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 № 9

Выполнил:

студент группы ИВТАПбд-21

Карпова А.С.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту**

Необходимо реализовать 9 вариант задания, а именно: из исходного .csv файла с начальными данными справки из деканата: №, дата, ФИО студента, размер стипендии, куда выдается справка.

**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**

Для решения данной задачи была написана программа, которая считывает данные из CSV-файла, и выводит эти данные в виде таблицы. Кроме того, программа содержит функции сортировки данных по номеру стипендии, дате и фамилии студента, а также функцию для сохранения новых данных в файл.

Функция parse\_csv(file) отвечает за парсинг CSV-файла. Она считывает данные из файла и преобразовывает их в словарь got\_data с ключами, соответствующими индексам строк в файле. Значениями в словаре являются другие словари, содержащие информацию о номере стипендии, дате, имени и фамилии студента, а также месте, где выдается справка.

Функции sort\_by\_fio(c), sort\_by\_number(c), sort\_by\_values(c,value) отвечают за сортировку данных. Функция sort\_by\_fio() сортирует данные в словаре по фамилии студента, функция sort\_by\_number() сортирует данные по номеру стипендии, а функция sort\_by\_values() сортирует данные относительно значения номера стипендии больше, чем указанное в качестве параметра.

Функция add(file,c,number,date,fio,stip,kuda) позволяет добавлять новые данные в файл и словарь. Она принимает на вход имя файла, словарь данных, а также значения номера стипендии, даты, имени и фамилии студента, суммы стипендии и места выдачи справки. Функция добавляет новые данные в словарь, сохраняет изменения в файл с использованием функции write().

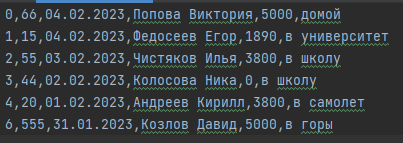
Основной блок программы сначала считывает данные из CSV-файла, используя функцию parse\_csv(), затем выводит их на экран с помощью функции print\_date(). Далее используются функции сортировки данных, а также функция добавления новых данных в файл.

Общее описание реализации: программа состоит из нескольких функций, взаимодействующих с данными внутри программы или с файлом. parse\_csv(), sort\_by\_fio(), sort\_by\_number() и sort\_by\_values() являются функциями обработки данных, а add() реализует действия пользователя с данными. Основная логика программы заключается в считывании данных из файла, выводе таблицы, и в последующих действиях пользователя, выполняющих сортировку данных или добавление новых записей в файл. Результаты данных действий на экране отображаются с использованием функции print\_date().

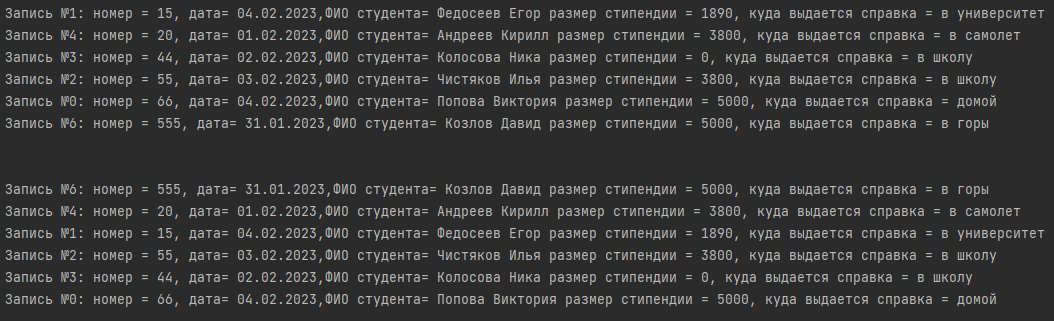
**Описание возникших затруднений**

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**

****

*Рис. 1.1 Исходный файл*

****

*Рис. 1.2 Вывод результата*

**Код программы**

import csv  
import os.path  
import os  
  
lst = os.listdir("D://papka") # your directory path  
number\_files = len(lst)  
print (number\_files)  
print("\n")  
  
def print\_date(dict): #функция на вывод таблицы  
 for a,b in dict.items():  
 print(  
 f"Id{a}: номер = {b['number']}, дата = {b['date']},ФИО студента = {b['fio']} стипендия = {b['stip']}, куда выдается справка = {b['kuda']}")  
  
def parse\_csv(file): # функция парсинга файла  
 got\_data = {}  
 with open(file, "r", encoding='utf-8') as raw\_csv:  
 for line in raw\_csv:  
 (idx, number, date, fio, stip, kuda) = line.replace("\n", "").split(",")  
 got\_data.update({int(idx): {"number": int(number), "date": date, "fio": fio, "stip": stip, "kuda": kuda}})  
 return got\_data  
  
def sort\_by\_fio(c) -> dict: # функция сортировки по месту выдачи справки  
 return dict(sorted(c.items(), key=lambda f: f[1]["fio"]))  
  
def sort\_by\_number(c) -> dict: # функция сортировки по номеру  
 return dict(sorted(c.items(), key=lambda f: f[1]["number"]))  
  
def sort\_by\_values(c,value)->dict: #функция сортировки по номеру больше какого-либо значения  
 return dict((a, b) for a, b in c.items() if b["number"] > value)  
  
data = parse\_csv('D:\\papka\\data.csv.txt') #путь до файла  
  
print\_date(data)#вывод таблицы  
print("\n")  
  
print\_date(sort\_by\_fio(data)) #сортировка по фамилии  
print("\n")  
  
print\_date(sort\_by\_number(data)) #сортировка по номеру  
print("\n")  
  
print\_date(sort\_by\_values(data, 22)) #сортировка по значению номера больше 22  
  
def add(file,c,number,date,fio,stip,kuda): #функция сохраненеия  
 with open(file, "w",encoding='utf-8') as f:  
 for a,b in c.items():  
 f.write(f"{a},{b['number']},{b['date']},{b['fio']},{b['stip']}.{b['kuda']}\n")  
 f.write(f"{len(c) + 1},{number},{date},{fio},{stip},{kuda}\n")  
 с.update({len(c) + 1: {"number": number, "date": date,"fio":fio, "stip": stip, "kuda": kuda}})  
#add('D:\\papka\\data.csv.txt', 5, 555, '31.01.2023', 'Петр Петров', 5000, 'в горы') #сохранение новых данных в файл