Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №3**

Дисциплина: «Разработка профессиональных приложений»

Файлы и словари

Вариант №11

Выполнил:

студент группы ИВТАПбд-21

Машковцев Н.М

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту**

Необходимо реализовать 11 вариант задания, а именно: Посты: №, ник автора, текст поста, количество лайков. Отсортировать по количеству лайков, и количество набранного текста.

**Требования к программе:**

1.    Пусть дана некоторая директория (папка). Посчитайте количество файлов в данной директории (папке) и выведите на экран.

2.    Пусть   дан   файл   data.csv,  в   котором   содержится информация в соответствии с вариантом:

Считайте информацию из файла в соответствующую структуру (словарь):

2.1. Выведите информацию об объектах, отсортировав их по одному полю (строковому).

2.2. Выведите информацию об объектах, отсортировав их по одному полю (числовому).

2.3. Выведите информацию, соответствующую какому-либо критерию (например, для студентов - тех, у кого возраст больше какого-либо значения)

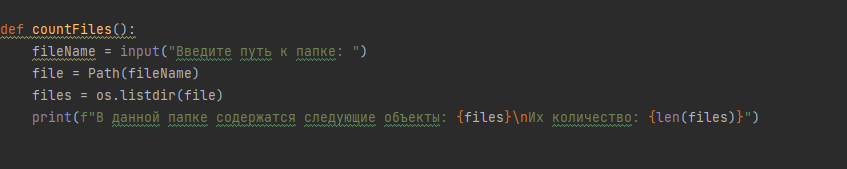
3.    Добавьте к программе возможность сохранения новых данных обратно в файл.

**Описание реализации**

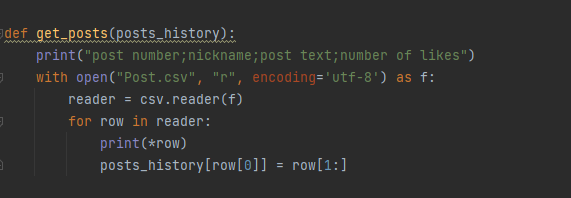
Созданы 6 функций:

1. Функция вывода кол-ва файлов в заданной директории

В которой вы считаем кол-во файлов в заданной папке при помощи класса def countFiles, библиотеки import os.

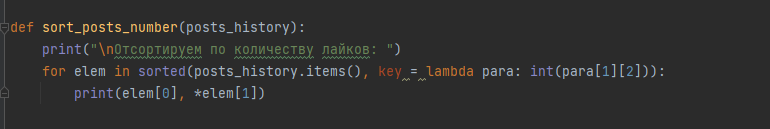


2. Функция чтения данных из файла и занесение их в словарь

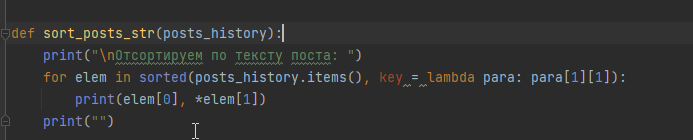


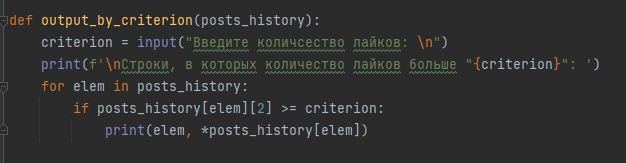
Происходит открытие файла формата .csv и считывание, дальше с помощью цикла for пробегаемся по данным и записываем их в словарь

3. Функция сортировки по целочисленному значению:



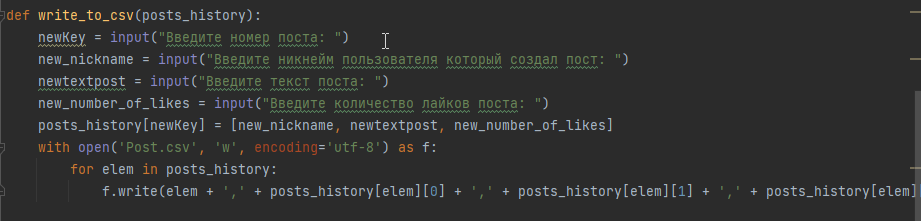
С помощью цикла for пробегаемся по элементам словаря. Сортировка происходит благодаря методу sorted() в параметрах которого указаны элементы словаря и функция lambda.

4. Сортировка по строковому полю  


5. Сортировка по критерию 

С помощью цикла for пробегаемся по словарю с данными и выводим строки содержащие нужное нам название товара.

6. Запись в файл

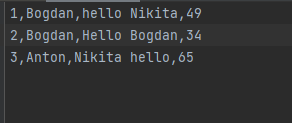


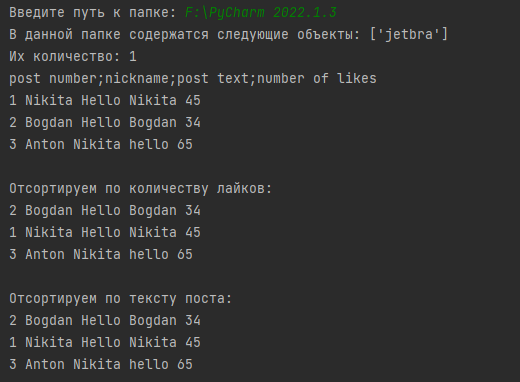
В начале функции происходит считывание данных с клавиатуры, а именно: ключ, дата, цена, название товара. Данные записываются в словарь. После этого открываем файл на чтение и благодаря циклу for записываем в него словарь, содержавший предыдущие и новые данные. После этого закрываем файл.

**Описание возникших затруднений**

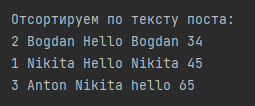
Затруднения возникли с сортировкой словаря по строковому и числовому полям так как в словаре в качестве значения ключа был список. Немного изучив данную ситуацию, проблема была решена. Других затруднений лабораторная работа не вызвала

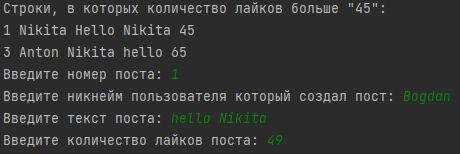
**Результаты работы программы**

 **1 рис. Исходный .csv файл**



**3 рис. Сортировка по количеству лайков**

 **4 рис. Сортировка по тексту поста**

 **5 рис. Сортировка по количеству лайков больше определенного значения**

**Код программы**

import os  
from pathlib import Path  
import csv  
  
def get\_posts(posts\_history):  
 print("post number;nickname;post text;number of likes")  
 with open("Post.csv", "r", encoding='utf-8') as f:  
 reader = csv.reader(f)  
 for row in reader:  
 print(\*row)  
 posts\_history[row[0]] = row[1:]  
  
  
def sort\_posts\_number(posts\_history):  
 print("\nОтсортируем по количеству лайков: ")  
 for elem in sorted(posts\_history.items(), key = lambda para: int(para[1][2])):  
 print(elem[0], \*elem[1])  
  
  
def sort\_posts\_str(posts\_history):  
 print("\nОтсортируем по тексту поста: ")  
 for elem in sorted(posts\_history.items(), key = lambda para: para[1][1]):  
 print(elem[0], \*elem[1])  
 print("")  
  
  
def output\_by\_criterion(posts\_history):  
 criterion = input("Введите количсество лайков: \n")  
 print(f'\nСтроки, в которых количество лайков больше "{criterion}": ')  
 for elem in posts\_history:  
 if posts\_history[elem][2] >= criterion:  
 print(elem, \*posts\_history[elem])  
  
  
def write\_to\_csv(posts\_history):  
 newKey = input("Введите номер поста: ")  
 new\_nickname = input("Введите никнейм пользователя который создал пост: ")  
 newtextpost = input("Введите текст поста: ")  
 new\_number\_of\_likes = input("Введите количество лайков поста: ")  
 posts\_history[newKey] = [new\_nickname, newtextpost, new\_number\_of\_likes]  
 with open('Post.csv', 'w', encoding='utf-8') as f:  
 for elem in posts\_history:  
 f.write(elem + ',' + posts\_history[elem][0] + ',' + posts\_history[elem][1] + ',' + posts\_history[elem][2] + '\n')  
  
  
def main():  
 # countFiles()  
 posts\_history = {}  
 get\_posts(posts\_history)  
 sort\_posts\_number(posts\_history)  
 sort\_posts\_str(posts\_history)  
 output\_by\_criterion(posts\_history)  
 write\_to\_csv(posts\_history)