《数据读取与处理》作业说明

《程序设计实践》作业 1

2021 年 06 月

一、作业内容

在给定的代码文件 `util.py` 中填充指定方法的具体逻辑，实现功能。

Python 作为常用的脚本语言，能够方便地对数据进行处理，本次作业要求实现一个数据读取的方法功能逻辑，考察对 Python 语言的掌握和一些基本语法的使用。

本次作业要求实现一个方法，能够从给定的文件路径中，读取一份按照英文逗号分隔的 `.csv` 格式文件，并将文件中包含的数据根据输入参数处理后按照要求的格式返回。方法参数和返回值已经定义好，同学们需要完成的内容是填充其中的逻辑并正确返回数据。

二、任务说明

现实场景中对于数据的处理，可能包含读取、筛选、排序、分页等操作。本次作业要求根据方法中相关参数的值，正确实现读取、筛选、排序、分页和输出的逻辑。

2.1 数据读取

在输入的 `.csv` 格式的文件内容中，第一行是列名称，按照英文逗号分隔；第二行至最后一行每行是一条数据，数据按照英文逗号分隔的每一列代表该行数据中对应相同列索引的属性的值。

比如以下内容：

```
1. id,name,age,remark
```

```
1,leo,33,excellent
2,cri,35,wonderful
```

第一行内容声明了该文件包含数据的属性集合，共有 4 个属性 `id`、`name`、`age` 和 `remark`；第二行和第三行分别表示两条数据，它们分别表示一位 {年龄 33 岁的 leo} 和一位 {年龄 35 岁的 cri} 的基本信息。

在读取文件内容后，每行数据应该被保存为一个字典，这个字典的键的集合为属性名称集合，每个键的值为对应该属性的值。比如对于第二行数据，可以得到这样一个字典（用 JSON 格式表示）：

```
{
    "id": 1,
    "name": "leo",
    "age": 33,
    "remark": "excellent"
}
```

2.2 分页

在实际的数据交互过程中，当全量数据较大时，通常会以分页的形式返回全量数据的一部分。

分页逻辑通常通过每页数量和页数索引来控制。举例而言，假定有 100 条数据分别为整数 1 至整数 100，设定每页数量为 15，第 1 页应该返回`[1,2,…,15]`，第 2 页应该返回`[16,17,…,30]`。 p.s.最后一页如果不全也应显示

2.3 排序

在实际使用场景中，人们往往会对数据某个特定的性质感兴趣，因而希望获得按照某种规则进行排序。最常见的规则是按照数据的某个属性进行排序，比如对于一个班级的同学，按照其年龄大小进行排序，可以快速获取到最年轻同学的

相关信息。

排序逻辑可以很复杂，在本次作业中，只考虑按照某个特定属性值进行排序的策略。对于类型为数值的数据排序逻辑即为数值大小，对于类型为字符串类型的数据顺序即为常规的字符串顺序大小（在本次作业中只会包含这两种情况）。顺序类型分为“升序”和“降序”两种。

2.4 筛选

在数据量较大的情况下，往往使用筛选条件快速缩小数据的范围。比如按照年龄筛选大于 60 岁的人群，可以将不需要分析的青少年数据排除在外。

在本次作业中，需要根据给定参数，按照参数代表的筛选条件对数据进行筛选，返回的数据中应该只包含符合筛选条件的数据集合。在本次作业中，如果筛选条件内容为数值类型，则通过相等条件进行筛选；如果筛选条件内容为**字符串**类型，则通过**模糊匹配**的条件进行筛选（“模糊匹配”指筛选内容是对应属性内容的子字符串）。

2.5 参数说明

在本次作业中，要求填写给定方法的功能逻辑。方法的参数已经给定，**不能对参数的顺序、含义和数据类型进行修改**。具体参数的含义如下：

- 参数 `input_file_path` 表示输入文件的路径；当文件不存在时，直接返回数据为空的返回类型对象（后简称为“空数据”）。
- 参数 `page_size` 和 `page` 表示分页操作，`page_size` 表示每一页的数据数量，`page` 表示页数（从 1 开始）；注意当 `page_size` 的值为 0 时，表示**不进行分页**，应该返回全量数据；当 `page_size` 为负数时，返回空数据；根据 `page_size` 的取值情况，若 `page` 不合法（page > total_data_number / page_size 或 page < 1）也返回空数据。
- 参数 `sort_key` 和 `sort_order` 表示排序参数，应该按照指定 `sort_key` 所标识列的属性值进行排序，`sort_order` 的值可能是 `asc` 或 `desc`，

分别表示升序或降序；当 `sort_key` 标识的属性不合法时，返回空数据，当 `sort_order` 的值不合法时，返回空数据。

- 参数 `filter_dict` 是一个字典，字典的键和值分别表示属性名称和对应的筛选条件内容。注意，`filter_dict` 中可能出现不在属性名称集合范围内的键，这些筛选条件应该被忽略；`filter_dict` 中也可能出现某个属性名称的键但其值为 None，这个筛选条件也应该被忽略；当 `filter_dict` 自身是一个 None 的时候，表示没有任何筛选条件。注意每次调用函数时，筛选条件可能会有多个。

```
注意最后的utf-8编码
file=open("data.csv","r",encoding='utf-8')
否则会产生乱码
```

2.6 返回值说明

返回值的类型在给定代码中已经定义，为 `ResponseData` 类型，包含了待返回的数据列表及数据总数。

`data_list` 是一个列表，列表中的每个元素应该是一个字典，表示从输入文件中读取的一条数据，该字典需要包含表示各个属性的键及对应的属性值。

`total` 是一个整型数据，表示数据总数。`total` 的值与最终返回的 `data_list` 列表的长度一致。

2.7 其它要求及提示

- 方法应该有正确的异常处理机制，遇到各类异常状况时（比如当输入的文件不存在、输入参数的类型与预期不符）时，应返回空数据（等同于文件中没有任何数据的情况），**方法在任何输入组合下都不应该抛出异常，**在遇到无法处理的情况时应该返回数据为空的返回类型对象；
- 对于文件内容中内容为空或按照逗号分割后列数小于属性数量的行，应该直接忽略该行数据；
- 每次调用方法时会做多个操作，处理顺序为：读取数据-分页-排序-筛选-返回最终结果，每一步骤的输入是上一步骤的输出。
- 代码、输入文件均为 UTF-8 格式，注意打开文件时的编码格式；（否则输

出结果中某些字符会与预期结果不符）

- 对于数值类型需要解析，本次作业中可能出现的类型为字符串、整型和浮点型，但由于所有数据均以文本格式读取，在默认状态下都是字符串，需要进行对应的数据解析和数据类型转换（否则在排序时会出现问题，例如“10”<“5”但 10 > 5）；
- 建议参考代码示例中的 `example.py` 文件并运行，确保 `fetch_data` 方法能够被外部方法正确调通；

三、提交格式

本次作业最终只需要提交完成后的 `util.py` 文件，按照以下格式组织提交的文件：

- [学号]_[姓名]_hw1
  - util.py

即创建一个名称为 `[学号]_[姓名]_hw1` 的文件夹，将 `util.py` 文件放在该文件夹中。比如对于学号为 `2030010001`，姓名为 `张三` 的同学，文件夹的名称应该为 `2030010001_张三_hw1`（**注意使用下划线而非其它字符拼接学号和姓名**）。将这个文件夹打包成 `.zip` 压缩包后，上传到网络学堂。

作业文件中提供了示例版本的作业提交格式。注意，文件夹中的学号和姓名将直接用于统计作业分数，**未按照要求格式命名的作业将无法被代码正确解析，会酌情扣分。**

四、评测方式

本次作业的输入输出格式较为固定，将采用自动化的评测方式。助教将准备若干测试样例（包含不同的输入文件、输入参数、预期返回值），使用代码调用同学们提交的代码文件，执行方法，判断方法返回的结果是否与预期相符。

评价指标	比例	备注
预置测试样例集合	80%	代码自动评测
提交格式满足要求	10%	代码自动按照格式解析
代码风格与可读性	10%	人工批阅

五、其它

注意事项：

- 批改环境将使用 Python3.8 环境（建议使用 Python3.7 或 3.8 或 3.9 版本完成作业），本次作业理论上不需要安装默认库之外的第三方库。
- 提交的文件夹和文件名称按照前文要求组织
- 鼓励可以将相关逻辑拆分到子方法中，提高代码可读性

希望同学们认真按照作业说明和要求完成作业的提交，避免由于提交格式原因丢失分数。如果有问题或疑问及时与助教（郝天翔，htx20@mails.tsinghua.edu.cn）联系，谢谢！