**《Python程序设计》第三次作业（类与文件流）**

1. 请基于1.py的框架定义一个类CampusForum，用于某校园论坛的运营。该类应具有下列属性及方法：
2. 初始化\_\_init\_\_()：包含以下三个子任务：

* 定义属性user\_pwd：记录论坛已注册用户的用户名及密码的字典。其值应从文件user\_pwd.txt中读取。文件的每行代表一位用户的名称及密码，二者由制表符**\t**分隔，如：第一行表示用户**u100903**的密码为**YjbyIu45Nl**。
* 定义属性post\_info：记录论坛内讨论贴各项信息的字典。其值应从文件post\_info.txt中读取。文件的每行是一个帖子的编号、内容、发帖日期、发帖用户、评论（不同项由制表符分隔），如：第一行表示编号**#702**的帖子由用户**u100903**于**2023/3/30**发布，内容为**人大校内或周边哪里有修ipad的地方推荐吗？**，该帖有两条评论：**不知道欸**和**中关村有**。

**注意**：出于隐私考虑，属性user\_pwd和post\_info不可外部调用。

* 呈现讨论贴：即按如下格式在屏幕上打印输出论坛内讨论贴的各项信息，包括编号、内容、发帖用户、发帖日期、评论数：

**#702：人大校内或周边哪里有修ipad的地方推荐吗？  
 (发帖人：u100903 日期：2023/3/30 评论数：2)  
--------------------------------------------------  
#768：上学期心理课成绩还没出是正常的吗？  
 (发帖人：u100482 日期：2023/3/30 评论数：1)  
--------------------------------------------------  
#783：蹲一个电动车  
 (发帖人：u100903 日期：2023/3/25 评论数：0)  
--------------------------------------------------**

**…**

1. 方法watch\_review(post\_id)：供用户（即实例）查看指定编号（post\_id）帖子的评论。注意，该论坛仅允许已注册用户查看评论，因此调用此方法需**外部输入**用户名（user）和密码（pwd）进行身份验证。若用户未通过验证打印输出：**登录失败！无权限查看评论！**，否则按如下格式打印输出相应评论：

**1楼：不知道欸  
2楼：中关村有**

1. 方法delete\_post(post\_id)：供用户（即实例）删除的指定编号的帖子（post\_id）。注意，仅发帖用户可进行删除操作，因此调用该方法也许**外部输入**用户名和密码以进行身份验证。若用户通过验证且为发帖用户则删除对应帖子并打印输出：**删除成功！**，否则打印输出：**您无权限删除该帖！**。

参考答案：

**class** CampusForum:  
 *# a.* **def** \_\_init\_\_(self):  
 *# 子任务1：方法1* **with** open(**"./data/user\_pwd.txt"**, encoding=**"utf8"**) **as** f:  
 self.\_\_user\_pwd = dict([tuple(line.strip(**"\n"**).split(**"\t"**)) **for** line **in** f])  
 *# 子任务2* self.\_\_post\_info = {}  
 **with** open(**"./data/post\_info.txt"**, encoding=**"utf8"**) **as** f:  
 **for** line **in** f.readlines():  
 item = line.strip(**"\n"**).split(**"\t"**)  
 self.\_\_post\_info[item[0]] = dict(zip([**"内容"**, **"发帖日期"**, **"发帖用户"**, **"评论"**], item[1:]))  
 review\_list = [r **for** r **in** self.\_\_post\_info[item[0]][**"评论"**].strip(**''''"[]'''**).split(**'", "'**) **if** r]  
 self.\_\_post\_info[item[0]][**"评论"**] = review\_list

*# 子任务3：方法1* **for** pid,pinfo **in** self.\_\_post\_info.items():  
 print(**"%s：%s"** % (pid,pinfo[**"内容"**]))  
 print(**"\t(发帖人：%s\t日期：%s\t评论数：%s)"** % (pinfo[**"发帖用户"**],pinfo[**"发帖日期"**],len(pinfo[**"评论"**])))  
 print(**"-"**\*50)

*# 子任务3：方法2*

…

dict1.append(f'{post\_id}：{post\_s["内容"]}\n (发帖人：{post\_s["发帖用户"]}\t日期：{post\_s["发帖日期"]}\t评论数：{len(post\_s["评论"])})\n{"-" \* 50}')

print('\n'.join(dict1))

…  
  
 *# b.* **def** watch\_review(self,post\_id):  
 user = input(**"请输入用户名："**)  
 pwd = input(**"请输入密码："**)  
  
 **if** (user,pwd) **in** self.\_\_user\_pwd.items():  
 **for** i, review **in** enumerate(self.\_\_post\_info[post\_id][**"评论"**]):  
 print(**"%d楼：%s"** % (i + 1, review))  
 **else**:  
 print(**"登录失败！无权限查看评论！"**)  
 **return**

*# c.* **def** delete\_post(self,post\_id):  
 user = input(**"请输入用户名："**)  
 pwd = input(**"请输入密码："**)  
  
 **if** self.\_\_post\_info[post\_id][**"发帖用户"**] == user **and** pwd == self.\_\_user\_pwd[user]:  
 self.\_\_post\_info.pop(post\_id)  
 print(**"删除成功！"**)  
 **else**:  
 print(**"您无权限删除该帖！"**)  
 **return**

1. movie\_rating.txt中是某流媒体播放平台上部分电影的用户评分信息，包括：电影编号、用户编号、评分和评分时间。各项信息按如下格式记录：

**9253:**

**2016504,4,2004-06-11**

**172720,3,2002-08-17**

**…**

**9307:**

**1945809,2,2003-01-2**

请计算movie\_rating.txt中每部电影的用户平均评分（保留2位小数），并将结果写入average\_rating.txt中。average\_rating.txt的每一行记录一部电影的编号和用户平均评分，两项之间由制表符分隔。

参考答案：

*# 方法1*

**with** open(**"./data/movie\_rating.txt"**,encoding=**"utf8"**) **as** f:  
 lines = f.readlines()  
  
current\_movie,movie\_rating = **None**,{}  
**for** line **in** lines:  
 **if** line.endswith(**":\n"**):  
 current\_movie = line.rstrip(**":\n"**)  
 movie\_rating[current\_movie] = []  
  
 **else**: movie\_rating[current\_movie].append(int(line.rstrip(**"\n"**).split(**","**)[1]))  
  
**with** open(**"./data/average\_rating.txt"**,**"w"**,encoding=**"utf8"**) **as** f:  
 **for** movie,ratings **in** movie\_rating.items():  
 f.write(**"%s\t%.2f\n"** % (movie,sum(ratings)/len(ratings)))

*方法2*

…

movie\_ratings[movie\_id]['total\_rating'] += int(rating)

movie\_ratings[movie\_id]['num\_rating'] += 1

…

avg\_rating = round(movie\_ratings[movie\_id]['total\_rating'] / movie\_ratings[movie\_id]['num\_rating'], 2)

…

1. 利用某统计分析工具对某样本的一些变量（rating、duration、n\_qna和n\_file）进行描述性统计的结果存储在variable\_summary.txt中。其中，每个变量的描述性统计结果按如下格式记录，包含变量名，以及该变量在各统计量上的结果： **rating  
   ---------------------------------------------------  
   Obs 99,441  
   Sum of Wgt. 99,441  
     
   Mean 4.6065  
   Std. Dev. .7600252  
     
   Variance .5776384  
   Skewness -2.393966  
   Kurtosis 9.503871**

请读取variable\_summary.txt中各变量在每个统计量上的结果，将结果整合为如下格式的表格并打印输出：

**|Obs |Sum of Wgt. |Mean |Std. Dev. |Variance |Skewness |Kurtosis   
-------------------------------------------------------------------------------------------------------  
rating |99,441 |99,441 |4.6065 |.7600252 |.5776384 |-2.393966 |9.503871   
duration |99,441 |99,441 |84.70259 |40.24217 |1619.432 |1.663295 |7.713831   
n\_qna |99,441 |99,441 |33.45144 |35.2412 |1241.942 |2.473053 |12.87551   
n\_file |99,441 |99,441 |20.58077 |24.89301 |619.6619 |3.424487 |25.96415**

注意：在该表中，每行的单元格之间由“**|**”分隔，每个单元格的长度为12个字符。

参考答案：

*# 方法1*

**with** open(**"./data/variable\_summary.txt"**,encoding=**"utf8"**) **as** f:  
 summaries = f.read().split(**"=============================="**)[1:]  
  
summary\_info = {}  
**for** summary **in** summaries:  
 items = [item.split(**" "**) **for** item **in** summary.strip(**"\n "**).split(**"\n"**) **if** item **and not** item.startswith(**"-"**)]  
 summary\_info[items[0][0]] = {**" "**.join(item[:-1]).strip(**" "**):item[-1] **for** item **in** items[1:]}  
  
headers,width = **None**,12  
**for** var,stats **in** summary\_info.items():  
 **if not** headers:  
 headers = **" "**\*width+**""**.join([**"|"**+stat+**" "**\*(width-len(stat)) **for** stat **in** stats.keys()])  
 print(headers)  
 print(**"-"**\*len(headers))  
 print(var+**" "**\*(width-len(var))+**""**.join([**"|"**+value+**" "**\*(width-len(value)) **for** value **in** stats.values()]))

*方法2*

…

for i in range(0, len(lines), 12):

var\_name = lines[i+1].strip()

obs = lines[i+3].split()[-1]

sum\_wgt = lines[i+4].split()[-1]

mean = lines[i+6].split()[-1]

std\_dev = lines[i+7].split()[-1]

variance = lines[i+9].split()[-1]

skewness = lines[i+10].split()[-1]

kurtosis = lines[i+11].split()[-1]

…

*方法3*

…

data=[

[' '\*12,'Obs','Sum of Wgt.','Mean','Std.Dev.','Variance','Skewness','Kurtosis'],

['-------------------------------------------------------------------------------------------------------'],

['rating',obs\_[0],wgt\_[0],mean\_[0],std\_[0],va\_[0],sk[0],ku\_[0]],

['duration',obs\_[1],wgt\_[1],mean\_[1],std\_[1],va\_[1],sk[1],ku\_[1]],

['n\_qna',obs\_[2],wgt\_[2],mean\_[2],std\_[2],va\_[2],sk[2],ku\_[2]],

['n\_file',obs\_[3],wgt\_[3],mean\_[3],std\_[3],va\_[3],sk[3],ku\_[3]]

]

for row in data:

line = '|'.join('{:<12}'.format(cell) for cell in row)

print(line)

*方法4*

…

for i in range(len(name)):

line = format(name[i],'12')

for record in list:

line += '|' + format(record[i],'12')

…

*方法5*

…

for i in range(len(variable\_name)):

value =[]

value.append(variable\_name[i])

for j in result.values():

value.append(j[i])

print("%-12s|%-12s|%-12s|%-12s|%-12s|%-12s|%-12s|%-12s"%(tuple(value)))

*方法6*

…

print("{:<12}|{:<12}|{:<12}|{:<12}|{:<12}|{:<12}|{:<12}|{:<12}".format(variable, obs, sum\_of\_wgt, mean, std\_dev, variance, skewness, kurtosis))

*方法7*

…

row = f"{name.ljust(12)}|{obs.ljust(12)}|{sum\_of\_wgt.ljust(12)}|{mean.ljust(12)}|{std\_dev.ljust(12)}|{variance.ljust(12)}|{skewness.ljust(12)}|{kurtosis.ljust(12)}\n"

…

*方法8【此方法有一些错误，需要把{:^12s}换成{:<12s}】*

…

for var\_name in data:

obs, sum\_of\_wgt, mean, std\_dev, variance, skewness, kurtosis = data[var\_name]

row = '|'.join(['{:^12s}']\*8)

print(row.format(var\_name, obs, sum\_of\_wgt, mean, std\_dev, variance, skewness, kurtosis))

**作业要求**：

1. **作业提交截止时间**：2023年4月21日晚24:00前（迟交会酌情扣分）
2. **作业提交方式**：发送至助教邮箱。发送邮件标题命名为：Python第三次作业+姓名+学号。
3. **文件命名方式**：每道题目的程序单独建一个python文件（.py或.ipynb），例如本次作业建3个文件，分别命名为1.py，2.py，3.py或1.ipynb，2.ipynb，3.ipynb。

本次作业统一打包成一个压缩文件(.zip或.rar)，以邮件附件的形式提交给助教，压缩文件命名方式为“Python第三次作业+姓名+学号.zip，或者“Python第三次作业+姓名+学号.rar”