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 ИВТАСбд-22

Курдюк В.С.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023**Задание по варианту**

Вариант 11:

Посты: №, ник автора, текст поста, количество лайков

1. Пусть дана некоторая директория (папка). Посчитать количество файлов в данной директории (папке) и вывести на экран.
2. Пусть   дан   файл   data.csv,   в   котором   содержится информация в соответствии с вариантом:

Считать информацию из файла в соответствующую структуру (словарь):

* 1. Вывести информацию об объектах, отсортировав их по одному полю (строковому).
  2. Вывести информацию об объектах, отсортировав их по одному полю (числовому).
  3. Вывести информацию, соответствующую какому-либо критерию (например, для студентов - тех, у кого возраст больше какого-либо значения)

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

**Описание реализации**

Для ввода таблицы с файла была использована библиотека csv, в которой благодаря методу .DictReader() возможно считать таблицу, т.к. все таблицы в формате .csv представлены в виде строк, элементы которых разделяются в нашем случае точкой с запятой, для этого надо в атрибуте delimetr указать в кавычках данный символ. После чего каждую строку в виде словаря записываем в список, чтобы получить список словарей.

Листинг 1. Код ввода таблицы

|  |
| --- |
| mylist = {}  with open("data.csv", newline='') as file:  reader = csv.DictReader(file, delimiter=";")  mylist = [row for row in reader] |

Сортировка была выполнена методом .sort класса List, в которой в качестве ключа сортировки указывается название столбца лямбда-функцией.

Листинг 2. Функция сортировки по имени

|  |
| --- |
| def sort\_name(mylist):  print("Сортировка по названию(по убыванию):")  mylist.sort(key=lambda x: x['Name'])  printlist(mylist) |

Листинг 3. Функция сортировки количеству лайков

|  |
| --- |
| def sort\_like(mylist):  print("Сортировка по стоимости(по убыванию):")  mylist.sort(key=lambda x: int(x['Likes']))  new\_list = list(reversed(mylist))  printlist(new\_list) |

Вывод элементов таблицы, с учётом определённых условий был осуществлён циклом for, в котором перебираются все элементы и сравниваются с введённым пользователем значением. После чего выводится требуемая нам строка.

Листинг 4. Функция вывода с условием

|  |
| --- |
| def sort\_cost(mylist, sign, quantity):  for x in mylist:  if sign == '>':  if int(x["Likes"]) > quantity:  print(x)  elif sign == '<':  if int(x["Likes"]) < quantity:  print(x)  elif sign == '!=':  if x["Likes"] != quantity:  print(x)  elif sign == '>=':  if int(x["Likes"]) >= quantity:  print(x)  elif sign == '<=':  if int(x["Likes"]) <= quantity:  print(x)  elif sign == '==':  if x["Likes"] == quantity:  print(x) |

Для вывода таблицы был использован тот же алгоритм что и в вводе, за исключением того, что теперь используется метод .DictWriter(), в котором нужно указывать по каким полям следует вводить значения. После чего методом .writeheader() записываются заголовки таблицы (названия столбцов), и далее записывается сама таблица методом .writerows()

Листинг 5. Функция вывода в файл

|  |
| --- |
| fields = ["п»їNumber", "Name", "Text", "Likes"]  with open('output.csv', 'w', newline='') as file:  writer = csv.DictWriter(file, fieldnames=fields)  writer.writeheader()  for row in mylist:  writer.writerow(row)  print("Сохранение завершено") |

**Описание возникших затруднений**

В данной лабораторной работе возникли трудности с изучением различных методов из библиотеки csv, а также проблема с форматом при сохранении таблицы, после редактирования в excel, т.к. их вариаций несколько. Также не сразу стало понятно, что существует несколько вариантов формата файла csv. Также была проблема с первыми «вводными» символами таблицы(это проблема так и не была решена)

**Описание альтернативных способов решения**

Алгоритм можно было реализовать немного иначе, например вместо ввода, который был осуществлён блоком with можно было использовать функцию open(). А для сортировки можно использовать функцию sort().

Тестирование

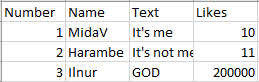


Рис 1. Входные данные

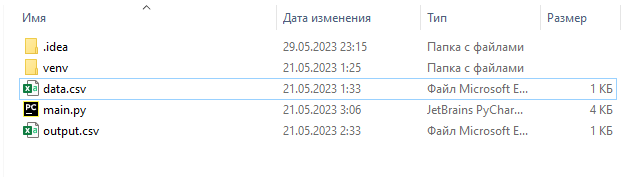


Рис 2. Входные данные

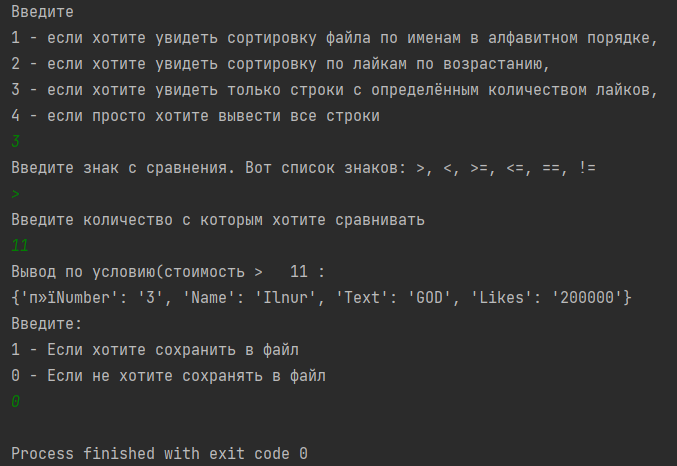


Рис 3. Результат выполнения программы без записи в файл

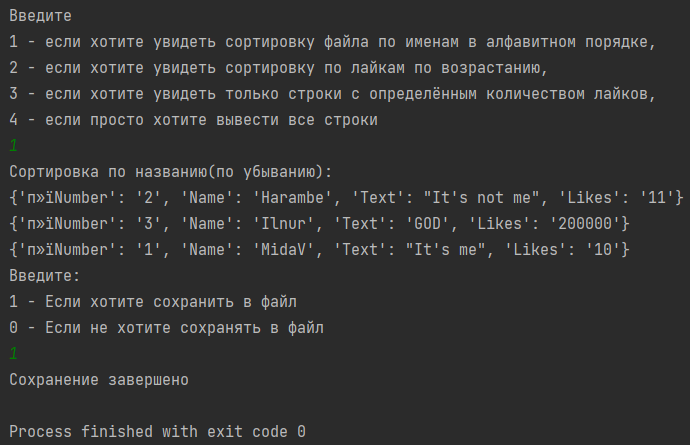


Рис 4. Результат выполнения программы с записью в файл

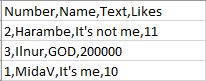


Рис 5. Результат выполнения программы с записью в файл

Вывод

В данной лабораторной работе были приобретены навыки владения библиотекой csv, знания использования списка словарей на языке Python а также знания использования файлов csv (чтение и запись). В результате выполнения была получена программа, которая считывает таблицу из csv файла, после чего сортирует по различным критериям и выводит полученную таблицу на экран и записывает в файл csv.

**Код программы**

import os

import csv

def printlist(mylist):

for x in mylist:

print(x)

def sort\_cost(mylist, sign, quantity):

for x in mylist:

if sign == '>':

if int(x["Likes"]) > quantity:

print(x)

elif sign == '<':

if int(x["Likes"]) < quantity:

print(x)

elif sign == '!=':

if x["Likes"] != quantity:

print(x)

elif sign == '>=':

if int(x["Likes"]) >= quantity:

print(x)

elif sign == '<=':

if int(x["Likes"]) <= quantity:

print(x)

elif sign == '==':

if x["Likes"] == quantity:

print(x)

def sort\_name(mylist):

print("Сортировка по названию(по убыванию):")

mylist.sort(key=lambda x: x['Name'])

printlist(mylist)

def sort\_like(mylist):

print("Сортировка по стоимости(по убыванию):")

mylist.sort(key=lambda x: int(x['Likes']))

new\_list = list(reversed(mylist))

printlist(new\_list)

if \_\_name\_\_ == '\_\_main\_\_':

print("Количество файлов в директории:", len(os.listdir(path=".")))

mylist = {}

with open("data.csv", newline='') as file:

reader = csv.DictReader(file, delimiter=";")

mylist = [row for row in reader]

key = int(

input("Введите \n1 - если хотите увидеть сортировку файла по именам в алфавитном порядке, \n2 - если хотите "

"увидеть сортировку по лайкам по возрастанию, \n3 - если хотите увидеть только строки с определённым "

"количеством лайков, \n4 - если просто хотите вывести все строки\n")

)

if key == 1:

sort\_name(mylist)

elif key == 2:

sort\_like(mylist)

elif key == 3:

sign = input("Введите знак с сравнения. Вот список знаков: >, <, >=, <=, ==, !=\n")

quantity = int(input("Введите количество с которым хотите сравнивать\n"))

print("Вывод по условию(стоимость", sign, " ", quantity, ": ")

sort\_cost(mylist, sign, quantity)

elif key == 4:

printlist(mylist)

print("Введите:\n1 - Если хотите сохранить в файл\n0 - Если не хотите сохранять в файл")

check = int(input())

if check == 1:

fields = ["п»їNumber", "Name", "Text", "Likes"]

with open('output.csv', 'w', newline='') as file:

writer = csv.DictWriter(file, fieldnames=fields)

writer.writeheader()

for row in mylist:

writer.writerow(row)

print("Сохранение завершено")

else:

exit()