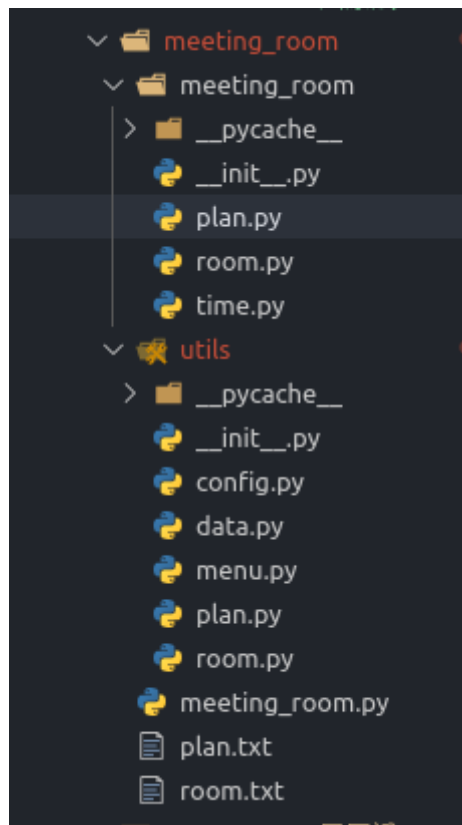


# 阶段1：综合实验 - 讨论室使用计划管理系统

## 1. 设计过程

### 1.1 代码结构



--目录结构--

meeting\_room -- 项目目录

meeting\_room/meeting\_room -- 定义数据存储类

meeting\_room/util -- 定义数据相关操作

meeting\_room.py -- 程序入口

plan.txt / room.txt -- 程序相关文件

#### 1.1.1 meeting\_room 目录

- meeting\_room.py -- 程序入口

```
import sys
from utils import data
from utils import menu

def main():
```

```

room_infos = [] # 会议室信息
plan_infos = [] # 计划信息
data.init_data(room_infos, plan_infos) # 初始化信息
# while True:
menu.main_menu() # 主菜单打印
menu.select_main_menu(room_infos, plan_infos)

if __name__ == '__main__':
    main()

```

### 1.1.2 meeting\_room/meeting\_room 目录

- plan.py -- 使用计划类

```

# 部分代码
class Plan:
    """ 讨论室使用计划类 """

    def __init__(self, id, room_id, date, begin_time, end_time,
contact):
        self._id = id # 计划 ID
        self._room_id = room_id # 房间号码
        self._date = date # 计划日期
        self._time = Time(begin_time, end_time) # 起止时间
        self._contact = contact # 联系人姓名

    def to_string(self):
        """ 转化为字符串 """
        return str(self._id) + ' ' + str(self._room_id) + ' ' + \
            self._date + ' ' + self._time.begin + ' ' + \
            self._time.end + ' ' + self._contact

```

- room.py -- 讨论室信息类

```

# 部分代码
class Room:
    """ 讨论室信息类 """

    def __init__(self, room_id, capacity, status=0, plan_id=-1):
        self._room_id = room_id # 房间号
        self._capacity = capacity # 可容纳人数
        self._status = status # 使用状态
        self._plan_id = plan_id # 计划ID

    def to_string(self):
        """ 数据转化为str 方便写入文件 """

```

```

        return str(self._room_id) + ' ' + str(self._capacity) + ' '
        + str(self._status) + ' ' + str(self._plan_id)

```

- **time.py** -- 起止时间类

```

# 部分代码
class Time:
    """ 时间类 """

    def __init__(self, begin, end):
        self._begin = begin
        self._end = end

```

### 1.1.3 meeting\_room/utils 目录

- **config.py** -- 定义一些项目变量

```

import sys

PATH = sys.path[0]
ROOM_PATH = PATH + '/room.txt'
PLAN_PATH = PATH + '/plan.txt'

def get_room_path():
    """ 获取room文件路径 """
    return ROOM_PATH

def get_plan_path():
    """ 获取plan文件路径 """
    return PLAN_PATH

```

- **data.py** -- 数据相关的操作定义

```

import os
from . import config
from meeting_room.room import Room
from meeting_room.plan import Plan

def init_data(room_infos, plan_infos):
    """ 初始化信息 """
    lines1 = read_file(config.get_room_path())
    lines2 = read_file(config.get_plan_path())

```

```

    for i, line in enumerate(lines1):
        # 初始化讨论室信息
        lines1[i] = line.replace('\n', '')
        r = lines1[i].split(' ')
        room_infos.append(Room(int(r[0]), int(r[1]), int(r[2]),
int(r[3])))
    for i, line in enumerate(lines2):
        # 初始化使用计划信息
        lines2[i] = line.replace('\n', '')
        p = lines2[i].split(' ')
        plan_infos.append(Plan(int(p[0]), int(p[1]), p[2], p[3],
p[4], p[5]))

def read_file(path):
    """ 读取文件函数 """
    if os.path.exists(path):
        with open(path, 'r') as f:
            return f.readlines()
    else:
        print('文件路径不存在')
        return []

def write_file(data, path):
    """ 写入文件函数 """
    d = []
    for r in data:
        d.append(r.to_string())
    with open(path, 'w+') as f:
        f.write('\n'.join(d))

```

- menu.py --定义系统所以菜单

```

# 部分代码
def main_menu():
    """ 主菜单打印 """
    tools.clear()
    print('{0:{1}^80}'.format('讨论室使用计划管理系统\n',
chr(12288)))
    print('{0:{1}^80}'.format('1.讨论室使用管理', chr(12288)))
    print('{0:{1}^80}'.format('2.讨论室信息管理', chr(12288)))
    print('{0:{1}^80}'.format('0.退出', chr(12288)))
    print('-'*40)
    # select_main_menu()

def select_main_menu(room_infos, plan_infos):
    """ 选择主菜单 """

```

```

select_id = int(input('输入序号: '))
tools.clear()
if select_id == 1: # 使用计划管理
    plan_menu()
    select_plan_menu(room_infos, plan_infos)
elif select_id == 2: # 讨论室信息管理
    room_menu()
    select_room_menu(room_infos, plan_infos)
elif select_id == 0: # 退出
    sys.exit('退出系统! ')
else:
    print('序号输入有误, 请重新输入! ')
    select_main_menu(room_infos, plan_infos)

```

- plan.py -- 使用计划信息相关操作

```

from meeting_room.plan import Plan
from meeting_room.time import Time
from . import room
from . import data
from . import config

def add(plan_infos, room_infos):
    """ 信息添加 """
    print('输入添加信息: 格式 (房间号 日期 起始时间 结束时间 联系人姓名) ')
    info = list(map(str, input('>>').split(' ')))
    i = room.find(room_infos, [], 0, room_id=int(info[0]))
    if i == -1:
        print('房间号不存在')
    else:
        id = plan_infos[len(plan_infos) - 1].id + 1
        plan_infos.append(
            Plan(id, int(info[0]), info[1], info[2], info[3],
info[4]))
        room_infos[i].status = 1
        room_infos[i].plan_id = id
        save(plan_infos) # 保存使用计划信息
        room.save(room_infos) # 保存讨论室信息
        print('添加成功')

# def delete():
#     """ 信息删除 """
#     pass

def update(plan_infos):
    """ 信息修改 """

```

```

print('输入修改ID: ')
id = int(input('>>'))
i = find(plan_infos, [], 0, id=id)
if i == -1:
    print('信息不存在')
else:
    print('输入修改后的信息: (日期、起止时间)')
    date = input('日期 >>')(代码部分)
    begin = input('起始时间 >>')
    end = input('终止时间 >>')
    plan_infos[i].date = date
    plan_infos[i].time = Time(begin, end)
    save(plan_infos)
    print('修改成功!!! ')

def find(plan_infos, room_infos=[], type=1, **kwargs):
    """ 信息查看 """
    if type == 0: # 信息检索 (id、日期、房间号)
        if 'id' in kwargs:
            for i, p in enumerate(plan_infos):
                if p.id == kwargs['id']:
                    return i
            return -1
        elif 'date' in kwargs:
            for i, p in enumerate(plan_infos):
                if p.date == kwargs['date']:
                    return i
            return -1
        else:
            print('输入查询日期和房间号:')
            date = input('>>')
            room_id = int(input('>>'))
            ps = []
            for p in plan_infos:
                if p.date == date and p.room_id == room_id:
                    ps.append(p)
            print_info(ps)
    else: # 全部信息查看
        print_info(plan_infos)

def save(plan_infos):
    """ 信息保存 """
    data.write_file(plan_infos, config.get_plan_path())

def print_info(infos):
    """ 打印信息 """
    tpl1 = '{:^10}\t{:^10}\t{:^20}\t{:^20}\t{:^10}'
    t_tpl1 = '{:^10}\t{:^10}\t{:^15}\t{:^15}\t{:^10}'

```

```

        print(t_tplt.format('ID', '房间号', '计划日期', '起止时间', '联系人'))
    for info in infos:
        print(tplt.format(info.id, info.room_id,
                           info.date, str(info.time),
                           info.contact))

```

- **room.py** -- 讨论室相关操作

```

from meeting_room.room import Room
from . import data
from . import config
from . import plan

def add(room_infos):
    """ 信息添加 """
    print('输入会议室信息: 格式 (房间号 可容纳人数) ')
    r = list(map(int, input('>> ').split(' ')))
    if find(room_infos, [], 0, room_id=r[0]) != -1:
        print('信息已存在')
    else:
        room_infos.append(Room(r[0], r[1]))
        save(room_infos)
        print('添加成功!!! ')

def delete(room_infos):
    """ 信息删除 """
    pass

def update(room_infos):
    """ 信息修改 """
    print('输入修改房间号: ')
    room_id = int(input('>>'))
    i = find(room_infos, [], 0, room_id=room_id)
    if i == -1:
        print('房间号不存在')
    else:
        print('输入修改后的信息: (可容纳人数) ')
        capacity = int(input('>>'))
        room_infos[i].capacity = capacity
        save(room_infos)
        print('修改成功!!! ')

def find(room_infos, plan_infos=[], type=2, **kwargs):
    """ 信息查看 """

```

```

if type == 0: # 信息检索 (房间号、日期、人数)
    if 'room_id' in kwargs: # 房间号
        for i, r in enumerate(room_infos):
            if r.room_id == kwargs['room_id']:
                return i
        return -1
    else: # 日期 + 人数
        print('输入查询日期和人数:')
        date = input('>>')
        num = int(input('>>'))
        rs = []
        for r in room_infos:
            if r.capacity >= num:
                if r.status == 0 or plan.find(plan_infos, [],
0, date=date) == -1:
                    rs.append(r)
        print_info(rs)
    else: # 全部信息查看
        print_info(room_infos)

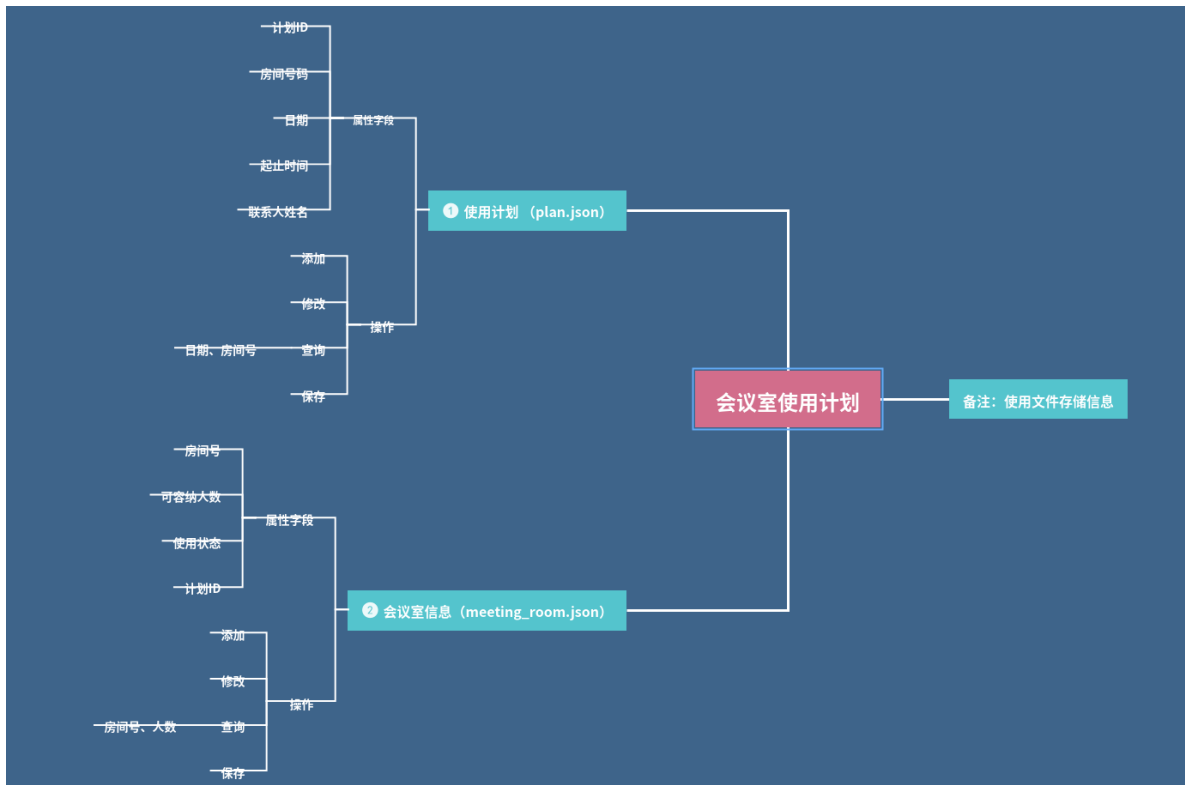
def save(room_infos):
    """ 信息保存 """
    data.write_file(room_infos, config.get_room_path())

def print_info(infos):
    """ 打印信息 """
    tplt = '{:^10}\t{:^10}\t{:^10}\t{:^10}'
    print(tplt.format('房间号', '可容纳人数', '使用状态', '计划ID'))
    for info in infos:
        status = '空闲' if info.status == 0 else '使用中'
        print(tplt.format(info.room_id, info.capacity, status,
info.plan_id))

```

## 1.2 设计说明





## 2. 测试结果

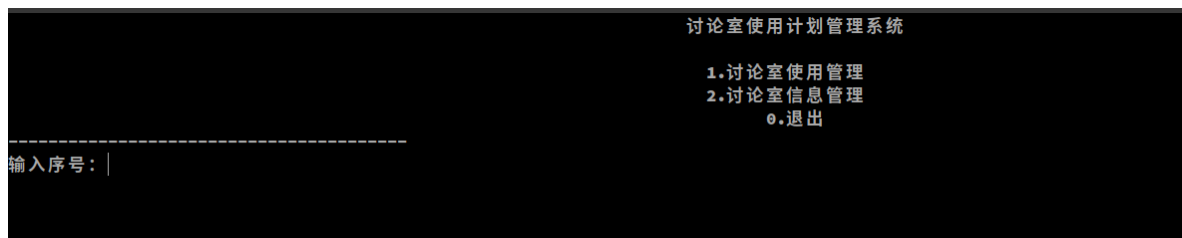
### 2.1 测试数据

```
/room.txt 文件
# 房间号 可容纳人数 使用状态 (0-空闲 1-使用中) 使用计划ID
101 100 1 1
102 50 0 -1
103 30 0 -1
104 100 0 -1
201 50 0 -1
202 50 1 2
203 30 0 -1
204 100 0 -1
301 50 0 -1
302 100 0 -1
```

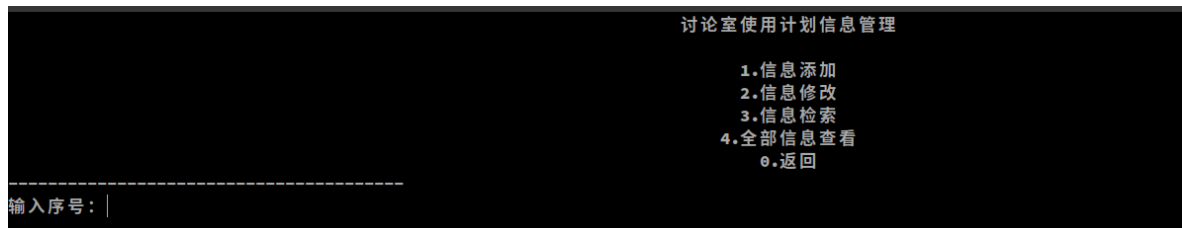
```
/plan.txt 文件
# 计划ID 房间号 日期 起始时间 终止时间 联系人姓名
0 101 2019-10-21 13:00 14:00 张三
1 101 2018-10-10 1:00 2:00 李四
2 202 2019-10-10 13:00 14:00 李四
```

### 2.2 测试结果

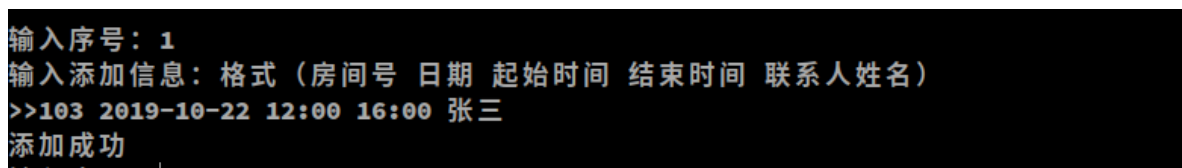
#### 2.2.1 主菜单



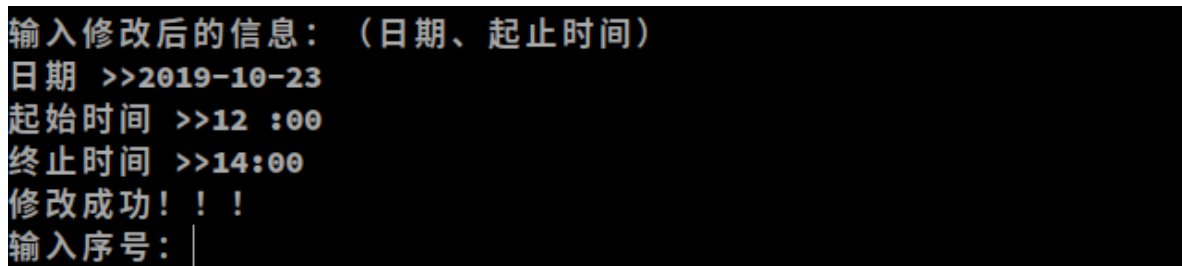
## 2.2.2 讨论室使用管理



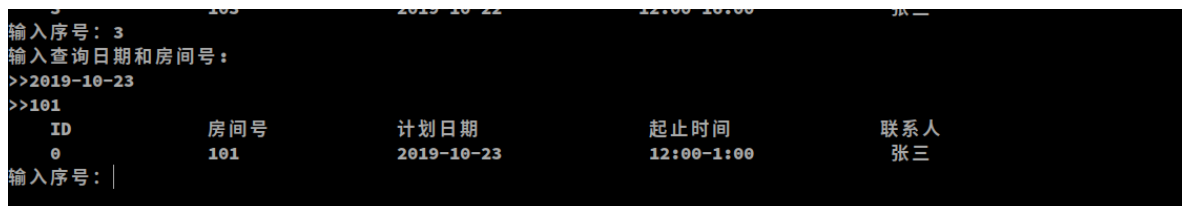
- 信息添加



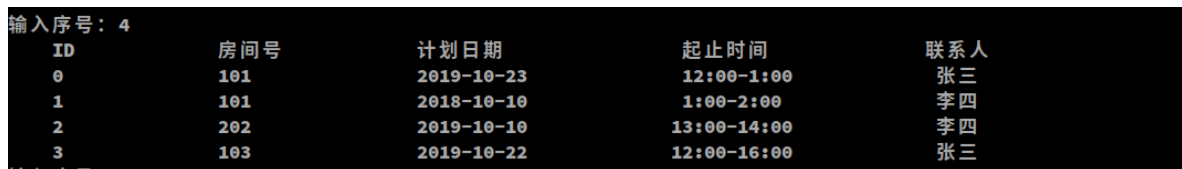
- 信息修改



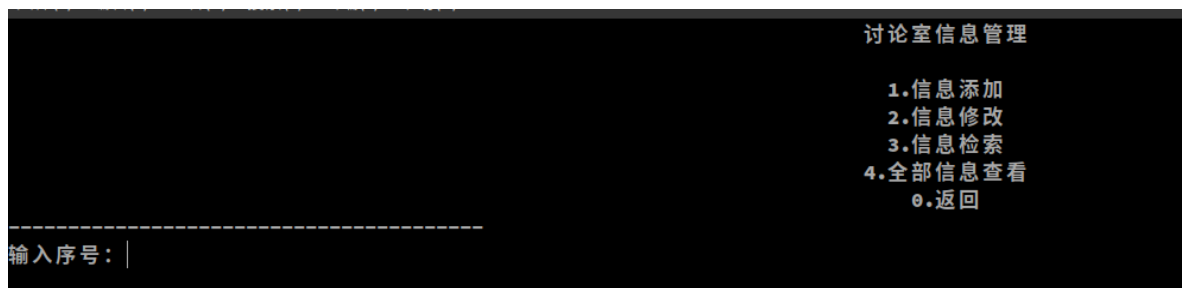
- 信息检索（日期、房间号）



- 全部信息查看



## 2.2.3 讨论室信息管理



- 信息添加

```
输入序号：1
输入会议室信息：格式（房间号 可容纳人数）
>> 310 100
添加成功!!!
输入序号：
```

- 信息修改

```
输入序号：2
输入修改房间号：
>>101
输入修改后的信息：（可容纳人数）
>>200
修改成功!!!
输入序号：
```

- 信息检索（日期、人数）

```
输入序号：3
输入查询日期和人数：
>>2019-10-21
>>20
  房间号      可容纳人数      使用状态      计划ID
  101         200          使用中        1
  102         50           空闲          -1
  103         30           使用中        3
  104         100          空闲          -1
  201         50           空闲          -1
  202         50           使用中        2
  203         30           空闲          -1
  204         100          空闲          -1
  301         50           空闲          -1
  302         100          空闲          -1
  310         100          空闲          -1
```

- 全部信息查看

```
输入序号：4
  房间号      可容纳人数      使用状态      计划ID
  101         200          使用中        1
  102         50           空闲          -1
  103         30           使用中        3
  104         100          空闲          -1
  201         50           空闲          -1
  202         50           使用中        2
  203         30           空闲          -1
  204         100          空闲          -1
  301         50           空闲          -1
  302         100          空闲          -1
  310         100          空闲          -1
```