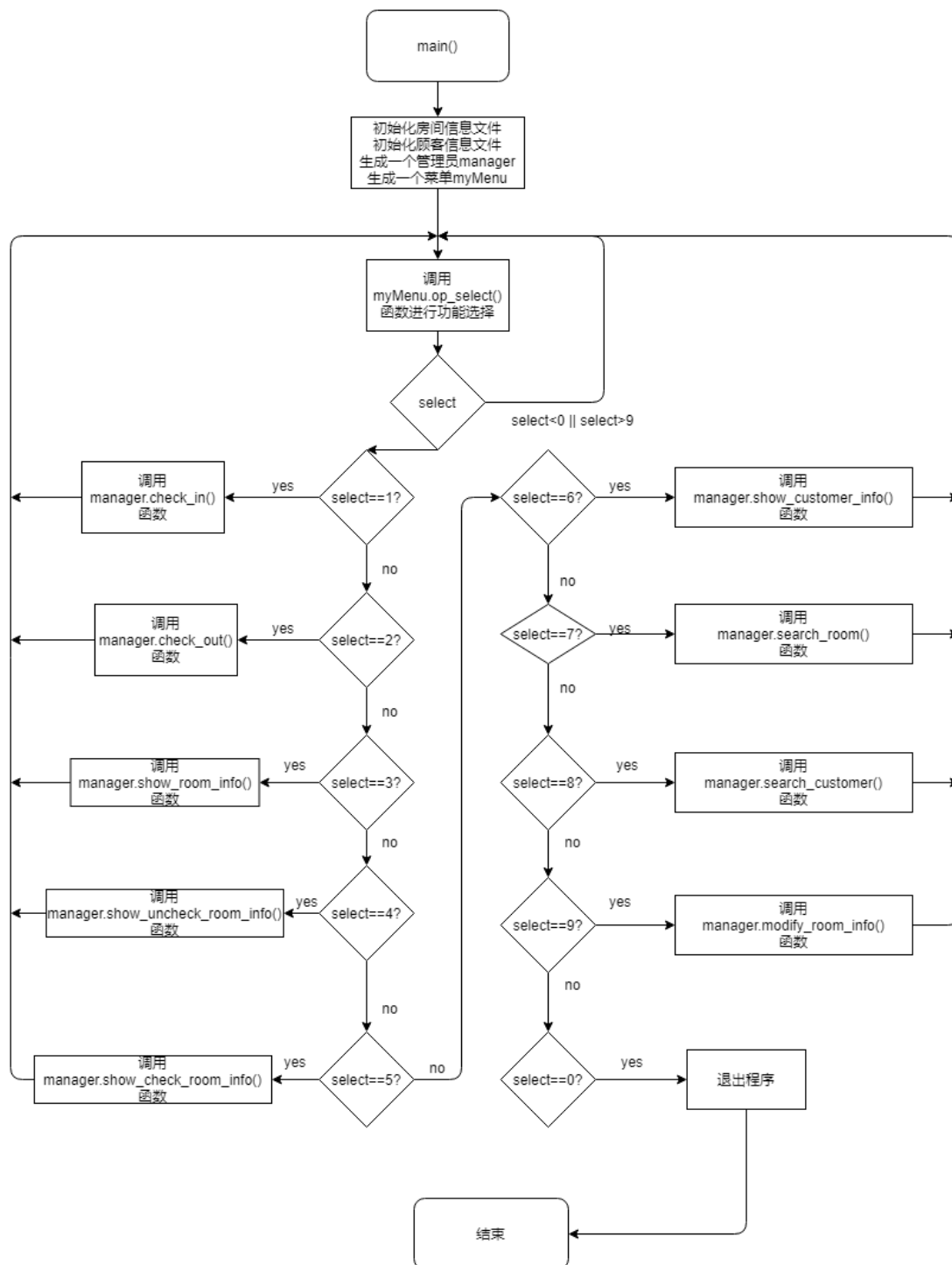
开发平台：vs2010

系统总框图

客房管理系统类图：

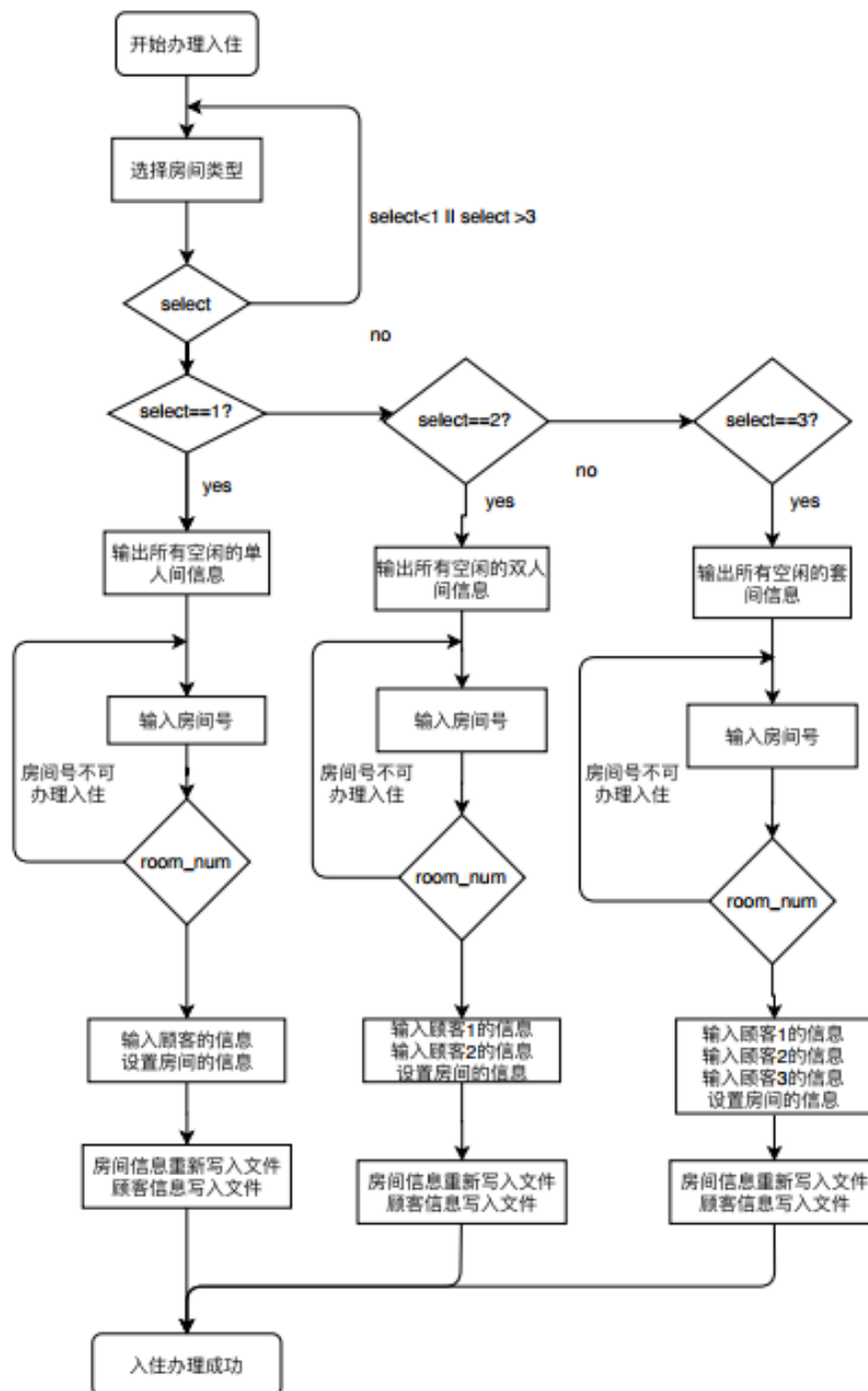


Main () 函数中函数的调用关系:

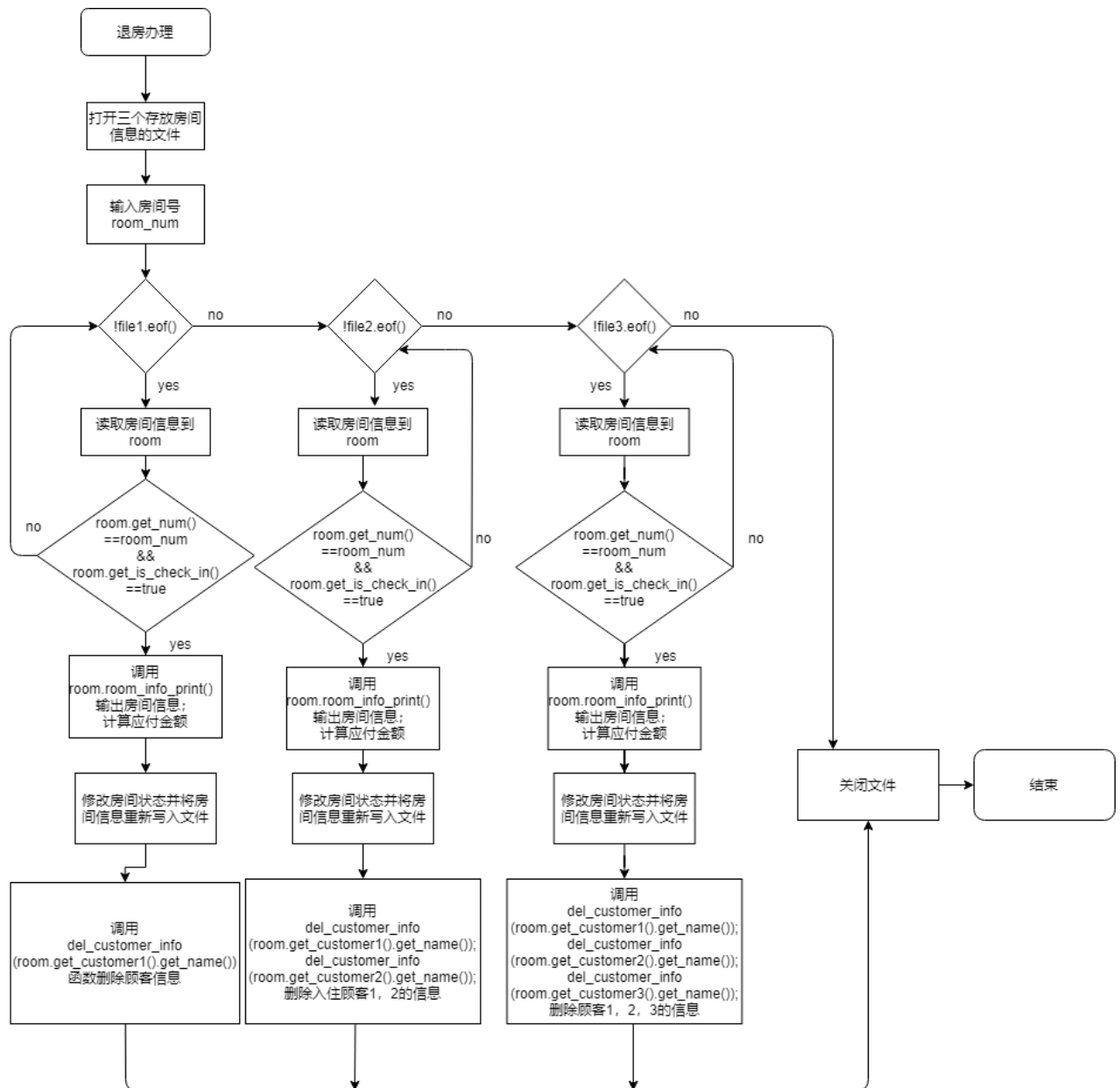


每个模块设计分析

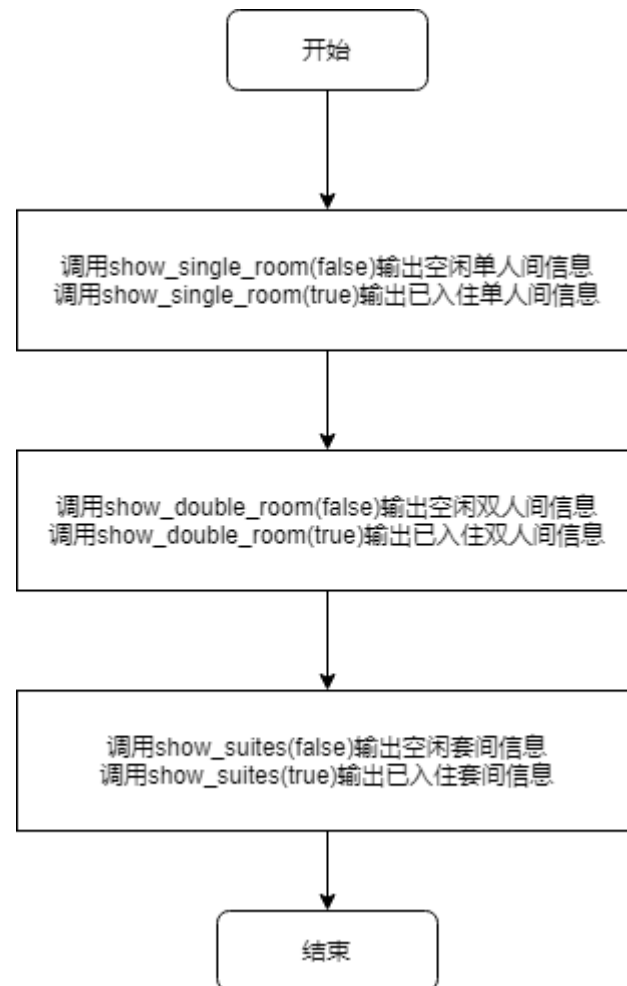
入住办理模块：



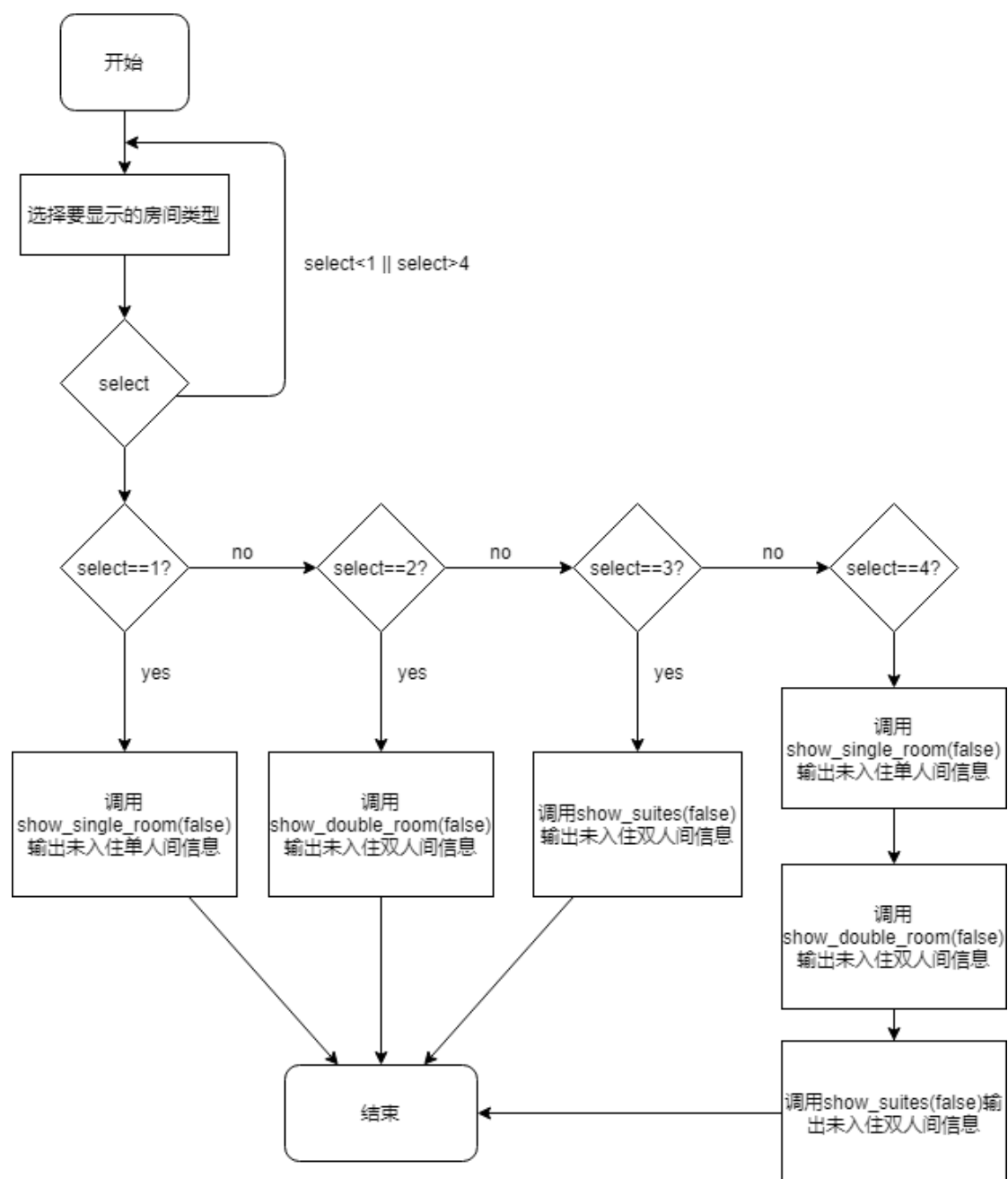
退房办理模块：



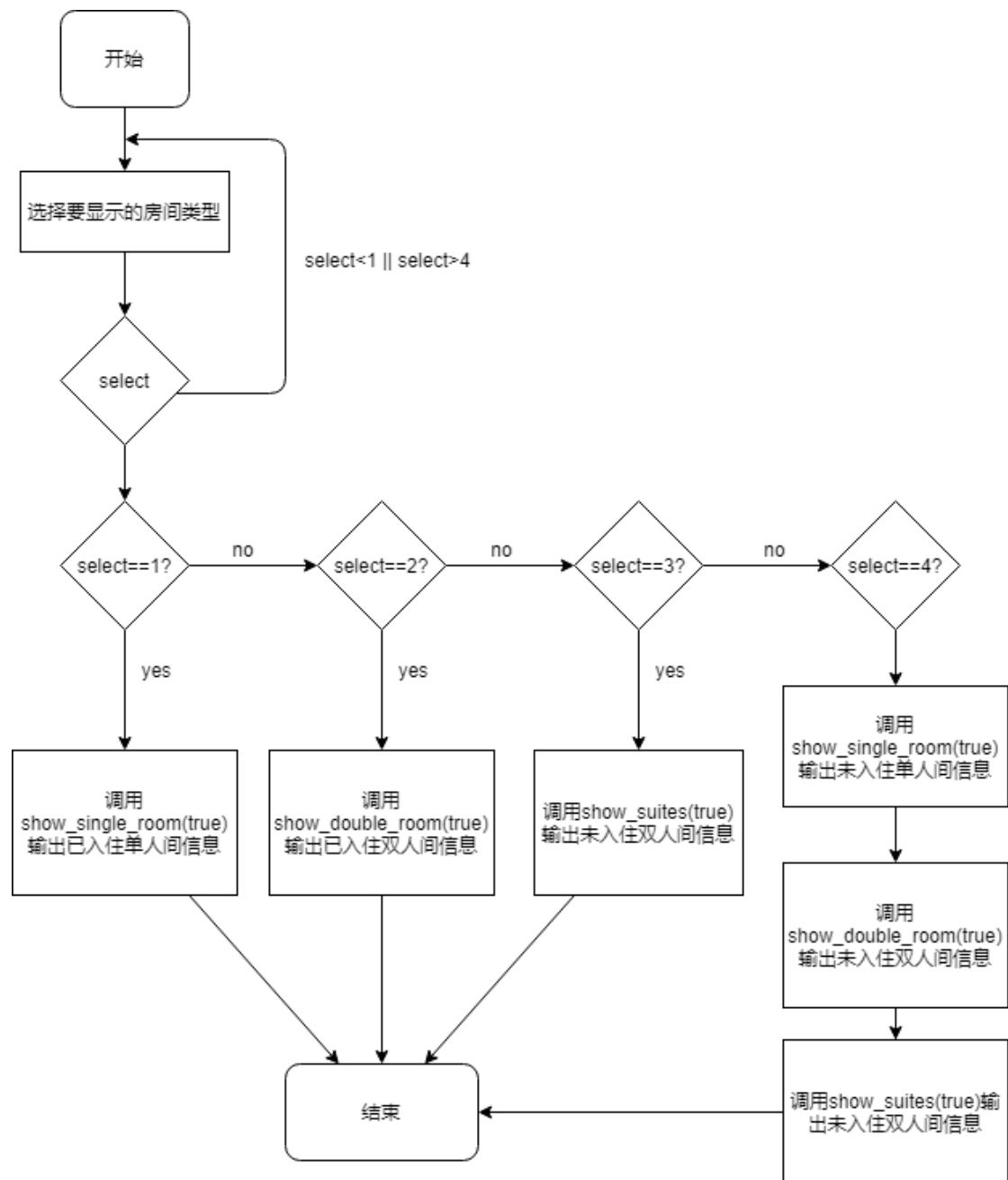
显示所有房间信息模块：



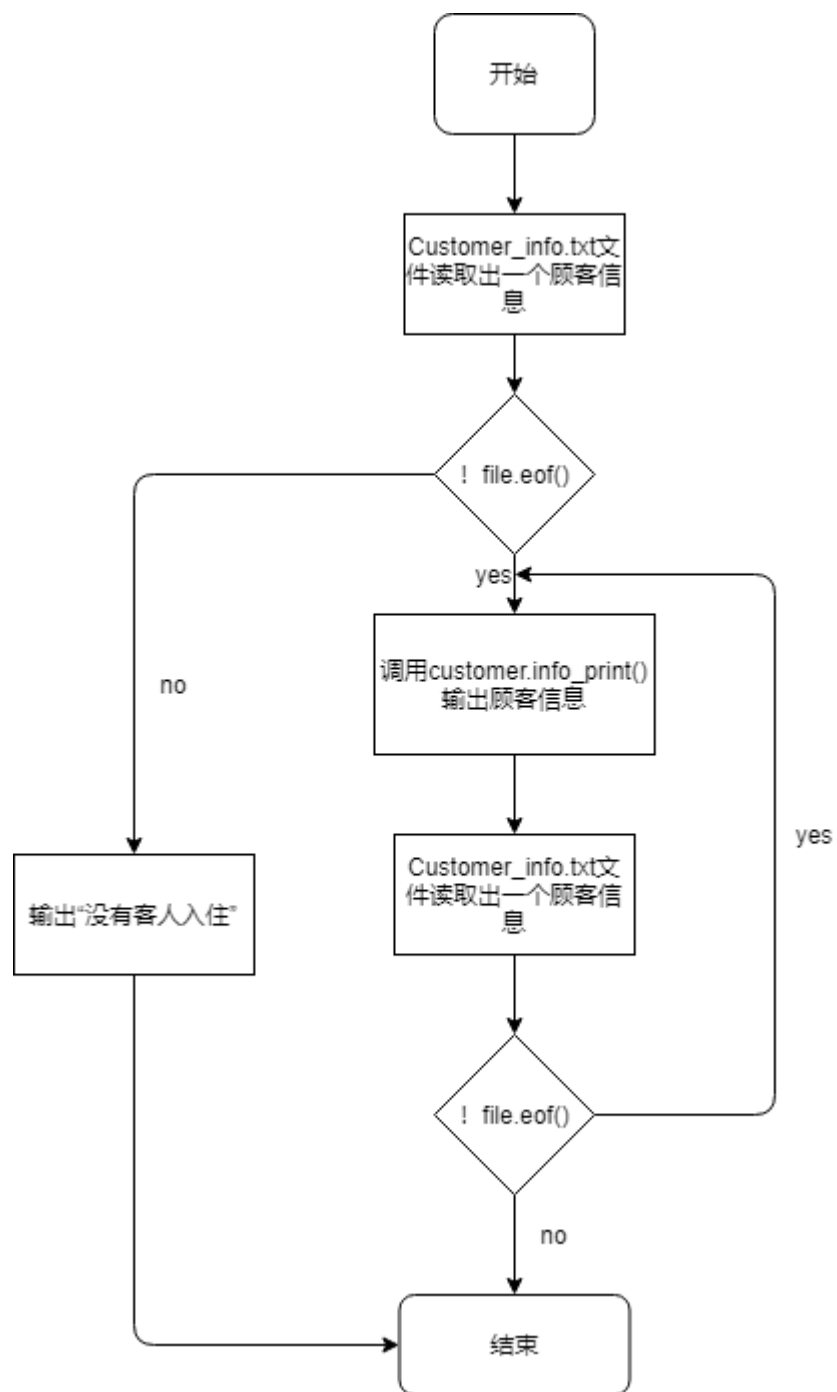
显示空闲房间信息模块：



显示已入住房间信息模块：

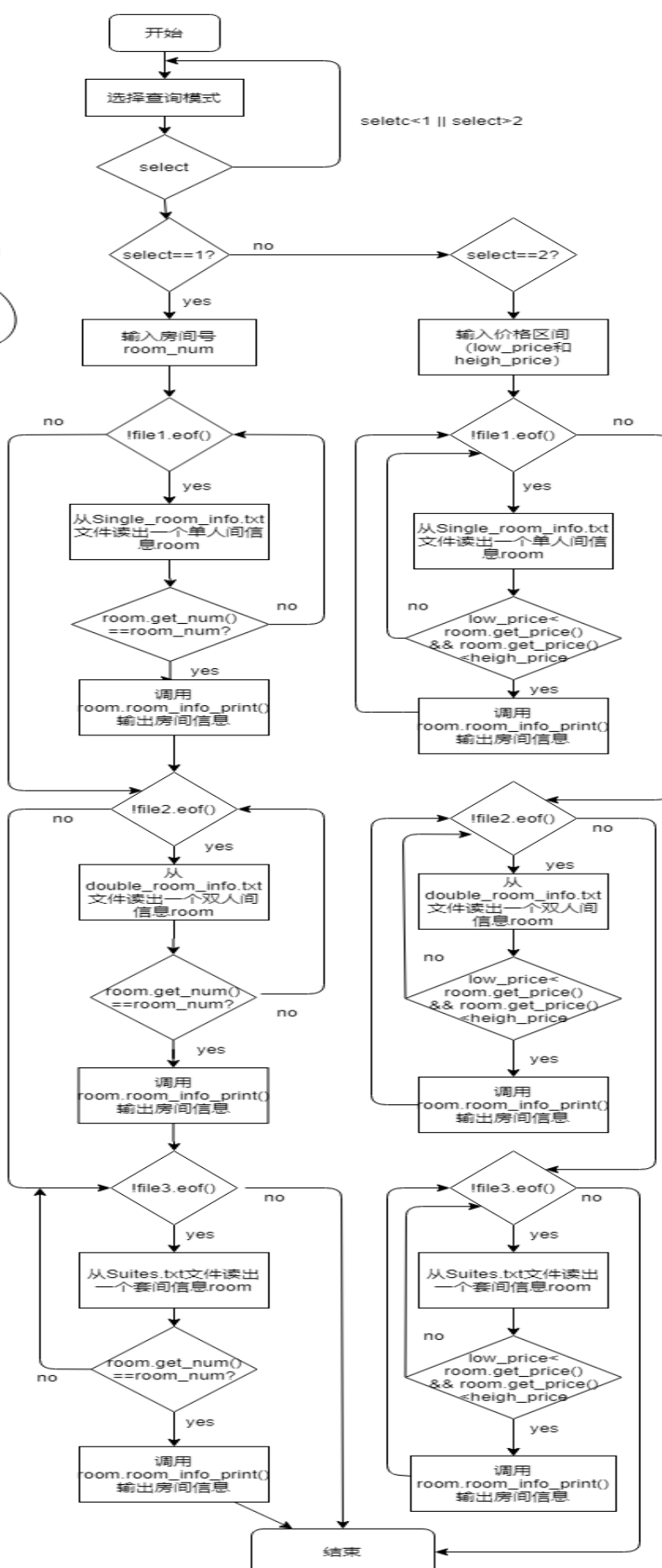


所有顾客信息输出模块：

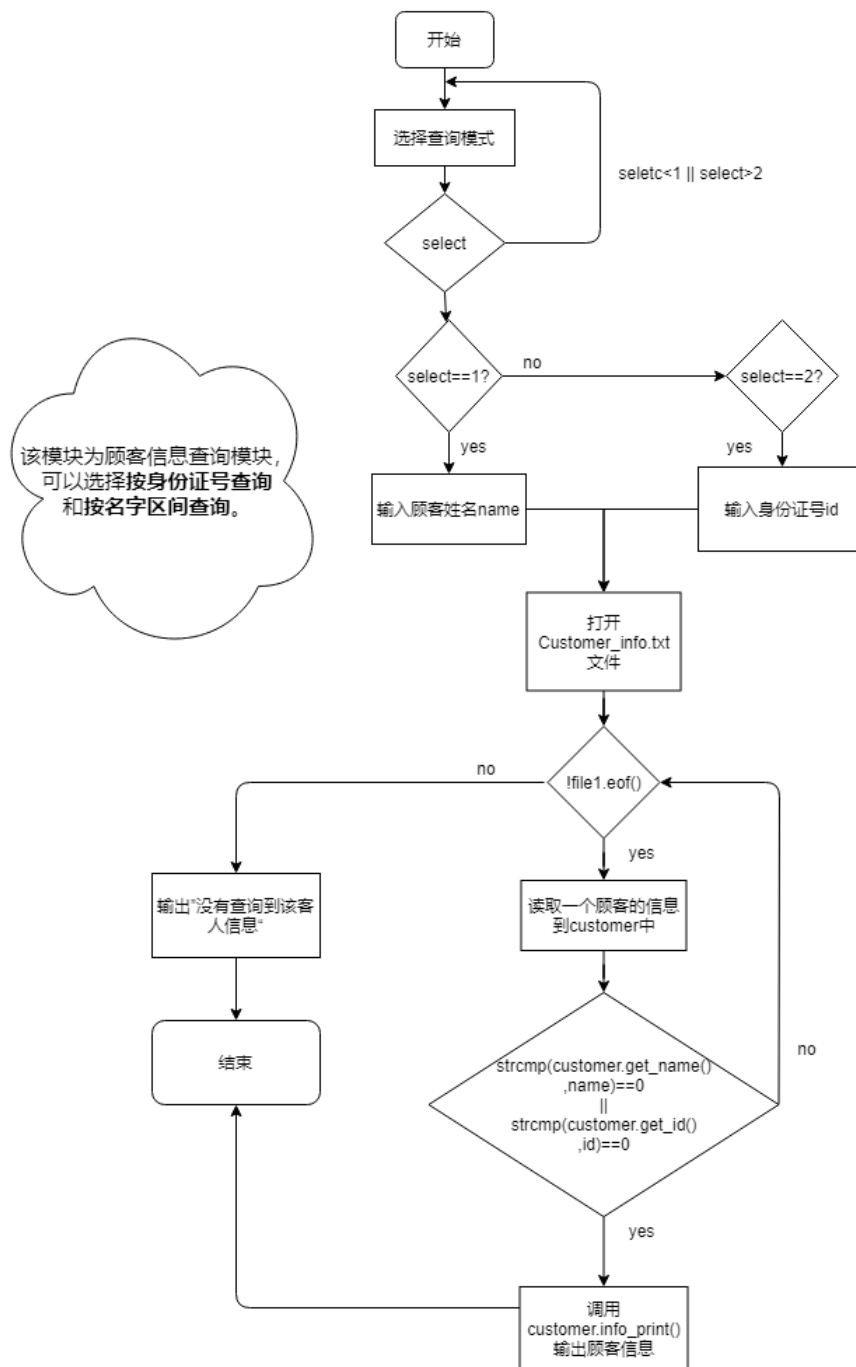


房间信息查询模块：

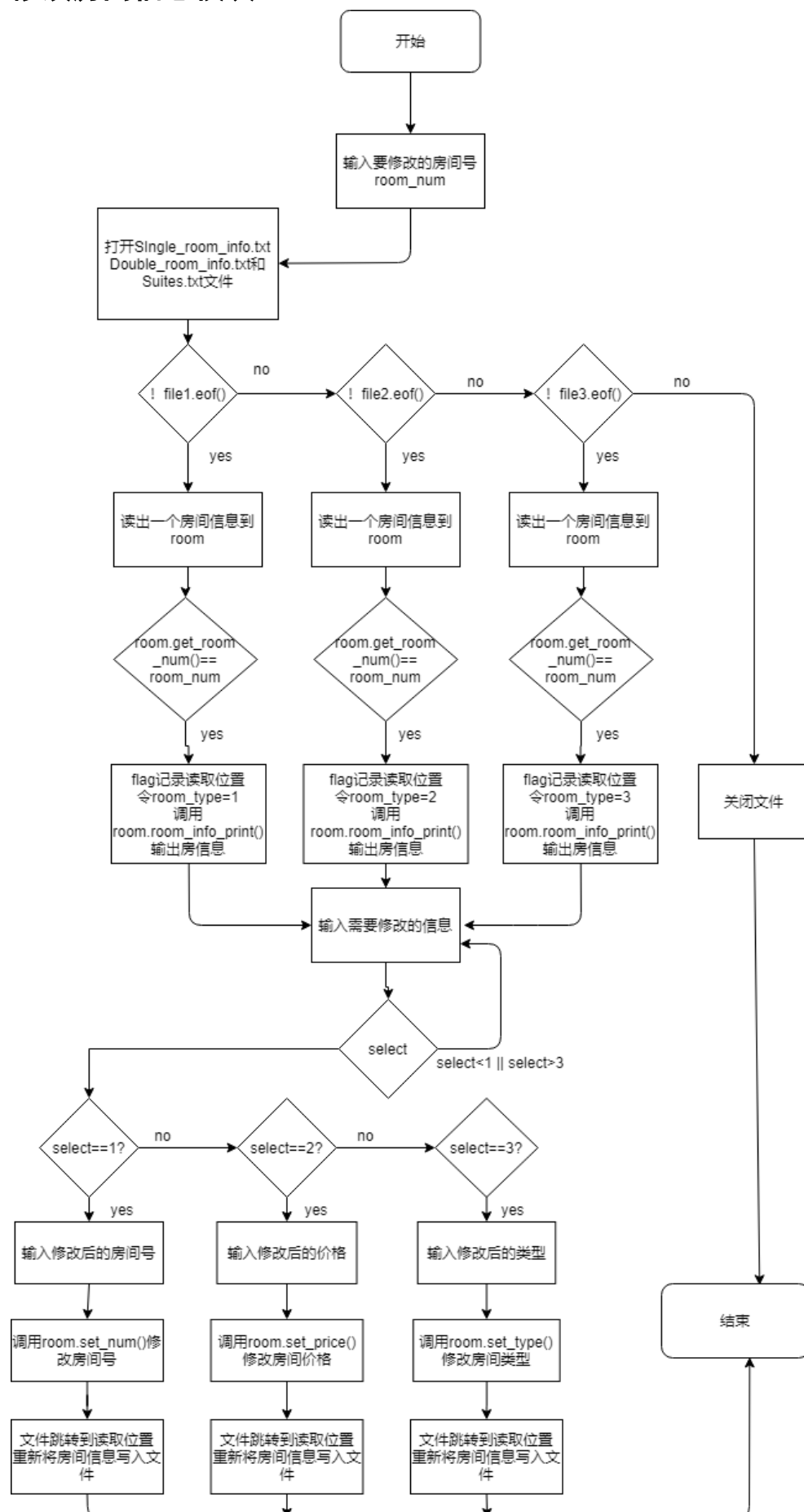
该模块为房间信息查询模块，
可以选择按房间号查询
和按价格区间查询。



顾客信息查询模块：



修改房间信息模块：



所有类的定义和函数说明

Menu 类:

```
class Menu{
public:
    int op_select();
};
```

说明：该类只有一个函数 `op_select()`，这个函数在 `main` 函数中被循环调用，用于输出操作的提示信息和清除屏幕上的内容。直到选择退出才退出该函数同时退出客房管理统。

Customer 类:

```
class Customer
{
private:
    char name[10];    //姓名
    char id[19];      //身份证号
    int room_num;     //入住房间编号
    int days;         //入住天数
    time_t date;      //入住日期

public:
    char* get_name();
    char* get_id();
    int get_room_num();
    int get_days();
    time_t get_date();

    void set_name(char* name);
    void set_id(char* id);
    void set_room_num(int room_num);
    void set_days(int days);
    void set_date(time_t &date);

    void info_print();
    void info_set(char *name,char *id,int room_num,int
days,time_t& date);
};
```

说明：该类表示顾客，用于存放办理入住时需要登记的信息和顾客入住后的房间号(room_num), 入住天数(days), 和办理入住时的日期(date)。char* get_name() 函数用于获取顾客的姓名；char* get_id() 函数用于获取顾客的身份证号；int get_room_num() 用于获取顾客的房间号；int get_days() 函数用于获取顾客办理入住的天数；time_t get_date() 用于获取顾客入住的日期。void set_name(char* name)。后面对应的set函数用于对各属性的设置。void info_print() 函数用于打印顾客的信息；void info_set(char *name, char *id, int room_num, int days, time_t& date) 函数用于一次性设置顾客对象的所有属性。

Room 类：

```
class Room{    //父类
protected:
    int num;           //房间编号
    int price;         //价格
    bool is_check_in;  //是否已有人入住
    int type;          //房间类型（经济间，标准间，豪华间）
    int days;          //入住天数
    time_t date;       //入住日期
public:
    int get_num();
    int get_price();
    bool get_is_check_in();
    int get_type();
    int get_days();
    time_t& get_date();

    void set_num(int num);
    void set_price(int price);
    void set_is_check_in(bool check);
    void set_type(int type);
    void set_days(int days);
    void set_date(time_t& date);
    void room_info_set(int num, int price, bool is_check_in, int
type);
};
```

说明：该类作为所有房间的父亲类，所有属性都为 protected 类型，方便其子类的访问。int get_num() 函数用于获取房间编号；int get_price() 函数用于获取房价价格；bool get_is_check_in() 函数用于获取房间的状态，如果为true 表示已有人入住，反之表示房间为空闲状态；int get_type() 函数用于获取房

间的类型，返回0表示房间为经济间，1表示标准间，2表示豪华间；`int get_days()`函数用于获取房间中顾客入住的天数；`time_t& get_date()`函数用于获取顾客入住的时间。`void room_info_set(int num, int price, bool is_check_in, int type)`函数用于对num, price, is_check_in, type这几个基本信息的设置（这些信息在没有客人入住的时候就已经可以知道）。对应得后面以set开头的函数就是对房间的各个属性进行设置。

Single_room类:

```
class Single_room:public Room//单人间
{
private:
    Customer customer;
public:
    void set_cutomer(Customer& customer);
    Customer& get_customer();
    void room_info_print();
};
```

说明：该类表示单人间，是Room的子类，继承了Room类的所有属性和函数。同时子类比父类多了一个customer属性，customer用于存放入住的客人的信息。`Customer& get_customer()`函数用于返回一个Customer类型的对象，该对象存放了入住该房间的客人的信息。`void set_cutomer(Customer& customer)`函数用于设置该房间入住客人的信息。`void room_info_print()`函数用于将该单人间的信息打印出来，如果已有顾客入住，就将入住顾客的基本信息也一起打印出来。

Double_room类:

```
class Double_room:public Room
{
private:
    Customer customer1;    //入住顾客1
    Customer customer2;    //入住顾客2
public:
    void set_cutomer1(Customer& customer1);
    void set_cutomer2(Customer& customer2);
    Customer& get_customer1();
    Customer& get_customer2();
    void room_info_print();
};
```

说明：该类表示的是双人间，同样是Room类的子类。其实例化的对象存放的是双人间的信息，由于是双人间，所以比单人间多了一个Customer类，用于存放入住顾客2的信息。其它函数功能分别为customer1,customer2这两个对象的获取和设置。void room_info_print() 函数用于打印该房间的相关信息。

Suite类:

```
class Suite:public Room
{
private:
    Customer customer1;
    Customer customer2;
    Customer customer3;
public :
    void set_cutomer1(Customer& customer1);
    void set_cutomer2(Customer& customer2);
    void set_cutomer3(Customer& customer3);
    Customer& get_customer1();
    Customer& get_customer2();
    Customer& get_customer3();
    void room_info_print();
};
```

说明：该类表示套间，其实例化的对象用于存放套间的信息。由于套间入住的顾客较多，所以其Customer类型的对象有3个，用于存放入住的顾客的信息。函数分别为对customer1,customer2和customer3这三个对象的获取和设置。void room_info_print() 函数用于输出该套间的信息和入住的3个对象的信息

Manager类:

```
class Manager
{
public:
    void check_in();           //客人入住办理
    void check_out();          //客人退房办理
    void show_customer_info(); //展示所有顾客的信息
    void show_room_info();     //展示所有房间的信息
    void show_uncheck_room_info(); //展示空闲的房间的信息
    void show_check_room_info(); //展示所有已入住的房间的信息
    void search_room();         //输入房间号，查询房间号信息
    void search_customer();     //输入客人姓名，查询客人信息
    void modify_room_info();    //修改房间价格

    void del_customer_info(char* name);
    void show_single_room(bool check_in); //显示单人间房间的信息
    void show_double_room(bool check_in); //显示双人间房间的信息
    void show_suites(bool check_in);      //显示套间的房间信息
    void init_room_info();                //初始化(文件)数据库
};
```

说明：该类表示管理员。

check_in() 函数实现客人的入住办理；

check_out() 函数实现客人的退房办理；

show_customer_info() 实现所有入住的客人的信息的打印；show_room_info() 实现所有房间的信息的打印；show_uncheck_room_info() 函数实现所有空闲房间的信息打印；show_check_room_info() 函数实现所有已有顾客入住的房间的信息的打印；search_room() 函数用于输入房间号或价格区间查找符合的房间的信息；search_customer() 函数用于输入顾客的姓名或身份证号查找该顾客的信息；modify_room_info() 函数用于对空闲的房间的信息进行修改，可以修改的房间信息为房间号，价格，和房间的类型；

del_customer_info(char* name) 函数用于删除入住顾客的信息，在管理员给顾客进行退房办理时调用；

show_single_room(bool check_in) 函数根据check_in的值输出符合的单人间的信息，如果check_in==true则输出所有已有顾客入住的单人间的信息，如果check_in==false则输出空闲的单人间的房间的信息。

show_double_room(bool check_in) 和上述一致，根据check_in的值输出符合的双人间的信息。

show_suites(bool check_in) 用于输出相应的套间的信息。

init_room_info() 函数用于管理员进入系统时将单人间，双人间，套间的信息初始化并写入相应的文件。

特色函数算法

特色算法 1:

```
void Manager::del_customer_info(char *name)
{
    fstream file1("Customer_info.txt",ios::in|ios::out);
    if(!file1)
    {
        cout<<"客人信息数据库访问失败";
        exit(0);
    }
    fstream file2("temp.txt",ios::out);//文件不存在会自动生成
    if(!file2)
    {
        cout<<"临时文件生成失败！ ";
        exit(0);
    }

    file1.clear();
    file1.seekg(0,ios::beg);
    Customer customer;
    file1.read((char*)&customer,sizeof(customer));
    while(!file1.eof())
    {
        if(strcmp(customer.get_name(),name)!=0)
        {
            file2.write((char*)&customer,sizeof(customer));
        }
        file1.read((char*)&customer,sizeof(customer));
    }
    file1.close();
    file2.close();
    remove("Customer_info.txt");
    rename("temp.txt","Customer_info.txt");
}
```

函数说明：该函数实现的是将某个客人的信息从存放客人信息的文件中删除。一般的做法是将文件中的所有信息读入到链表或数组中，然后对链表或数组进行遍历，将不需要删除的信息以覆盖的方式重新写入到文件中去。这样做的缺点是需要花费 $O(n)$ 的空间复杂度来存放文件的数据。我的做法是使用 fstream

file2("temp.txt", ios::out) 语句来创建一个空的临时文件。然后对存放顾客信息的文件(Customer_info.txt)进行顺序读取, 如果不是要删除的信息就写入到临时文件中(temp.txt)。当 Customer_info.txt 文件中的数据读取完后, temp.txt 文件中保存的就是删除指定信息后的数据。最后使用 remove("Customer_info.txt") 将原来的 Customer_info.txt 文件删除, 再用 rename("temp.txt", "Customer_info.txt") 将临时文件 temp.txt 的名字修改为 Customer_info.txt。这样就实现了在 $O(1)$ 的空间复杂度内实现了对顾客信息的删除。

特色算法 2:

```
fstream file1("Single_room_info.txt", ios::in|ios::out);
if(!file1)
{
    cout<<"单人间数据库访问失败!";
    exit(0);
}
Single_room room;
do{
    cout<<"输入房间号: ";
    cin>>room_num;
    file1.clear(); //将文件的状态初始化
    file1.seekg(0L, ios::beg); //跳转到文件开头

    while(!file1.eof())
    {
        flag=file1.tellg(); //得到文件位置指针, 便于数据的修改
        file1.read((char*)&room, sizeof(room));
        if(room.get_num()==room_num && room.get_is_check_in()==false)
        //查找到该房间号
        {
            search=true;
            break;
        }
    }

    if(!search)
    {
        cout<<"房间号错误或该房间已有顾客入住!请重新输入房间号"<<endl;
```

```

    }
}while(!search); //对每次输入的房间号进行检查，看是否存在和空闲
cout<<"输入顾客姓名： ";
cin>>name;
cout<<"输入身份证号： ";
cin>>id;
cout<<"输入入住天数： ";
cin>>days;
date=time(0);
Customer customer;
customer.info_set(name,id,room_num,days,date);
room.set_days(days);
room.set_is_check_in(true);
room.set_cutomer(customer);
room.set_date(date);

fstream file2("Customer_info.txt",ios::in|ios::out|ios::app);
if(!file2)
{
    cout<<"客人信息数据库访问失败";
    exit(0);
}
file1.seekp(flag,ios::beg);
file1.write((char*)&room,sizeof(room)); //重新写入房间信息
file2.write((char*)&customer,sizeof(customer)); //写入客人信息

file1.close();
file2.close();
cout<<"入住办理成功！"<<endl;

```

函数说明：上述代码段是Manager::check_in()函数中的一部分，该函数实现客人的入住办理。客人入住办理时需要输入房间号，然后查询这个房间号是否存在或是否已经有人入住。如果存在且没有人入住就输入客人的信息。最后修改房间的信息重新写入到文件中，同时将客人的信息写入到文件中。

但是如果输入的房间号不对，就查询不到符合的信息。所以这里加了一个do{}while()循环来重复输入直到房间号可以办理入住。但是对于新输入的房间号进行查询时，发现即使输入的房间号是正确的，也会提示房间号不能办理入住。这是因为第一次查询结束后，文件的指针到了文件的末尾。所以总是查询不到输入的房间的信息，要想解决这个问题可以在每一次读完文件之后将文件关闭，第二次要读的时候再重新打开（重新打开文件时，文件指针在文件开始出）。但是这样重复的关闭和打开文件，程序的开销就会增大，为了解决这个问题。我在每次对文件读写之前使用了file1.clear()函数，这个函数清除了文件中指针的状态，然后再用file1.seekg(0L, ios::beg)函数将文件位置指针跳转到文件的开始处。这样就避免了每次从文件头开始访问之前都要重新打开文

件。

同时在读取到符合的房间信息的时候用flag=file1.tellg()语句将读取到这个房间信息的位置记录了下来，对房间的信息进行修改过后，再使用file1.seekp(flag,ios::beg)语句将写入指针跳转到刚才的位置，最后在这个位置重新写入房间的信息file1.write((char*)&room,sizeof(room))。这样写入会用新的房间信息覆盖原来的房间信息，就实现了房间信息的修改。这样做就不用将整个文件的数据读到链表进行修改后再写入，减少了时间和空间的开销。

存在的不足与对策

| 不足 | 对策 |
|--|---------------------------|
| 不够美观，依然停留在控制台界面 | 自学图形界面 |
| modify_room_info()函数只实现了对房间的一些基本信息的修改。有些修改需要同时修改存放房间数据的文件和存放顾客数据的文件，这部分内容较为复杂没有实现。 | 继续编程补充相应的功能 |
| 输入数据时不能判断数据是否合法，可能导致程序出错。比如输入身份证号时输入的位数大于了19位程序就会出错。 | 继续对代码进行优化，当输入数据不合法时给出提示信息 |

使用说明

使用 vs2010 打开项目“客房管理系统 2.0” 点击运行即可运行。程序运行后，会输出如图所示的功能选项。选择对应的数字后就可以进行操作。房间信息在程序启动时已经写入到文件中，不需要手动输入。

```
#####
#                                     #
#                               欢迎登录客房管理系统                               #
#                                     #
#####
#                                     #
#       1.      客人入住办理                                     #
#       2.      客人退房办理                                     #
#       3.      显示所有房间信息                               #
#       4.      显示空闲房间信息                               #
#       5.      显示已入住房间信息                             #
#       6.      显示所有客人信息                               #
#       7.      查询房间信息                                   #
#       8.      查询客人信息                                   #
#       9.      修改房间信息                                   #
#       0.      退出客房管理系统                               #
#####
给出你的操作:
```

客人入住办理功能演示：

```
给出你的操作:1
选择房间类型  1: 单人间   2: 双人间   3: 套间
选择: 1

-----当前空闲的单人间如下-----
房间号: 101      价格: 150/天      类型: 经济间      状态: 空闲
房间号: 102      价格: 150/天      类型: 经济间      状态: 空闲
房间号: 103      价格: 150/天      类型: 经济间      状态: 空闲
房间号: 201      价格: 220/天      类型: 标准间      状态: 空闲
房间号: 202      价格: 220/天      类型: 标准间      状态: 空闲
房间号: 203      价格: 220/天      类型: 标准间      状态: 空闲
房间号: 301      价格: 300/天      类型: 豪华间      状态: 空闲
房间号: 302      价格: 300/天      类型: 豪华间      状态: 空闲
房间号: 303      价格: 300/天      类型: 豪华间      状态: 空闲

输入房间号: 101
输入顾客姓名: 张先生
输入身份证号: 520221199708020999
输入入住天数: 2
入住办理成功!

给出你的操作:1
选择房间类型  1: 单人间   2: 双人间   3: 套间
选择: 2

-----当前空闲的双人间如下-----
房间号: 401      价格: 200/天      类型: 经济间      状态: 空闲
房间号: 402      价格: 200/天      类型: 经济间      状态: 空闲
房间号: 403      价格: 200/天      类型: 经济间      状态: 空闲
房间号: 501      价格: 250/天      类型: 标准间      状态: 空闲
房间号: 502      价格: 250/天      类型: 标准间      状态: 空闲
房间号: 503      价格: 250/天      类型: 标准间      状态: 空闲
房间号: 601      价格: 350/天      类型: 豪华间      状态: 空闲
房间号: 602      价格: 350/天      类型: 豪华间      状态: 空闲
房间号: 603      价格: 350/天      类型: 豪华间      状态: 空闲

输入房间号: 401
输入顾客1姓名: 李先生
输入顾客1身份证号: 520221198703080999
输入入住天数: 1
输入顾客2姓名: 赵先生
输入顾客2身份证号: 520221198704060111
入住办理成功!
```


显示所有房间信息功能演示:

给出你的操作:3

| -----单人间----- | | | |
|---|-----------|---------|---------|
| 房间号: 102 | 价格: 150/天 | 类型: 经济间 | 状态: 空闲 |
| 房间号: 103 | 价格: 150/天 | 类型: 经济间 | 状态: 空闲 |
| 房间号: 201 | 价格: 220/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 202 | 价格: 220/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 203 | 价格: 220/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 301 | 价格: 300/天 | 类型: 豪华间 | 状态: 空闲 |
| 房间号: 302 | 价格: 300/天 | 类型: 豪华间 | 状态: 空闲 |
| 房间号: 303 | 价格: 300/天 | 类型: 豪华间 | 状态: 空闲 |
| 房间号: 101 | 价格: 150/天 | 类型: 经济间 | 状态: 已入住 |
| 入住顾客: 张先生 入住天数: 2 入住日期: Wed May 20 21:38:40 2020 | | | |
| -----双人间----- | | | |
| 房间号: 402 | 价格: 200/天 | 类型: 经济间 | 状态: 空闲 |
| 房间号: 403 | 价格: 200/天 | 类型: 经济间 | 状态: 空闲 |
| 房间号: 501 | 价格: 250/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 502 | 价格: 250/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 503 | 价格: 250/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 601 | 价格: 350/天 | 类型: 豪华间 | 状态: 空闲 |
| 房间号: 602 | 价格: 350/天 | 类型: 豪华间 | 状态: 空闲 |
| 房间号: 603 | 价格: 350/天 | 类型: 豪华间 | 状态: 空闲 |
| 房间号: 401 | 价格: 200/天 | 类型: 经济间 | 状态: 已入住 |
| 入住顾客1: 李先生 入住顾客2: 赵先生 入住天数: 1 入住日期: Wed May 20 21:38:59 2020 | | | |
| -----套间----- | | | |
| 房间号: 701 | 价格: 300/天 | 类型: 经济间 | 状态: 空闲 |

显示所有空闲房间功能演示:

给出你的操作:4

| -----空闲房间查询----- | | | |
|------------------|-----------|---------|--------------------|
| 选择要查询的房间类型! | 1: 单人间 | 2: 双人间 | 3: 套间 4: 所有类型房间 |
| 选择: 1 | | | |
| 房间号: 102 | 价格: 150/天 | 类型: 经济间 | 状态: 空闲 |
| 房间号: 103 | 价格: 150/天 | 类型: 经济间 | 状态: 空闲 |
| 房间号: 201 | 价格: 220/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 202 | 价格: 220/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 203 | 价格: 220/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 301 | 价格: 300/天 | 类型: 豪华间 | 状态: 空闲 |
| 房间号: 302 | 价格: 300/天 | 类型: 豪华间 | 状态: 空闲 |
| 房间号: 303 | 价格: 300/天 | 类型: 豪华间 | 状态: 空闲 |
| 请按任意键继续. . . | | | |

给出你的操作:4

| -----空闲房间查询----- | | | |
|------------------|-----------|---------|--------------------|
| 选择要查询的房间类型! | 1: 单人间 | 2: 双人间 | 3: 套间 4: 所有类型房间 |
| 选择: 2 | | | |
| 房间号: 402 | 价格: 200/天 | 类型: 经济间 | 状态: 空闲 |
| 房间号: 403 | 价格: 200/天 | 类型: 经济间 | 状态: 空闲 |
| 房间号: 501 | 价格: 250/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 502 | 价格: 250/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 503 | 价格: 250/天 | 类型: 标准间 | 状态: 空闲 |
| 房间号: 601 | 价格: 350/天 | 类型: 豪华间 | 状态: 空闲 |
| 房间号: 602 | 价格: 350/天 | 类型: 豪华间 | 状态: 空闲 |
| 房间号: 603 | 价格: 350/天 | 类型: 豪华间 | 状态: 空闲 |
| 请按任意键继续. . . | | | |

显示已入住房间信息功能演示：

```
给出你的操作:5
-----已入住房间查询-----
选择要查询的剩余的房間类型! 1: 單人間 2: 雙人間 3: 套間 4: 所有类型房間
选择: 1
-----單人間-----
房間号: 101 价格: 150/天 类型: 經濟間 状态: 已入住
入住顧客: 張先生 入住天数: 2 入住日期: Wed May 20 21:38:40 2020
```

```
给出你的操作:5
-----已入住房間查詢-----
选择要查詢的剩余的房間类型! 1: 單人間 2: 雙人間 3: 套間 4: 所有类型房間
选择: 2
-----雙人間-----
房間号: 401 价格: 200/天 类型: 經濟間 状态: 已入住
入住顧客1: 李先生 入住顧客2: 趙先生 入住天数: 1 入住日期: Wed May 20 21:38:59 2020
```

显示所有顧客信息功能演示：

```
给出你的操作:6
-----所有入住客人信息如下-----
顧客姓名: 張先生
身份证号: 520221199708020999
房間号 : 101
入住日期: Wed May 20 21:38:40 2020
入住天数: 2

顧客姓名: 李先生
身份证号: 520221198703080999
房間号 : 401
入住日期: Wed May 20 21:38:59 2020
入住天数: 1

顧客姓名: 趙先生
身份证号: 520221198704060111
房間号 : 401
入住日期: Wed May 20 21:38:59 2020
入住天数: 1
-----
```

查詢房間信息功能演示：

```
给出你的操作:7
选择查詢模式 1: 按房間号查詢 2: 按价格区间查詢
选择: 1
輸入房間号: 102
房間号: 102 价格: 150/天 类型: 經濟間 状态: 空闲
请按任意键继续
```

```

给出你的操作:7
选择查询模式 1: 按房间号查询 2: 按价格区间查询
选择: 2
输入价格区间:
最低价: 100
最高价: 200
房间号: 101 价格: 150/天 类型: 经济间 状态: 已入住
入住顾客: 张先生 入住天数: 2 入住日期: Wed May 20 21:38:40 2020

房间号: 102 价格: 150/天 类型: 经济间 状态: 空闲
房间号: 103 价格: 150/天 类型: 经济间 状态: 空闲

```

查询客人信息功能演示:

```

给出你的操作:8
选择查询模式 1: 按姓名查询 2: 按身份证号查询
选择: 1
输入客人姓名: 张先生
顾客姓名: 张先生
身份证号: 520221199708020999
房间号 : 101
入住日期: Wed May 20 21:38:40 2020
入住天数: 2

```

```

给出你的操作:8
选择查询模式 1: 按姓名查询 2: 按身份证号查询
选择: 2
输入身份证号: 520221199708020999
顾客姓名: 张先生
身份证号: 520221199708020999
房间号 : 101
入住日期: Wed May 20 21:38:40 2020
入住天数: 2

```

客人退房办理功能演示:

```

给出你的操作:2
输入房间号: 101
-----房间信息-----
房间号: 101 价格: 150/天 类型: 经济间 状态: 已入住
入住顾客: 张先生 入住天数: 2 入住日期: Wed May 20 21:38:40 2020

需付费用: 300元
退房办理成功!

```

给出你的操作:6

-----所有入住客人信息如下-----

顾客姓名: 李先生
身份证号: 520221198703080999
房间号 : 401
入住日期: Wed May 20 21:38:59 2020
入住天数: 1

顾客姓名: 赵先生
身份证号: 520221198704060111
房间号 : 401
入住日期: Wed May 20 21:38:59 2020
入住天数: 1

房间信息修改功能演示:

给出你的操作:9

输入房间号对其信息进行修改: 101

-----该房间信息-----

房间号: 101 价格: 150/天 类型: 经济间 状态: 空闲

选择要修改的房间信息. 1: 房间号 2: 房间价格 3: 房间类型

选择: 2

输入修改后的房间价格: 198

给出你的操作:3

-----单人间-----

房间号: 101 价格: 198/天 类型: 经济间 状态: 空闲

程序源代码

Main.c

```
#include "Menu.h"
#include "Manager.h"
#include "Customer.h"
#include <fstream>

int main()
{
    Menu myMenu;
    Manager manager;

    manager.init_room_info();
    while(1)
    {
        switch(myMenu.op_select())
        {
            case 1:
                manager.check_in();//客人入住办理
                system("pause");
                break;
            case 2:
                manager.check_out();//客人退房办理
                system("pause");
                break;
            case 3:
                manager.show_room_info();//显示所有房间信息
                system("pause");
                break;
            case 4:
                manager.show_uncheck_room_info();//显示未入住房间信息
                system("pause");
                break;
            case 5:
                manager.show_check_room_info();//显示已入住房间信息
                system("pause");
                break;
            case 6:
                manager.show_customer_info();//显示所有客人信息
                system("pause");
                break;
```

```

        case 7:
            manager.search_room();//查询房间信息
            system("pause");
            break;
        case 8:
            manager.search_customer();//查询客人信息
            system("pause");
            break;
        case 9:
            manager.modify_room_info();//修改房间信息
            system("pause");
            break;
        case 0:
            exit(0);
    }
}
return 0;
}

```

Menu. h

```

#ifndef MENU_H
#define MENU_H
#include <iostream>
using namespace std;
class Menu{
public:
    int op_select();
};
#endif

```

Menu. cpp

```

#include "Menu.h"
int Menu::op_select()
{
    int select;
    do
    {
        system("cls");
        cout<<"
#####\n";
        cout<<" #                               #\n";
        cout<<" #                               欢迎登录客房管理系统 #\n";
    }
}

```

```

        cout<<" #                                     #\n";
        cout<<"
#####\n";
        cout<<" #          1.   客人入住办理          #\n";
        cout<<" #          2.   客人退房办理          #\n";
        cout<<" #          3.   显示所有房间信息        #\n";
        cout<<" #          4.   显示空闲房间信息        #\n";
        cout<<" #.        5.   显示已入住房间信息        #\n";
        cout<<" #.        6.   显示所有客人信息        #\n";
        cout<<" #.        7.   查询房间信息            #\n";
        cout<<" #.        8.   查询客人信息            #\n";
        cout<<" #          9.   修改房间信息            #\n";
        cout<<" #          0.   退出客房管理系统          #\n";
        cout<<"
#####\n";
        cout<<"给出你的操作:";
        cin>>select;
    }while(select<0 || select>9);
    return select;//返回选择的功能
}

```

Customer.h

```

#ifndef CUSTOMER_H
#define CUSTOMER_H
#include <ctime>
class Customer
{
private:
    char name[10];
    char id[19];    //身份证号
    int room_num;   //入住房间编号
    int days;       //入住天数
    time_t date;    //入住日期

public:
    char* get_name();
    char* get_id();
    int get_room_num();
    int get_days();
    time_t get_date();

    void set_name(char* name);
    void set_id(char* id);

```

```

    void set_room_num(int room_num);
    void set_days(int days);
    void set_date(time_t &date);

    void info_print();
    void info_set(char *name, char *id, int room_num, int days, time_t&
date);
};
#endif

```

Customer.cpp

```

#include "Customer.h"
#include <iostream>
using namespace std;

char* Customer::get_name()
{
    return this->name;
}
char* Customer::get_id()
{
    return this->id;
}
int Customer::get_room_num()
{
    return this->room_num;
}
int Customer::get_days()
{
    return this->days;
}
time_t Customer::get_date()
{
    return this->date;
}
void Customer::set_name(char *name)
{
    strcpy(this->name, name);
}
void Customer::set_id(char *id)
{
    strcpy(this->id, id);
}

```



```

void Customer::set_days(int days)
{
    this->days=days;
}
void Customer::set_room_num(int room_num)
{
    this->room_num=room_num;
}
void Customer::set_date(time_t &date)
{
    this->date=date;
}
void Customer::info_set(char *name,char *id,int room_num,int
days,time_t& date)
{
    strcpy(this->name,name);
    strcpy(this->id,id);
    this->room_num=room_num;
    this->days=days;
    this->date=date;
}
void Customer::info_print()
{
    cout<<"顾客姓名: "<<this->name<<endl;
    cout<<"身份证号: "<<this->id<<endl;
    cout<<"房间号  : "<<this->room_num<<endl;
    cout<<"入住日期: "<<ctime(&(this->date));
    cout<<"入住天数: "<<this->days<<endl;
}

```

Room. h

```

#ifndef ROOM_H
#define ROOM_H
#include <iostream>
using namespace std;
#include "Customer.h"
enum{ECONOMY_ROOM,STANDARD_ROOM,DELUXE_ROOM}; //房间的类型
#define UNCHECK false
#define CHECK true
class Room{    //父类
protected:
    int num;    //房间编号

```

```

    int price;           //价格
    bool is_check_in;    //是否已有人入住
    int type;            //房间类型（经济间，标准间，豪华间）
    int days;            //入住天数
    time_t date;         //入住日期
public:
    int get_num();
    int get_price();
    bool get_is_check_in();
    int get_type();
    int get_days();
    time_t& get_date();

    void set_num(int num);
    void set_price(int price);
    void set_is_check_in(bool check);
    void set_type(int type);
    void set_days(int days);
    void set_date(time_t& date);

    void room_info_set(int num,int price,bool is_check_in,int type);
};
#endif

```

Room. cpp

```

#include "Room.h"
#include <fstream>

int Room::get_num()
{
    return this->num;
}
int Room::get_price()
{
    return this->price;
}
bool Room::get_is_check_in()
{
    return this->is_check_in;
}
int Room::get_days()
{

```

```

        return this->days;
    }
    void Room::set_num(int num)
    {
        this->num=num;
    }
    void Room::set_price(int price)
    {
        this->price=price;
    }
    void Room::set_type(int type)
    {
        this->type=type;
    }
    void Room::set_days(int days)
    {
        this->days=days;
    }
    void Room::set_is_check_in(bool check)
    {
        this->is_check_in=check;
    }
    void Room::set_date(time_t& date)
    {
        this->date=date;
    }
    time_t& Room::get_date()
    {
        return this->date;
    }
    void Room::room_info_set(int num,int price,bool is_check_in,int type)
    {
        this->num=num;
        this->price=price;
        this->is_check_in=is_check_in;
        this->type=type;
    }
}

```

Single_room.h

```

#ifndef SINGLE_ROOM_H
#define SINGLE_ROOM_H
#include "Room.h"
class Single_room:public Room//派生类

```

```

{
private:
    Customer customer;
public:
    void set_cutomer(Customer& customer);
    Customer& get_customer();
    //void room_info_set(int num,int price,bool is_check_in,int type);
    void room_info_print();//
};
#endif

```

Single_room.cpp

```

#include "Single_room.h"
void Single_room::set_cutomer(Customer& customer)
{
    this->customer=customer;
}
Customer& Single_room::get_customer()
{
    return this->customer;
}
void Single_room::room_info_print()
{
    cout<<"房间号: "<<this->num<<"\t";
    cout<<"价格: "<<this->price<<"/天"<<"\t";
    switch(this->type)
    {
    case 0:
        cout<<"类型: 经济间"<<"\t";
        break;
    case 1:
        cout<<"类型: 标准间"<<"\t";
        break;
    case 2:
        cout<<"类型: 豪华间"<<"\t";
        break;
    }
    if(is_check_in)
    {
        cout<<"状态: 已入住"<<endl;
        cout<<"入住顾客: "<<this->customer.get_name()<<"\t";
        cout<<"入住天数: "<<this->days<<"\t";
        cout<<"入住日期: "<<ctime(&(this->date));
    }
}

```

```

    }
    else
    {
        cout<<"状态: 空闲"<<"\t";
    }
    cout<<endl;
}

```

Double_room.h

```

#include "Room.h"
class Double_room:public Room
{
private:
    Customer customer1;    //入住顾客1
    Customer customer2;    //入住顾客2
public:
    void set_cutomer1(Customer& customer1);
    void set_cutomer2(Customer& customer2);
    Customer& get_customer1();
    Customer& get_customer2();
    //void room_info_set(int num,int price,bool is_check_in,int type);
    void room_info_print();
};

```

Double_room.cpp

```

#include "Double_room.h"

void Double_room::set_cutomer1(Customer& customer1)
{
    this->customer1=customer1;
}
void Double_room::set_cutomer2(Customer& customer2)
{
    this->customer2=customer2;
}
Customer& Double_room::get_customer1()
{
    return this->customer1;
}

```

```

Customer& Double_room::get_customer2()
{
    return this->customer2;
}
void Double_room::room_info_print()
{
    cout<<"房间号: "<<this->num<<"\t";
    cout<<"价格: "<<this->price<<"/天"<<"\t";
    switch(this->type)
    {
        case 0:
            cout<<"类型: 经济间"<<"\t";
            break;
        case 1:
            cout<<"类型: 标准间"<<"\t";
            break;
        case 2:
            cout<<"类型: 豪华间"<<"\t";
            break;
    }
    if(is_check_in)
    {
        cout<<"状态: 已入住"<<endl;
        cout<<"入住顾客1: "<<this->customer1.get_name()<<"\t";
        cout<<"入住顾客2: "<<this->customer2.get_name()<<"\t";
        cout<<"入住天数: "<<this->days<<"\t";
        cout<<"入住日期: "<<ctime(&(this->date));
    }
    else
    {
        cout<<"状态: 空闲"<<"\t";
    }
    cout<<endl;
}
}

```

Suite.h

```

#include "Room.h"
#include "Customer.h"
class Suite:public Room
{
private:
    Customer customer1;
    Customer customer2;
}

```

```

    Customer customer3;
public :
    void set_cutomer1(Customer& customer1);
    void set_cutomer2(Customer& customer2);
    void set_cutomer3(Customer& customer3);
    Customer& get_customer1();
    Customer& get_customer2();
    Customer& get_customer3();
    //void room_info_set(int num,int price,bool is_check_in,int type);
    void room_info_print();
};

```

Suite. cpp

```

#include "Suite.h"

void Suite::set_cutomer1(Customer& customer1)
{
    this->customer1=customer1;
}
void Suite::set_cutomer2(Customer& customer2)
{
    this->customer2=customer2;
}
void Suite::set_cutomer3(Customer& customer3)
{
    this->customer2=customer3;
}
Customer& Suite::get_customer1()
{
    return this->customer1;
}
Customer& Suite::get_customer2()
{
    return this->customer2;
}
Customer& Suite::get_customer3()
{
    return this->customer3;
}

void Suite::room_info_print()
{

```

```

cout<<"房间号: "<<this->num<<"\t";
cout<<"价格: "<<this->price<<"/天"<<"\t";
switch(this->type)
{
case 0:
    cout<<"类型: 经济间"<<"\t";
    break;
case 1:
    cout<<"类型: 标准间"<<"\t";
    break;
case 2:
    cout<<"类型: 豪华间"<<"\t";
    break;
}
if(is_check_in)
{
    cout<<"状态: 已入住"<<endl;
    cout<<"入住顾客1: "<<this->customer1.get_name()<<"\t";
    cout<<"入住顾客2: "<<this->customer2.get_name()<<"\t";
    cout<<"入住顾客3: "<<this->customer3.get_name()<<"\t";
    cout<<"入住天数: "<<this->days<<"\t";
    cout<<"入住日期: "<<ctime(&(this->date));
}
else
{
    cout<<"状态: 空闲"<<"\t";
}
cout<<endl;
}
}

```

Manager.h

```

#ifndef MANAGER_H
#define MANAGER_H
class Manager
{
public:
    void check_in();           //客人入住办理
    void check_out();          //客人退房办理
    void show_customer_info(); //展示所有顾客的信息
    void show_room_info();     //展示所有房间的信息
    void show_uncheck_room_info(); //展示空闲的房间的信息
    void show_check_room_info(); //展示所有已入住的房间的信息
    void search_room();        //输入房间号，查询房间号信息

```



```

void search_customer();           //输入客人姓名，查询客人信息
void modify_room_info();         //修改房间价格

void del_customer_info(char* name);
void show_single_room(bool check_in); //显示单人间房间的信息
void show_double_room(bool check_in); //显示双人间房间的信息
void show_suites(bool check_in);      //显示套间的房间信息
void init_room_info();              //初始化(文件)数据库
};
#endif

```

Manager.cpp

```

#include "Manager.h"
#include "Customer.h"
#include "Room.h"
#include "Single_room.h"
#include "Double_room.h"
#include "Suite.h"
#include <fstream>
#include <iostream>
using namespace std;

/*客人入住办理*/
void Manager::check_in()
{
    cout<<"选择房间类型  1: 单人间  2: 双人间  3: 套间"<<endl;
    int select;
    do{
        cout<<"选择: ";
        cin>>select;
    }while(select<1 || select>3);
    int room_num;
    char name[10];
    char id[19];
    int days;
    time_t date;
    bool search=false;
    streampos flag; //相对起点的位置
    switch(select)
    {
        case 1:

```

```

    {
        cout<<"-----当前空闲的单人间如下-----"
        -----"<<endl;
        this->show_single_room(UNCHECK);           //显示空闲的
        单人间信息
        cout<<"-----"
        -----"<<endl;

        fstream file1("Single_room_info.txt",ios::in|ios::out);
        if(!file1)
        {
            cout<<"单人间数据库访问失败!";
            exit(0);
        }
        Single_room room;
        do{
            cout<<"输入房间号: ";
            cin>>room_num;
            file1.clear();           //将文件的状态初始化
            file1.seekg(0L,ios::beg); //跳转到文件开头

            while(!file1.eof())
            {
                flag=file1.tellg();           //得到文件位置
                指针，便于数据的修改
                file1.read((char*)&room,sizeof(room));
                if(room.get_num()==room_num &&
                room.get_is_check_in()==false) //查找到该房间号
                {
                    search=true;
                    break;
                }
            }

            if(!search)
            {
                cout<<"房间号错误或该房间已有顾客入住!请重新输入房间号"
                "<<endl;
            }
        }while(!search); //对每次输入的房间号进行检查，看是否存在和空闲

        cout<<"输入顾客姓名: ";
        cin>>name;
        cout<<"输入身份证号: ";
    }

```

```

        cin>>id;
        cout<<"输入入住天数: ";
        cin>>days;

        date=time(0);

        Customer customer;
        customer.info_set(name,id,room_num,days,date);

        room.set_days(days);
        room.set_is_check_in(true);
        room.set_cutomer(customer);
        room.set_date(date);

        fstream file2("Customer_info.txt",ios::in|ios::out|ios::app);
        if(!file2)
        {
            cout<<"客人信息数据库访问失败";
            exit(0);
        }
        file1.seekp(flag,ios::beg);
        file1.write((char*)&room,sizeof(room));           //重新写入房间
        file2.write((char*)&customer,sizeof(customer));   //写入客人信

        file1.close();
        file2.close();
        cout<<"入住办理成功! "<<endl;
        break;
    }
    case 2:
    {
        cout<<"-----当前空闲的双人间如下-----"
        <<endl;
        this->show_double_room(UNCHECK);
        cout<<"-----"
        <<endl;
        fstream file1("Double_room_info.txt",ios::in|ios::out);
        if(!file1)
        {
            cout<<"双人间数据库访问失败!";
            exit(0);
        }
    }
}

```

信息

息

```

Double_room room;
do{
    cout<<"输入房间号: ";
    cin>>room_num;
    file1.clear();           //将文件的状态初始化
    file1.seekg(0L,ios::beg); //跳转到文件开头

    while(!file1.eof())
    {
        flag=file1.tellg();           //得到文件位置
        指针,便于数据的修改
        file1.read((char*)&room,sizeof(room));
        if(room.get_num()==room_num &&
room.get_is_check_in()==false) //查找到该房间号
        {
            search=true;
            break;
        }
    }
    if(!search)
    {
        cout<<"房间号错误或该房间已有顾客入住!请从新输入房间号
"<<endl;
    }
}while(!search);           //对每次输入的房间号进行检查,看是否存在或空
闲

```

```

cout<<"输入顾客1姓名: ";
cin>>name;
cout<<"输入顾客1身份证号: ";
cin>>id;
cout<<"输入入住天数: ";
cin>>days;
date=time(0);
Customer customer1;
customer1.info_set(name,id,room_num,days,date); //顾客1信息设置

cout<<"输入顾客2姓名: ";
cin>>name;
cout<<"输入顾客2身份证号: ";
cin>>id;
Customer customer2;
customer2.info_set(name,id,room_num,days,date); //顾客2信息设置

```

```

room.set_days(days);
room.set_is_check_in(true);
room.set_cutomer1(customer1);
room.set_cutomer2(customer2);
room.set_date(date); //房间信息设置

fstream file2("Customer_info.txt",ios::in|ios::out|ios::app);
if(!file2)
{
    cout<<"客人信息数据库访问失败";
    exit(0);
}
file1.seekp(flag,ios::beg);
file1.write((char*)&room,sizeof(room)); //重新写入房间

file2.write((char*)&customer1,sizeof(customer1)); //写入顾客1

file2.write((char*)&customer2,sizeof(customer2)); //写入顾客2

file1.close();
file2.close();
cout<<"入住办理成功! "<<endl;
break;
}
case 3:
{
    cout<<"-----当前空闲的套间如下-----"
    <<endl;
    this->show_suites(UNCHECK);
    cout<<"-----"
    <<endl;
    fstream file1("Suites.txt",ios::in|ios::out);
    if(!file1)
    {
        cout<<"套间数据库访问失败!";
        exit(0);
    }
    Suite room;
    do{
        cout<<"输入房间号: ";
        cin>>room_num;
        file1.clear(); //将文件的状态初始化

```

```

        file1.seekg(0L,ios::beg); //跳转到文件开头

        while(!file1.eof())
        {
            flag=file1.tellg(); //得到文件位置
            指针，便于数据的修改
            file1.read((char*)&room,sizeof(room));
            if(room.get_num()==room_num &&
            room.get_is_check_in()==false) //查找到该房间号
            {
                search=true;
                break;
            }
        }

        if(!search)
        {
            cout<<"房间号错误或该房间已有顾客入住!请重新输入房间号
            "<<endl;
        }
    }while(!search); //对每次输入的房间号进行检查，看是否存在和空闲
    cout<<"输入顾客1姓名: ";
    cin>>name;
    cout<<"输入顾客1身份证号: ";
    cin>>id;
    cout<<"输入入住天数: ";
    cin>>days;
    date=time(0);
    Customer customer1;
    customer1.info_set(name,id,room_num,days,date); //顾客1信息设置

    cout<<"输入顾客2姓名: ";
    cin>>name;
    cout<<"输入顾客2身份证号: ";
    cin>>id;
    Customer customer2;
    customer2.info_set(name,id,room_num,days,date); //顾客2信息设置

    cout<<"输入顾客3姓名: ";
    cin>>name;
    cout<<"输入顾客3身份证号: ";
    cin>>id;
    Customer customer3;
    customer3.info_set(name,id,room_num,days,date); //顾客3信息设置

```

```

        room.set_days(days);
        room.set_is_check_in(true);
        room.set_cutomer1(customer1);
        room.set_cutomer1(customer2);
        room.set_cutomer3(customer3);
        room.set_date(date); //房间信息设置

        fstream file2("Customer_info.txt",ios::in|ios::out|ios::app);
        if(!file2)
        {
            cout<<"客人信息数据库访问失败";
            exit(0);
        }
        file1.seekp(flag,ios::beg);
        file1.write((char*)&room,sizeof(room)); //重新写入房间
信息
        file2.write((char*)&customer1,sizeof(customer1)); //写入客人1
信息
        file2.write((char*)&customer2,sizeof(customer2)); //写入客人2
信息
        file2.write((char*)&customer3,sizeof(customer3)); //写入客人3
信息

        file1.close();
        file2.close();
        cout<<"入住办理成功! "<<endl;
        break;
    }
}
}
/*客人退房办理*/
void Manager::check_out()
{
    fstream file1("Single_room_info.txt",ios::in|ios::out);
    if(!file1)
    {
        cout<<"单人间数据库访问失败";
        exit(0);
    } //打开单人间数据库

    fstream file2("Double_room_info.txt",ios::in|ios::out);
    if(!file2)

```

```

{
    cout<<"双人间数据库访问失败";
    exit(0);
} //打开双人间数据库

fstream file3("Suites.txt",ios::in|ios::out);
if(!file3)
{
    cout<<"双人间数据库访问失败";
    exit(0);
} //打开套间数据库

int room_num;
cout<<"输入房间号: ";
cin>>room_num;
bool search=false;
streampos flag;
while(!file1.eof())
{
    Single_room room;
    flag=file1.tellg();
    file1.read((char*)&room,sizeof(room));
    if(room.get_num()==room_num && room.get_is_check_in()==true)
    {
        search=true;
        cout<<"-----房间信息-----"
" <<endl;
        room.room_info_print();
        cout<<"需付费: " <<room.get_price() *room.get_days() <<"元
" <<endl;

        room.set_is_check_in(false);
        room.set_days(0);

        del_customer_info(room.get_customer().get_name());
        //顾客信息删除

        file1.seekp(flag,ios::beg);
        file1.write((char*)&room,sizeof(room));
        cout<<"退房办理成功!" <<endl;
        break;
    }
}
while(!file2.eof())

```



```

{
    Double_room room;
    flag=file1.tellg();
    file2.read((char*)&room,sizeof(room));
    if(room.get_num()==room_num && room.get_is_check_in()==true)
    {
        search=true;
        cout<<"-----房间信息-----"
" <<endl;

        room.room_info_print();
        cout<<"需付费用: " <<room.get_price() *room.get_days()<<"元
" <<endl;

        room.set_is_check_in(false);
        room.set_days(0);

        del_customer_info(room.get_customer1().get_name());
        del_customer_info(room.get_customer2().get_name());
        //顾客信息删除

        file1.seekp(flag,ios::beg);
        file1.write((char*)&room,sizeof(room));
        cout<<"退房办理成功!"<<endl;
        break;
    }
}
while(!file3.eof())
{
    Suite room;
    flag=file1.tellg();
    file3.read((char*)&room,sizeof(room));
    if(room.get_num()==room_num && room.get_is_check_in()==true)
    {
        search=true;
        cout<<"-----房间信息-----"
" <<endl;

        room.room_info_print();
        cout<<"需付费用: " <<room.get_price() *room.get_days()<<"元
" <<endl;

        room.set_is_check_in(false);
        room.set_days(0);

```

```

        del_customer_info(room.get_customer1().get_name());
        del_customer_info(room.get_customer2().get_name());
        del_customer_info(room.get_customer3().get_name());
        //顾客信息删除

        file1.seekp(flag, ios::beg);
        file1.write((char*)&room, sizeof(room));
        cout<<"退房办理成功!"<<endl;
        break;
    }
}
if(search==false)
{
    cout<<"房间号错误或该房间没有人入住!"<<endl;
}
file1.close();
file2.close();
file3.close();
}

/*显示所有房间信息*/
void Manager::show_room_info()
{
    cout<<endl<<"-----单人间-----"
    <<endl;
    this->show_single_room(false);
    this->show_single_room(true);

    cout<<endl<<"-----双人间-----"
    <<endl;
    this->show_double_room(false);
    this->show_double_room(true);

    cout<<endl<<"-----套间-----"
    <<endl;
    this->show_suites(false);
    this->show_suites(true);
}

/*显示所有未入住房间信息*/
void Manager::show_uncheck_room_info()
{
    cout<<"-----空闲房间查询-----"

```

```

-----"<<endl;
    cout<<"选择要查询的房间类型!    1: 单人间    2: 双人间    3: 套间    4: 所有
类型房间"<<endl;
    int select;
    do
    {
        cout<<"选择: ";
        cin>>select;
    }while(select<1 || select>4);
    switch(select)
    {
    case 1:
        {
            this->show_single_room(false);
            break;
        }
    case 2:
        {
            this->show_double_room(false);
            break;
        }
    case 3:
        {
            this->show_suites(false);
            break;
        }
    case 4:
        {

            cout<<"-----单人间-----
-----"<<endl;
            this->show_single_room(false);
            cout<<"-----双人间-----
-----"<<endl;
            this->show_double_room(false);
            cout<<"-----套间-----
-----"<<endl;
            this->show_suites(false);
            break;
        }
    }
}

/*显示所有已入住房间信息*/

```

```

void Manager::show_check_room_info()//展示所有已入住的房间
{
    cout<<"-----已入住房间查询-----"
    -----<<endl;
    cout<<"选择要查询的剩余的房间类型!    1: 单人间    2: 双人间    3: 套间    4:
所有类型房间"<<endl;
    int select;
    do
    {
        cout<<"选择: ";
        cin>>select;
    }while(select<1 || select>4);
    switch(select)
    {
        case 1:
        {
            cout<<"-----单人间-----"
            -----<<endl;
            this->show_single_room(true);
            break;
        }
        case 2:
        {
            cout<<"-----双人间-----"
            -----<<endl;
            this->show_double_room(true);
            break;
        }
        case 3:
        {
            this->show_suites(true);
            break;
        }
        case 4:
        {
            cout<<"-----单人间-----"
            -----<<endl;
            this->show_single_room(true);
            cout<<"-----双人间-----"
            -----<<endl;
            this->show_double_room(true);
            cout<<"-----套间-----"
            -----<<endl;
            this->show_suites(true);
        }
    }
}

```

```

        break;
    }
}

/*显示所有客人信息*/
void Manager::show_customer_info()
{
    int num=0;
    cout<<"-----所有入住客人信息如下-----"
    "<<endl;
    fstream file("Customer_info.txt",ios::in|ios::out);
    if(!file)
    {
        cout<<"用户信息数据库访问失败";
        exit(0);
    }
    Customer customer;
    file.clear();
    file.seekg(0,ios::beg);

    file.read((char*)&customer,sizeof(customer));
    while(!file.eof())
    {
        customer.info_print();
        cout<<endl;
        file.read((char*)&customer,sizeof(customer));
        num++;
    }
    if(num==0)
    {
        cout<<"暂且没有客人入住"<<endl;
    }
    cout<<"-----"
    "--"<<endl;
    file.close();
}

/*输入房间号或价格，查询符合的房间信息*/
void Manager::search_room()
{
    int select;
    do
    {

```

```

        cout<<"选择查询模式  1: 按房间号查询    2: 按价格区间查询"<<endl;
        cout<<"选择: ";
        cin>>select;

    }while(select<1 || select>2);

    fstream file1("Single_room_info.txt",ios::in|ios::out);
    if(!file1)
    {
        cout<<"单人间数据库访问失败";
        exit(0);
    }//打开单人间数据库

    fstream file2("Double_room_info.txt",ios::in|ios::out);
    if(!file2)
    {
        cout<<"双人间数据库访问失败";
        exit(0);
    }//打开双人间数据库

    fstream file3("Suites.txt",ios::in|ios::out);
    if(!file3)
    {
        cout<<"双人间数据库访问失败";
        exit(0);
    }//打开套间数据库

    switch(select)
    {
    case 1://按房间号查询
        {
            int room_num;
            cout<<"输入房间号: ";
            cin>>room_num;
            while(!file1.eof())
            {
                Single_room room;
                file1.read((char*)&room,sizeof(room));
                if(room.get_num()==room_num)
                {
                    room.room_info_print();
                    break;
                }
            }
        }
    }

```

```

while(!file2.eof())
{
    Double_room room;
    file2.read((char*)&room,sizeof(room));
    if(room.get_num()==room_num)
    {
        room.room_info_print();
        break;
    }
}
while(!file3.eof())
{
    Suite room;
    file3.read((char*)&room,sizeof(room));
    if(room.get_num()==room_num)
    {
        room.room_info_print();
        break;
    }
}
break;
}

case 2://按价格区间查询
{
    int low_price;
    int heigh_price;
    cout<<"输入价格区间: "<<endl;
    cout<<"最低价: ";
    cin>>low_price;
    cout<<"最高价: ";
    cin>>heigh_price;

    while(!file1.eof())
    {
        Single_room room;
        file1.read((char*)&room,sizeof(room));
        if(low_price<room.get_price() &&
room.get_price()<heigh_price)
        {
            room.room_info_print();
        }
    }
}

```

```

        while(!file2.eof())
        {
            Double_room room;
            file2.read((char*)&room,sizeof(room));
            if(low_price<room.get_price() &&
room.get_price()<heigh_price)
            {
                room.room_info_print();
            }
        }
        while(!file3.eof())
        {
            Suite room;
            file3.read((char*)&room,sizeof(room));
            if(low_price<room.get_price() &&
room.get_price()<heigh_price)
            {
                room.room_info_print();
            }
        }
        break;
    }
}
file1.close();
file2.close();
file3.close();
}

/* 输入客人姓名或身份证号，查询客人信息*/
void Manager::search_customer()
{
    int select;
    char name[10];
    char id[19];
    do
    {
        cout<<"选择查询模式    1: 按姓名查询    2: 按身份证号查询"<<endl;
        cout<<"选择: ";
        cin>>select;

    }while(select<1 || select>2);
    if(select==1)
    {
        cout<<"输入客人姓名: ";
    }
}

```



```

cin>>name;

}
else
{
    cout<<"输入身份证号: ";
    cin>>id;
}
fstream file("Customer_info.txt",ios::in|ios::out);
if(!file)
{
    cout<<"客人信息数据库访问失败"<<endl;
    exit(0);
}
bool flag=false;//标记是否查询到
file.clear();
file.seekg(0,ios::beg);
while(!file.eof())
{
    Customer customer;
    file.read((char *)&customer,sizeof(customer));
    if(strcmp(name,customer.get_name())==0 ||
strcmp(id,customer.get_id())==0)
    {
        customer.info_print();
        flag=true;
        break;
    }
}
if(!flag)
{
    cout<<"没有查询到该客人的信息!"<<endl;
}
file.close();
}

void Manager::del_customer_info(char *name)
{
    fstream file1("Customer_info.txt",ios::in|ios::out);
    if(!file1)
    {
        cout<<"客人信息数据库访问失败";
        exit(0);
    }
}

```

```

fstream file2("temp.txt",ios::out);//文件不存在会自动生成
if(!file2)
{
    cout<<"临时文件生成失败! ";
    exit(0);
}

file1.clear();
file1.seekg(0,ios::beg);
Customer customer;
file1.read((char*)&customer,sizeof(customer));
while(!file1.eof())
{
    if(strcmp(customer.get_name(),name)!=0)
    {
        file2.write((char*)&customer,sizeof(customer));
    }
    file1.read((char*)&customer,sizeof(customer));
}
file1.close();
file2.close();
remove("Customer_info.txt");
rename("temp.txt","Customer_info.txt");

}
/*修改房间信息*/
void Manager::modify_room_info()
{
    int room_num;
    cout<<"输入房间号对其信息进行修改: ";
    cin>>room_num;

    fstream file1("Single_room_info.txt",ios::in|ios::out);
    if(!file1)
    {
        cout<<"单人间数据库访问失败";
        exit(0);
    }//打开单人间数据库

    fstream file2("Double_room_info.txt",ios::in|ios::out);
    if(!file2)
    {
        cout<<"双人间数据库访问失败";
        exit(0);
    }
}

```

```

} //打开双人间数据库

fstream file3("Suites.txt",ios::in|ios::out);
if(!file3)
{
    cout<<"套间数据库访问失败";
    exit(0);
} //打开套间数据库

fstream file4("Customer_info.txt",ios::in|ios::out);
if(!file4)
{
    cout<<"顾客信息数据库访问失败";
    exit(0);
}

streampos flag1,flag2,flag3;    //用于标记三个文件中读到符合的房间信息的
位置
int room_type=0;                //用于标记输入的房间号属于什么类型的房间,
Single_room room1;
Double_room room2;
Suite room3;
while(!file1.eof())
{
    flag1=file1.tellg();
    file1.read((char*)&room1,sizeof(room1));
    if(room1.get_num()==room_num && room1.get_is_check_in()==false)
    {
        room_type=1;
        cout<<"-----该房间信息-----"
"<<endl;
        room1.room_info_print();
        cout<<"-----"
"<<endl;
        break;
    }
}
while(!file2.eof())
{
    flag2=file2.tellg();
    file2.read((char*)&room2,sizeof(room2));
    if(room2.get_num()==room_num && room1.get_is_check_in()==false)
    {
        room_type=2;

```

```

        cout<<"-----该房间信息-----"
"<<endl;
        room2.room_info_print();
        cout<<"-----"
"<<endl;
        break;
    }
}
while(!file3.eof())
{
    flag3=file2.tellg();
    file3.read((char*)&room3,sizeof(room3));
    if(room3.get_num()==room_num && room1.get_is_check_in()==false)
    {
        room_type=3;
        cout<<"-----该房间信息-----"
"<<endl;
        room3.room_info_print();
        cout<<"-----"
"<<endl;
        break;
    }
}
if(room_type==0)
{
    cout<<"没有查找到该房间的信息，请核对输入是否正确"<<endl;
    return;
}

    cout<<"选择要修改的房间信息。  1: 房间号    2: 房间价格    3: 房间类型"
"<<endl;

```

```

int select;           //要修改的内容
int num;              //修改后的房间号
int type;             //修改后的类型
int price;            //修改后的价格
do
{
    cout<<"选择: ";
    cin>>select;
}while(select<1 || select>5);
switch(select)
{
case 1:

```

```

{
    cout<<"输入修改后的房间号: ";
    cin>>num;
    if(room_type==1)
    {
        room1.set_num(num);
        file1.seekp(flag1,ios::beg);
        file1.write((char*)&room1,sizeof(room1));
    }
    if(room_type==2)
    {
        room2.set_num(num);
        file2.seekp(flag2,ios::beg);
        file2.write((char*)&room2,sizeof(room2));
    }
    if(room_type==3)
    {
        room3.set_num(num);
        file3.seekp(flag3,ios::beg);
        file3.write((char*)&room3,sizeof(room3));
    }
    cout<<"修改成功! "<<endl;
    break;
}
case 2:
{
    cout<<"输入修改后的房间价格: ";
    cin>>price;
    if(room_type==1)
    {
        room1.set_price(price);
        file1.seekp(flag1,ios::beg);
        file1.write((char*)&room1,sizeof(room1));
    }
    if(room_type==2)
    {
        room2.set_price(price);
        file2.seekp(flag2,ios::beg);
        file2.write((char*)&room2,sizeof(room2));
    }
    if(room_type==3)
    {
        room3.set_price(price);
        file3.seekp(flag3,ios::beg);

```

```

        file3.write((char*)&room3,sizeof(room3));
    }
    break;
}
case 3:
{
    cout<<"输入修改后的房间类型:   0: 经济间   1: 标准间   2: 豪华间
" ;

    cin>>type;
    if(room_type==1 && type>=0 && type<=2)
    {
        room1.set_type(type);
        file1.seekp(flag1,ios::beg);
        file1.write((char*)&room1,sizeof(room1));
    }
    if(room_type==2 && type>=0 && type<=2)
    {
        room2.set_type(type);
        file2.seekp(flag2,ios::beg);
        file2.write((char*)&room2,sizeof(room2));
    }
    if(room_type==3 && type>=0 && type<=2)
    {
        room3.set_type(type);
        file3.seekp(flag3,ios::beg);
        file3.write((char*)&room3,sizeof(room3));
    }
    break;
}
}

file1.close();
file2.close();
file3.close();
file4.close();
}

/*初始化房间信息*/
void Manager::init_room_info()
{
    remove("Customer_info.txt");           //删除旧的用户信息文件
    fstream file("Customer_info.txt",ios::out); //创建新的空白用户信息文件
    if(!file)
    {

```

```

        cout<<"客人数据文件创建失败";
        exit(0);
    }
    fstream file1("Single_room_info.txt",ios::in|ios::out);
    fstream file2("Double_room_info.txt",ios::in | ios::out);
    fstream file3("Suites.txt",ios::in|ios::out);
    if(!file1 || !file2 || !file3)
    {
        cout<<"房间信息初始化失败";
        exit(0);
    }

    Single_room room1;
    int num=100;
    int price=150;
    for(int i=1;i<=3;i++)
    {
        room1.room_info_set(num+i,price,UNCHECK,ECONOMY_ROOM);//
        file1.write((char*)&room1,sizeof(room1));
    }//单人间, 经济间

    num=200;
    price=220;
    for(int i=1;i<=3;i++)
    {
        room1.room_info_set(num+i,price,UNCHECK,STANDARD_ROOM);//
        file1.write((char*)&room1,sizeof(room1));
    }//单人间, 标准间

    num=300;
    price=300;
    for(int i=1;i<=3;i++)
    {
        room1.room_info_set(num+i,price,UNCHECK,DELUXE_ROOM);//
        file1.write((char*)&room1,sizeof(room1));
    }//单人间, 豪华间

    Double_room room2;
    num=400;
    price=200;
    for(int i=1;i<=3;i++)
    {
        room2.room_info_set(num+i,price,UNCHECK,ECONOMY_ROOM);
    }

```

```

        file2.write((char*)&room2,sizeof(room2));
} // 双人间, 经济间

num=500;
price=250;
for(int i=1;i<=3;i++)
{
    room2.room_info_set(num+i,price,UNCHECK,STANDARD_ROOM);
    file2.write((char*)&room2,sizeof(room2));
} // 双人间, 标准间

num=600;
price=350;
for(int i=1;i<=3;i++)
{
    room2.room_info_set(num+i,price,UNCHECK,DELUXE_ROOM);
    file2.write((char*)&room2,sizeof(room2));
} // 双人间, 豪华间

Suite room3;
num=700;
price=300;
for(int i=1;i<=3;i++)
{
    room3.room_info_set(num+i,price,UNCHECK,ECONOMY_ROOM);
    file3.write((char*)&room3,sizeof(room3));
} // 套间, 经济间

num=800;
price=350;
for(int i=1;i<=3;i++)
{
    room3.room_info_set(num+i,price,UNCHECK,STANDARD_ROOM);
    file3.write((char*)&room3,sizeof(room3));
} // 套间, 标准间

num=900;
price=400;
for(int i=1;i<=3;i++)
{
    room3.room_info_set(num+i,price,UNCHECK,DELUXE_ROOM);
    file3.write((char*)&room3,sizeof(room3));
} // 套间, 豪华间

```



```

        file1.close();
        file2.close();
        file3.close();
    }

    //按照房间的类型和状态列出符合的房间信息
    void Manager::show_single_room(bool check_in)
    {
        Single_room room1;
        fstream file1("Single_room_info.txt",ios::in|ios::out);
        if(!file1)
        {
            cout<<"单人间数据库访问失败!";
            exit(0);
        }
        file1.read((char*)&room1,sizeof(room1));
        while(!file1.eof())
        {
            if(room1.get_is_check_in()==check_in)
            {
                room1.room_info_print();
            }
            file1.read((char*)&room1,sizeof(room1));//
        }
        file1.close();
    }

    void Manager::show_double_room(bool check_in)
    {
        Double_room room2;
        fstream file2("Double_room_info.txt",ios::in|ios::out);
        if(!file2)
        {
            cout<<"双人间数据库访问失败! ";
            exit(0);
        }
        file2.read((char*)&room2,sizeof(room2));
        while(!file2.eof())
        {
            if(room2.get_is_check_in()==check_in)
            {
                room2.room_info_print();
            }
            file2.read((char*)&room2,sizeof(room2));
        }
    }

```

```

    }
    file2.close();
}
void Manager::show_suites(bool check_in)
{
    Suite room3;
    fstream file3("Suites.txt",ios::in|ios::out);
    if(!file3)
    {
        cout<<"套间数据库访问失败！";
        exit(0);
    }
    file3.read((char *)&room3,sizeof(room3));
    while(!file3.eof())
    {
        if(room3.get_is_check_in()==check_in)
        {
            room3.room_info_print();
        }
        file3.read((char *)&room3,sizeof(room3));
    }
    file3.close();
}

```