Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Разработка профессиональных приложений

(название дисциплины)

**Отчёт по лабораторной работе №3**

«Файлы и словари»

(название (тема) работы)

**Вариант №13**

Выполнил:

студент группы ИВТАПбд-21

Молчанов А. В.

Проверил(а):

преподаватель кафедры «ВТ»

Исхаков И. И.

Ульяновск

2023

**Постановка задачи**

1. Дана некоторая директория (папка). Необходимо посчитать количество файлов в данной директории (папке) и вывести на экран;

2. Дан файл data.csv, в котором содержится следующая информация в соответствии с вариантом: счётчик посетителей магазина: номер посетителя, дата и время, признак (bool) входа или выхода, пол.

Необходимо:

* вывести информацию об объектах, отсортировав их по одному полю (строковому);
* вывести информацию об объектах, отсортировав их по одному полю (числовому);
* вывести информацию, соответствующую какому-либо критерию (пользователи, которые находятся в магазине на данный момент).

Также, программа должна сохранять новые данные обратно в .csv файл.

**Описание реализации**

В соответствии с требованиями к программе, сначала была реализована первая её часть, а именно — подсчёт количества файлов в заданной директории (папке). Для этого использовалась вспомогательная библиотека Path, которая позволяет манипулировать путями файловых систем в любой операционной системе и её методы iterdir и rglob.

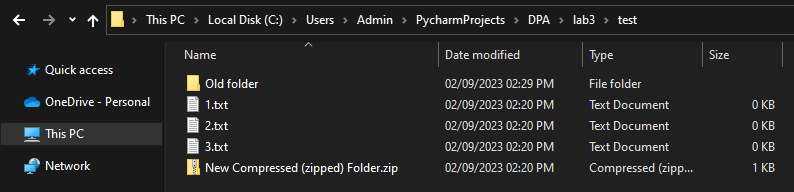
После этого пошла работа с .csv файлами по требованиям, прописанным выше. Для этого использовалась специальная библиотека импорта и экспорта для электронных таблиц и баз данных csv, а также csv.DictReader и csv.DictWriter для чтения и записи файлов соответственно. Для сортировки списков использовались методы sorted и filter, а также лямбда-функции с условиями сортировки.

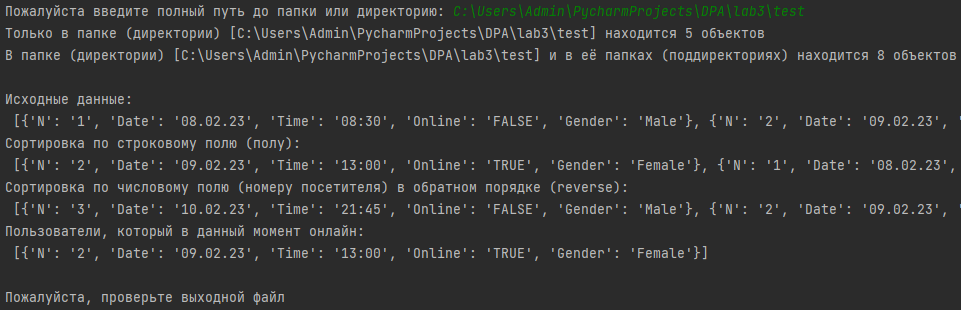
В конце было реализовано сохранение новых данных в выходной файл output.csv.

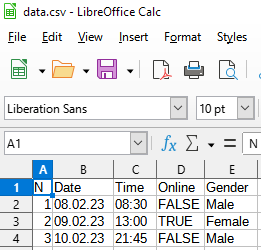
**Исходный код реализации**

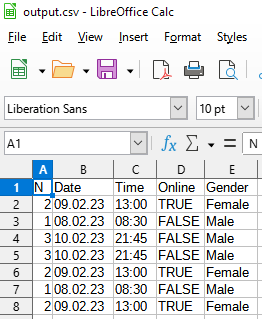
import csv # Библиотека импорта и экспорта для электронных таблиц и баз данных  
from pathlib import Path # Библиотека позволяет манипулировать путями файловых систем в любой операционной системе  
  
  
path = input("Пожалуйста введите полный путь до папки или директорию: ")  
directory = Path(path) # Получаем объект пути до указанной папки или директории  
  
if not directory.is\_dir(): # Если полученный объект действительно является корректной директорией  
 print(ValueError(f"[{directory}] не существует или не является директорией!"))  
else:  
 # Метод iterdir создаёт итератор, который случайным образом перечисляет файлы, не включая подпапки и поддиректории  
 print(f"Только в папке (директории) [{path}] находится {len(list(directory.iterdir()))} объектов")  
 # Метод rglob создаёт итератор, который случайным образом перечисляет файлы, включая подпапки и поддиректории  
 print(f"В папке (директории) [{path}] и в её папках (поддиректориях) находится {len(list(directory.rglob('\*')))} объектов")  
  
print()  
  
fI = open("C:\\Users\\Admin\\PycharmProjects\\DPA\\lab3\\data.csv", 'r') # Открываем файл для чтения (read)  
# Класс DictReader модуля csv создаёт объект, который работает как обычный reader(), но отображает информацию о каждой строке в качестве словаря dict  
reader = csv.DictReader(fI, fieldnames = None, restkey = None, restval = None, dialect = "excel")  
  
l = list(reader) # Преобразуем его в список словарей  
  
print("Исходные данные:\n", l)  
  
# Сортируем список при помощи лямбда-функции по столбцу с полом из таблицы  
sSL = sorted(l, key = lambda d: str(d['Gender']))  
# Сортируем список при помощи лямбда-функции по столбцу с номером клиента из таблицы и переворачиваем список (reverse)  
nSL = sorted(l, key = lambda d: str(d['N']), reverse = True)  
  
print("Сортировка по строковому полю (полу):\n", sSL)  
print("Сортировка по числовому полю (номеру посетителя) в обратном порядке (reverse):\n", nSL)  
  
newL = []  
  
"""  
  
# Вывод информации по времени  
newL = list(filter(lambda x: datetime.datetime.strptime(x['Time'], '%H:%M').hour > 12, l))  
print(newL)  
  
"""  
  
# Отбор посетителей по онлайну (только те, кто в данный момент находится в магазине)  
newL = list(filter(lambda x: x['Online'] == 'TRUE', l))  
print("Пользователи, который в данный момент онлайн:\n", newL)  
  
# Создаём файл для записи  
with open("C:\\Users\\Admin\\PycharmProjects\\DPA\\lab3\\output.csv", 'w', newline = '') as csvfile:  
 writer = csv.DictWriter(csvfile, fieldnames = reader.fieldnames) # Класс writer служит для записи данных в файл  
 writer.writeheader() # Записываем заголовок таблицы  
  
 for row in sSL:  
 # Метод writerow записывает параметр строки row в файл  
 writer.writerow(row)  
  
 for row in nSL:  
 writer.writerow(row)  
  
 for row in newL:  
 writer.writerow(row)  
  
print()  
print("Пожалуйста, проверьте выходной файл")

**Пример работы программы**









**Описание возникших затруднений**

Основные затруднения при выполнении данной лабораторной работой были связаны с первичным знакомством с форматом .csv и Python-библиотекой csv для импорта и экспорта электронных таблиц и баз данных.

Также, значительные трудности возникли с сортировкой списков словарей через лямбда-функции, так как до этого они мной не применялись.

**Описание альтернативных способов решения**

Первый вариант задания данной лабораторной работы я пытался сделать через обычный .txt файл. Из-за этого в самом начале возникло очень много нюансов по сортировке словарей и списков, заполненных таким образом, а именно с тем, как связать их между собой. Для этого пришлось бы продумать и вручную написать достаточно много вещей, что, по-сути, уже было сделано за нас в библиотеке csv, которой я и решил воспользоваться по совету одногруппника.

Таким образом, альтернативные способы решения у данной задачи есть, но они гораздо более сложные с точки зрения написания кода и его количества по сравнению с реализацией этой же задачи при помощи готовой библиотеки csv.

**Вывод**

В ходе выполнения данной лабораторной работы мной были получены практические навыки работы с .csv и не только файлами на языке программирования Python. Сделано это было при помощи использования готовых библиотек, таких как csv и pathlib.

Также, я научился считывать количество файлов в определённой директории (папке) на своём компьютере, познакомился с новой структурой данных — словарём, а также с их сортировкой при помощи лямбда-функций