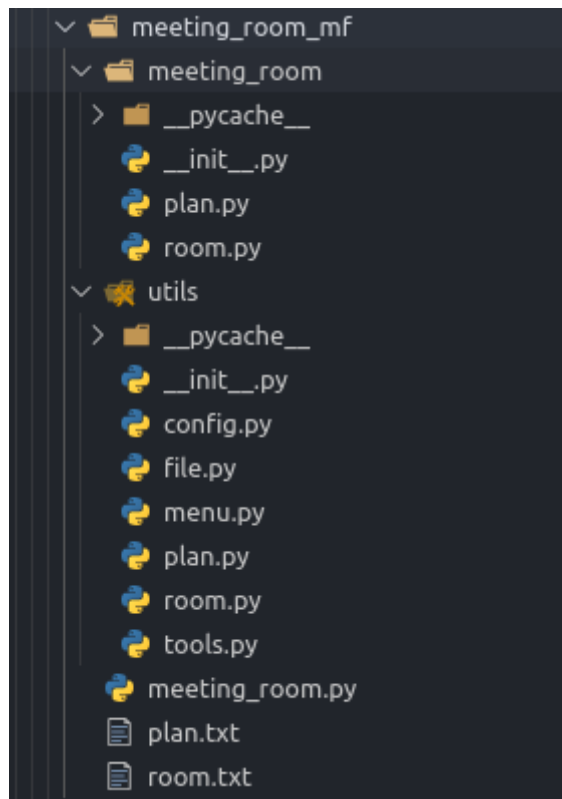


讨论室使用计划系统

1. 代码结构



--目录结构--

meeting_room_mf -- 项目目录

meeting_room_mf/meeting_room -- 定义数据存储类

meeting_room_mf/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 meeting_room_mf 目录

- meeting_room.py -- 程序入口

```
import sys
from utils import menu
from utils import room
from utils import plan
from utils import tools

def main():
    room_infos = [] # 会议室信息
```

```

plan_infos = [] # 计划信息
room.init(room_infos) # 初始化讨论室信息
plan.init(plan_infos) # 初始化使用计划信息
tools.clear() # 清除输出
menu.main_menu() # 主菜单打印
menu.select_main_menu(room_infos, plan_infos)

if __name__ == '__main__':
    main()

```

1.2 meeting_room_mf/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._b_time = begin_time # 起始时间
        self._e_time = end_time # 结束时间
        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._b_time + ' ' + self._e_time
+ ' ' + 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)

```

1.3 meeting_room_mf/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

```

- **file.py** -- 数据相关的操作定义

```

import os

def read(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(infos, path):
    """ 写入文件函数 """
    lines = []
    for info in infos:
        lines.append(info.to_string())
    with open(path, 'w+') as f:

```

```
f.write('\n'.join(lines))
```

- menu.py --定义系统所以菜单

```
import sys
import os
from . import tools
from . import room
from . import plan

def main_menu():
    """ 主菜单打印 """
    print('\n\t讨论室使用计划管理系统\n')
    print('\t-- 1.使用计划管理')
    print('\t-- 2.讨论室信息管理')
    print('\t-- 0.退出')
    print('-'*40)

def room_menu():
    """ 讨论室信息菜单 """
    print('\n\t讨论室信息管理\n')
    print('\t-- 1.信息添加')
    print('\t-- 2.信息修改')
    print('\t-- 3.信息检索')
    print('\t-- 4.全部信息查看')
    print('\t-- 0.返回')
    print('-'*40)

def plan_menu():
    """ 使用计划菜单 """
    print('\n\t讨论室使用计划信息管理\n')
    print('\t-- 1.信息添加')
    print('\t-- 2.信息修改')
    print('\t-- 3.信息检索')
    print('\t-- 4.全部信息查看')
    print('\t-- 0.返回')
    print('-'*40)

def select_main_menu(room_infos, plan_infos):
    """ 选择主菜单 """
    select_id = tools.int_input()
    while select_id != 0:
        if select_id == 1: # 使用计划管理
            tools.clear() # 清除终端输出
            plan_menu()
```

```

        select_plan_menu(room_infos, plan_infos)
    elif select_id == 2: # 讨论室信息管理
        tools.clear()
        room_menu()
        select_room_menu(room_infos, plan_infos)
    else:
        print('序号输入有误, 请重新输入! ')
        select_main_menu(room_infos, plan_infos)
else:
    sys.exit('退出系统! ')

def select_room_menu(room_infos, plan_infos):
    """ 讨论室信息菜单选择 """
    select_id = tools.int_input()
    while select_id != 0:
        if select_id == 1: # 信息添加
            room.add(room_infos)
        elif select_id == 2: # 信息修改
            room.update(room_infos)
        elif select_id == 3: # 信息检索
            print('输入查询日期和人数:')
            date = input('>> 日期(YY-MM-DD): ')
            num = int(input('>> 人数: '))
            room.find(room_infos, 0, plan_infos, date=date,
num=num)
        elif select_id == 4: # 全部信息查看
            room.find(room_infos, 1, plan_infos)
        else:
            print('Err:序号输入有误, 请重新输入! ')
            select_room_menu(room_infos, plan_infos)
        select_id = tools.int_input()
    else:
        tools.clear()
        main_menu()
        select_main_menu(room_infos, plan_infos)

def select_plan_menu(room_infos, plan_infos):
    """ 使用计划菜单选择 """
    select_id = tools.int_input()
    while select_id != 0:
        if select_id == 1: # 信息添加
            plan.add(plan_infos, room_infos)
        elif select_id == 2: # 信息修改
            plan.update(plan_infos)
        elif select_id == 3: # 信息检索
            print('输入查询日期和房间号:')
            date = input('>> 日期:')
            room_id = int(input('>> 房间号:'))
            plan.find(plan_infos, 0, room_infos, date=date,
room_id=room_id)

```

```

elif select_id == 4: # 全部信息查看
    plan.find(plan_infos, 1)
else:
    print('序号输入有误, 请重新输入! ')
    select_plan_menu(room_infos, plan_infos)
select_id = tools.int_input()
else:
    tools.clear()
    main_menu()
    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 . import room
from . import file as f
from . import config
from . import tools

def init(plan_infos):
    """ 初始化数据 """
    lines = f.read(config.get_plan_path())
    for i, line in enumerate(lines):
        # 初始化使用计划信息
        lines[i] = line.replace('\n', ' ')
        p = lines[i].split(' ')
        plan_infos.append(Plan(int(p[0]), int(p[1]), p[2], p[3],
p[4], p[5]))

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(' ')))
    if len(info) == 5:
        i = room.find(room_infos, room_id=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, 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! ')

```

```

else:
    print('输入格式有误, 请重新输入! ')
    add(plan_infos, room_infos)

def update(plan_infos):
    """ 信息修改 """
    print('输入修改ID: ')
    id = int(input('>> '))
    i = find(plan_infos, 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:')
        begin_time = input('>> 起始时间:')
        end_time = input('>> 终止时间:')
        plan_infos[i].date = date
        plan_infos[i].b_time = begin_time
        plan_infos[i].e_time = end_time
        save(plan_infos)
        print('修改成功!!! ')

def find(plan_infos, type=0, room_infos=[], **kwargs):
    """ 信息查看 """
    if type == 0: # 条件查询
        if 'id' in kwargs: # 根据ID查询
            for i, p in enumerate(plan_infos):
                if p.id == kwargs['id']:
                    return i
            return -1
        elif 'date' in kwargs and 'room_id' in kwargs: # 根据日期
和房间号查询
            ps = []
            for p in plan_infos:
                if p.date == kwargs['date'] and p.room_id ==
kwargs['room_id']:
                ps.append(p)
            print_info(ps)
            return
        elif 'date' in kwargs: # 根据日期查询
            for i, p in enumerate(plan_infos):
                if p.date == kwargs['date']:
                    # print(kwargs['date'])
                    return i
            return -1
        elif type == 1: # 全部信息查看
            print_info(plan_infos)

def save(plan_infos):

```

```

""" 信息保存 """
f.write(plan_infos, config.get_plan_path())

def print_info(infos):
    """ 打印信息 """
    tplt = '{:^10}\t{:^10}\t{:^10}\t{:^10}\t{:^10}'
    print('\n全部信息: \n')
    print(tplt.format('ID', '房间号', '计划日期', '起止时间', '联系人'))
    print('-'*80)
    for info in infos:
        print(tplt.format(info.id, info.room_id,
                           info.date, info.b_time + '-' +
info.e_time, info.contact))
    print('-'*80)

```

- **room.py** -- 讨论室相关操作

```

from meeting_room.room import Room
from . import file as f
from . import config
from . import plan
from . import tools

def init(room_infos):
    """ 初始化讨论室信息 """
    lines = f.read(config.get_room_path())
    for i, line in enumerate(lines):
        # 初始化讨论室信息
        lines[i] = line.replace('\n', ' ')
        r = lines[i].split(' ')
        room_infos.append(Room(int(r[0]), int(r[1]), int(r[2]),
int(r[3])))

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(str, input('>> ').split(' ')))
    if(len(r) == 2): # 输入格式错误处理
        if find(room_infos, room_id=r[0]) != -1:
            print('Warning:讨论室信息已存在! ')
        else:
            room_infos.append(Room(r[0], r[1]))
            save(room_infos)
            print('添加成功!!! ')
    else:

```



```

        print('Err:格式有误, 请重新输入! ')
        add(room_infos)

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

def find(room_infos, type=0, plan_infos=[], **kwargs):
    """ 信息查询 """
    if type == 0: # 条件查询
        if 'room_id' in kwargs: # 根据房间号查询
            for i, r in enumerate(room_infos):
                if str(r.room_id) == kwargs['room_id']:
                    return i
            return -1
        elif 'date' in kwargs and 'num' in kwargs: # 根据日期和人
            数查询
            rs = []
            for r in room_infos:
                if r.capacity >= kwargs['num']:
                    # 使用状态为 0 或 使用日期输入日期
                    if r.status == 0: # 未预约
                        rs.append(r)
                    else: # 已预约 查询预约时间
                        p = plan.find(plan_infos, id=r.plan_id)
                        # 获取 ID 为 plan_id 的索引
                        if plan_infos[p].date != kwargs['date']:
                            rs.append(r)
            print_info(rs)
        elif type == 1: # 全部信息查看
            print_info(room_infos, plan_infos)

def save(room_infos):
    """ 信息保存 """

```

```

f.write(room_infos, config.get_room_path())

def print_info(room_infos, plan_infos=[]):
    """ 打印信息 """
    tplt = '{:^10}\t{:^10}\t{:^10}\t{:^10}'
    print('\n全部信息:\n')
    print(tplt.format('房间号', '可容纳人数', '使用状态', '计划ID'))
    print('-'*60)
    for info in room_infos:
        status = '空闲' if info.status == 0 else '已预约'
        plan_id = info.plan_id if info.plan_id != -1 else ''
        print(tplt.format(info.room_id, info.capacity, status,
plan_id))
    print('-'*60)

```

- tools.py -- 工具文件

```

import platform
import os
from . import plan

BEGIN_TIME = '9:00'
TEMP_TIME = '15:00'
END_TIME = '21:00'

def _get_platform():
    """ 获取系统 """
    return platform.system()

def clear():
    """ 清除终端输出 """
    sysstr = _get_platform()
    if sysstr == "Windows":
        os.system('cls')
    elif sysstr == "Linux":
        os.system('clear')
    else:
        return

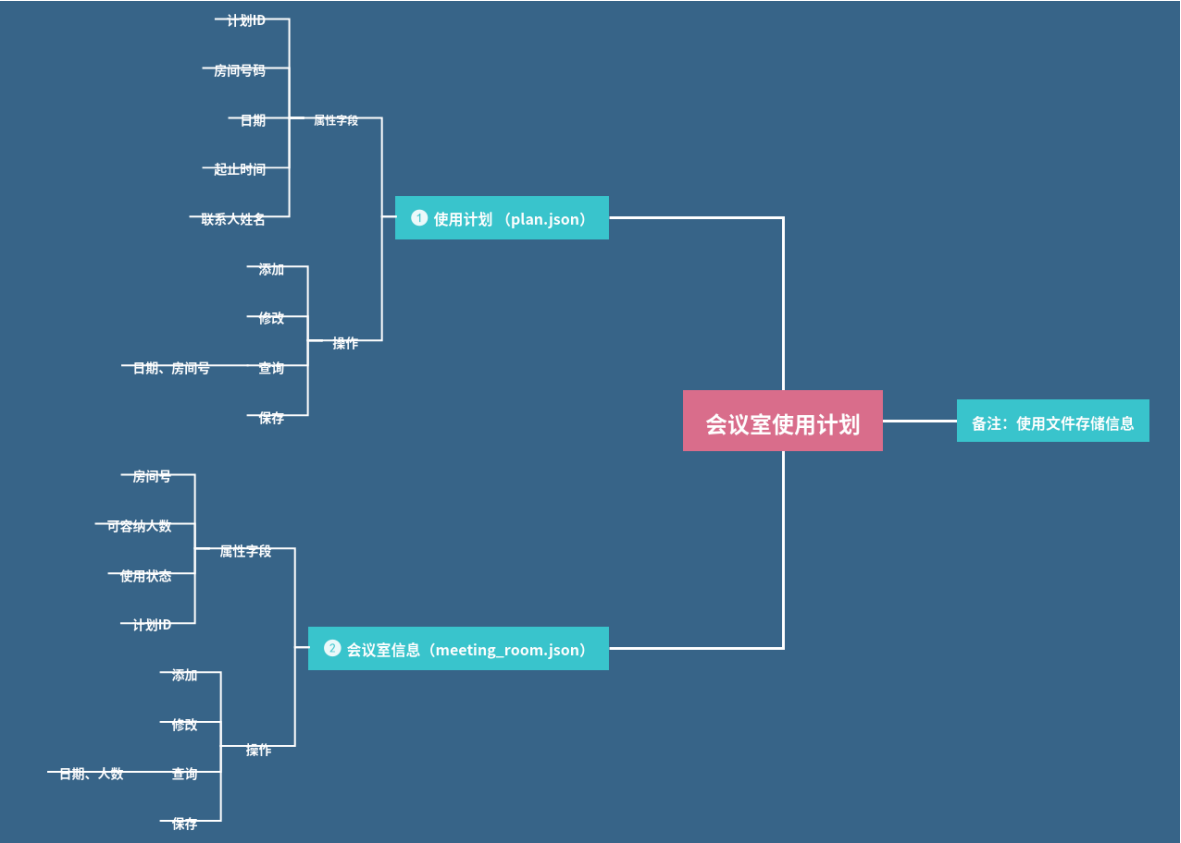
def int_input():
    """ 整数输入错误获取 """
    while True:
        try:
            select_id = int(input('输入序号: '))
            return select_id

```

```
except ValueError:
    print('Err:序号输入有误, 请重新输入! ')

def is_empty(list):
    """ 判断列表是否为空 """
    return False if list else True
```

2. 设计说明



3. 测试结果

3.1 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
101 300 1 0
102 50 0 -1
103 30 1 1
104 100 0 -1
201 1000 0 -1
202 50 1 2
203 200 1 3
204 100 0 -1
301 50 0 -1
302 100 0 -1
303 100 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 2019-10-23 9:00 12:00 张三
1 103 2019-10-10 13:00 16:00 李四
2 202 2019-10-10 17:00 20:00 李四
3 203 2019-10-11 12:00 13:00 11
```

3.2 测试结果

3.1 主菜单

```
讨论室使用计划管理系统

-- 1.使用计划管理
-- 2.讨论室信息管理
-- 0.退出
-----
输入序号: 
```

3.2 讨论室使用管理

讨论室使用计划信息管理

- 1.信息添加
- 2.信息修改
- 3.信息检索
- 4.全部信息查看
- 0.返回

输入序号：1
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）
>> 102 2019-10-29 9:00 12:00 LL
添加成功！
输入序号：2
输入修改ID：
>> 5
信息不存在！
输入序号：2
输入修改ID：
>> 4
输入修改后的信息：（日期、起止时间）
>> 日期:2019-10-29
>> 起始时间:12:00
>> 终止时间:14:00
修改成功!!!
输入序号：3
输入查询日期和房间号：
>> 日期:2019-10-23
>> 房间号:101

全部信息：

ID	房间号	计划日期	起止时间	联系人
0	101	2019-10-23	9:00-12:00	张三

输入序号：4

全部信息：

ID	房间号	计划日期	起止时间	联系人
0	101	2019-10-23	9:00-12:00	张三
1	103	2019-10-10	13:00-16:00	李四
2	202	2019-10-10	17:00-20:00	李四
3	203	2019-10-11	12:00-13:00	LL
4	102	2019-10-29	12:00-14:00	LL

3.3 讨论室信息管理

```
讨论室信息管理

-- 1.信息添加
-- 2.信息修改
-- 3.信息检索
-- 4.全部信息查看
-- 0.返回
-----
输入序号：1
输入会议室信息：格式（房间号 可容纳人数）
>> 308 200
添加成功!!!
输入序号：2
输入修改房间号：
>> 308
输入修改后的信息：（可容纳人数）
>> 100
修改成功!!!
输入序号：3
输入查询日期和人数：
>> 日期(YY-MM-DD): 2019-10-10
>> 人数：20

全部信息：

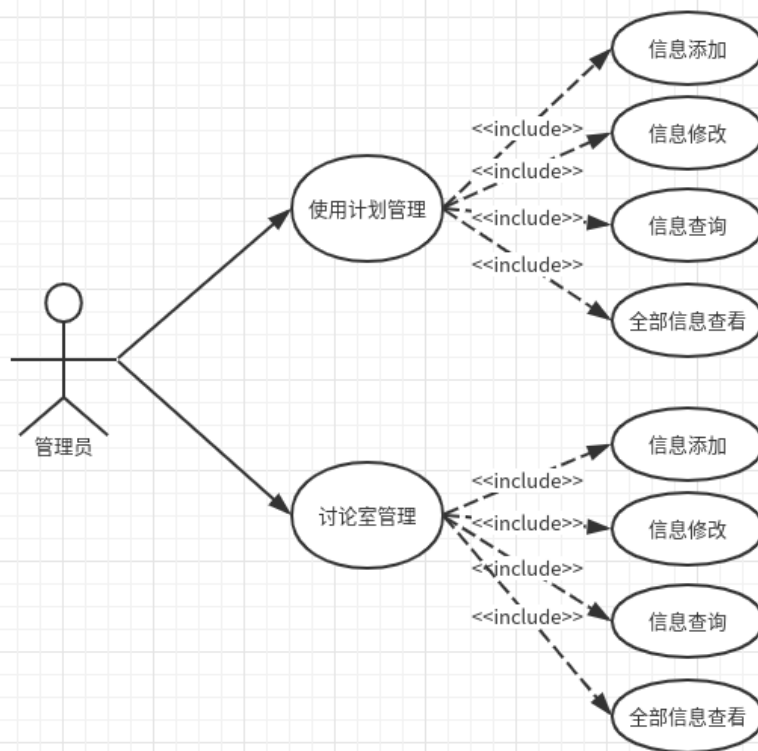
    房间号      可容纳人数      使用状态      计划ID
-----
    101          300          已预约          0
    102           50          空闲
    104          100          空闲
    201         1000          空闲
    203          200          已预约          3
    204          100          空闲
    301           50          空闲
    302          100          空闲
    303          100          空闲
    305          200          空闲
    307          200          空闲
    308          100          空闲
-----

输入序号：4

全部信息：

    房间号      可容纳人数      使用状态      计划ID
-----
    101          300          已预约          0
    102           50          空闲
    103           30          已预约          1
    104          100          空闲
    201         1000          空闲
    202           50          已预约          2
    203          200          已预约          3
    204          100          空闲
    301           50          空闲
    302          100          空闲
    303          100          空闲
    305          200          空闲
    307          200          空闲
    308          100          空闲
-----
```

4. 用例图



5. 类图

