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 № 14

Выполнил:

студент группы ИВТАПбд-21

Мухаметзянов Т. А.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту**

Необходимо реализовать 14 вариант задания, а именно: из исходного .csv файла с начальными данными Временная шкала температурных показаний: №, широта, долгота, показание температуры, дата и время.

**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 5 функций:

1. Функция вывода кол-ва файлов в заданной директории

В которой вы считаем кол-во файлов в заданной папке при помощи os.walk (создает кортеж всех файлов в папке и возвращает генератор)

def countFilesInDir(path):  
 print(f'Кол-во файлов в директории {path} = {sum(len(files) for \_, \_, files in os.walk(path))}')

2. Функция Сортировки

def sorting(typeBySort):  
 with open('data.csv', newline='') as file:  
 reader = csv.DictReader(file, delimiter=',')  
 match typeBySort:  
 case '1':  
 sortedlist = sorted(reader, key=lambda x: int(x['number']), reverse=False)  
 case '2':  
 sortedlist = sorted(reader, key=lambda row: datetime.strptime(row['date'], "%Y-%m-%d %H:%M"),  
 reverse=False)  
 case '3':  
 sortedlist = filter(lambda row: int(row['number']) >= 10, reader)  
 case \_:  
 print('Error')

Происходит открытие файла формата .csv и считывание, далее в зависимости от ответа пользователя начинаем сортировку с помощью функции sorted и ключа (номер колонки для сортировки с типом значений в этой колонке)

Происходит сортировка по дате и целочисленным значениям, а также с помощью функции filter выводим только те данные, который подходят под условия.

3. Чтение из файла

def readCsv(path):  
 with open(path, mode='r') as f:  
 reader = csv.DictReader(f)  
 for row in reader:  
 print(row)

Открывается файл с помощью DictReader, что позволяет выводить данные в виде словаря.

4. Запись в новых записей в файл

def writeCsv():  
 field\_names = ['number', 'width', 'longitude', 'temperature', 'date']  
 lines = {'number': int(input("number: ")), 'width': int(input("width: ")), 'longitude': int(input("longitude: ")),  
 "temperature": int(input("temperature: ")), 'date': input("date (format(%Y-%m-%d %H:%M): ")}  
  
 with open('data.csv', 'a+', newline='') as f:  
 writer = DictWriter(f, fieldnames=field\_names)  
 writer.writerow(lines)  
 print('\ndata.csv:\n')  
 f.close()  
 readCsv('data.csv')

есть список названий колонок и словарь с ключами и значениями ключей, который уже вводит пользователь, дальше просто записываем в старый файл новые записи с помощью DictWriter, что позволяет хранить словари в .csv файле.

5. Создание нового файла .csv и запись в него сортировок

with open('final.csv', 'w') as outcsv:  
 # configure writer to write standard csv file  
 writer = csv.writer(outcsv, delimiter=',', quotechar='|', quoting=csv.QUOTE\_MINIMAL, lineterminator='\n')  
 writer.writerow(['number', 'width', 'longitude', 'temperature', 'date'])  
 for item in sortedlist:  
 # Write item to outcsv  
 writer.writerow([item['number'], item['width'], item['longitude'], item['temperature'], item['date']])

Сначала записывается список названий колонок, дальше бежим по нашему словарю и выводим значения каждого ключа по строкам.

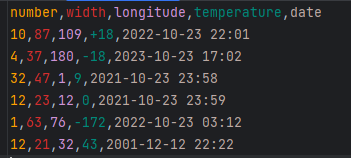
**Описание возникших затруднений**

Работа с чтением и записями в .csv файл была произведена впервые поэтому потребовалось время чтобы разобраться в процессе сохранения и чтения файлов в Python.

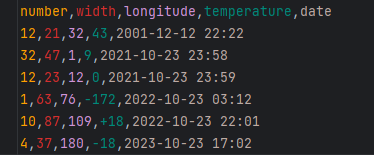
**Результаты работы программы**



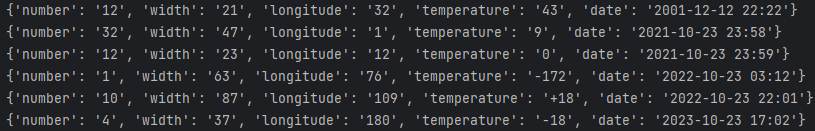
**1 рис. Функция вывода кол-ва файлов в директории**

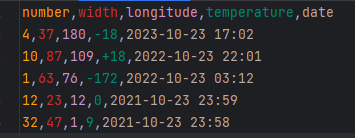


**2 рис. Исходный файл**

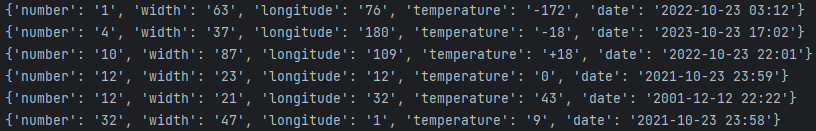


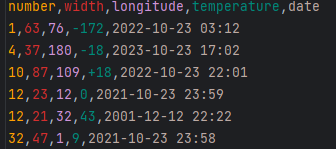
**3 рис. Сортировка по Дате и времени**



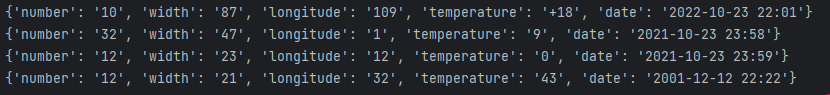


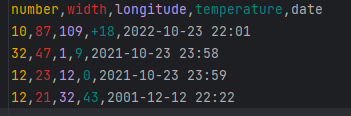
**4 рис. Сортировка по целочисленному значению (1 столбец)**





**5 рис. Сортировка по критерию (номер не менее 10)**





**Код программы**

import csv  
import os  
from datetime import datetime  
from csv import DictWriter  
  
  
def countFilesInDir(path):  
 print(f'Кол-во файлов в директории {path} = {sum(len(files) for \_, \_, files in os.walk(path))}')  
  
  
def readCsv(path):  
 with open(path, mode='r') as f:  
 reader = csv.DictReader(f)  
 for row in reader:  
 print(row)  
  
  
def writeCsv():  
 field\_names = ['number', 'width', 'longitude', 'temperature', 'date']  
 lines = {'number': int(input("number: ")), 'width': int(input("width: ")), 'longitude': int(input("longitude: ")),  
 "temperature": int(input("temperature: ")), 'date': input("date (format(%Y-%m-%d %H:%M): ")}  
  
 with open('data.csv', 'a+', newline='') as f:  
 writer = DictWriter(f, fieldnames=field\_names)  
 writer.writerow(lines)  
 print('\ndata.csv:\n')  
 f.close()  
 readCsv('data.csv')  
  
  
def sorting(typeBySort):  
 with open('data.csv', newline='') as file:  
 reader = csv.DictReader(file, delimiter=',')  
 match typeBySort:  
 case '1':  
 sortedlist = sorted(reader, key=lambda x: int(x['number']), reverse=False)  
 case '2':  
 sortedlist = sorted(reader, key=lambda row: datetime.strptime(row['date'], "%Y-%m-%d %H:%M"),  
 reverse=False)  
 case '3':  
 sortedlist = filter(lambda row: int(row['number']) >= 10, reader)  
 case \_:  
 print('Error')  
 with open('final.csv', 'w') as outcsv:  
 # configure writer to write standard csv file  
 writer = csv.writer(outcsv, delimiter=',', quotechar='|', quoting=csv.QUOTE\_MINIMAL, lineterminator='\n')  
 writer.writerow(['number', 'width', 'longitude', 'temperature', 'date'])  
 for item in sortedlist:  
 # Write item to outcsv  
 writer.writerow([item['number'], item['width'], item['longitude'], item['temperature'], item['date']])  
 readCsv('final.csv')  
  
  
match input("What are you want to do?\n1 - Output count files in dir\n2 - Output a data from .csv\n"  
 "3 - Write a new record in .csv\n4 - Let`s Sort!!!\n"):  
 case '1':  
 countFilesInDir(input("Enter name of directory: "))  
 case '2':  
 readCsv('data.csv')  
 case '3':  
 writeCsv()  
 case '4':  
 sorting(input("Выберите тип сортировки:\n1 - Сортировка по номеру записи (integer)"  
 "\n2 - Сортировка по дате\n3 - сортировка по критерию\n"))