

# 第五周周报

## 本周工作

完成基本数据类型的定义,包括:建筑物,景区,日记等,确定了每个基本类型的成员变量

代码文件: [DataType.py](#)

成员变量代码如下:

```

from typing import Tuple,Set,List
import time
...

## @type Location 位置
## @type Name 名称
## @type Function 功能
## @type Text 文本
...

Location = Tuple[float,float]
Function = Set[str]
Name = str
Text = List[str]

...

## @brief 建筑类
## @param buildingLocation 位置
## @param buildingName 名称
## @param buildingFunction 功能
...

class Building:
    buildingLocation:Location = tuple()
    buildingName:Name = ""
    buildingFunction:Function = set()
    ...

    ## @brief 初始化函数和修改以及获取成员变量的函数
    ...

    def __init__(self,location,function,name):
        self.__buildingLocation = location
        self.__buildingName = name
        self.__buildingFunction = function

    ...

## @brief 评论类
## @param commentOwner 评论者
## @param commentText 评论内容
## @param commentScore 评论分数
...

class Comment:
    commentOwner:Name = ""
    commentText:Text = list()
    commentScore:int = 0
    commentTime:Tuple = time.localtime(time.time())

```

```

def __init__(self,owner,text,score,time):
    self.__commentOwner = owner
    self.__commentText = text
    self.__commentScore = score
    self.__commentTime = time

...

## @brief 日记类
## @param journalName 日记名称
## @param journalGrade 日记评分
## @param journalContent 日记内容
## @param journalComment 日记评论
## @param journalDate 日记日期
...

class journal:
    journalName:Name = ""
    journalGrade:float = 0
    journalContent:Text = list()
    journalComment:Set[Comment]
    journalDate:Tuple = time.localtime(time.time())

    def __init__(self,name,grade,content,comment,date):
        self.__journalName = name
        self.__journalGrade = grade
        self.__journalContent = content
        self.__journalComment = comment
        self.__journalDate = date

```

实现推荐模块的排序算法,实现根据传入的对象数组和属性值完成以该对象特定属性排序的功能

代码文件: [Recommend.py](#)

```

from DataType import *
import random

...

## @note 排序算法
## @param array 对象列表
## @param key 排序关键字(字符串格式)
## @param mode 排序方式(0:升序,1:降序)
## @return 排序后的对象列表(原列表不变)
...

def sort(array:List,key:str,mode:int = 0):
    if len(array) <= 1:return array
    base = getattr(array[0],key)
    left = [x for x in array[1:] if getattr(x,key) < base]
    right = [x for x in array[1:] if getattr(x,key) >= base]
    if mode == 0:return sort(left,key,mode) + [array[0]] + sort(right,key,mode)
    else:return list(reversed(sort(left,key,mode) + [array[0]] + sort(right,key,mode)))

```

## 下周计划

- 完成推荐模块的开发,包括排序和推荐(协同过滤)
- 完成查询模块的开发