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 19

Выполнил:

студент группы ИВТАПбд-21

Романов Н.Ю

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту**

Вариант 19:

Чеки: №, дата и время, сумма, наименование 1 товарной позиции

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. Функция ввода таблицы

|  |
| --- |
| def inputlist(filename):  mylist = []  with open(filename, newline='') as r\_file:  file\_reader = csv.DictReader(r\_file, delimiter=";")  for row in file\_reader:  mylist.append(row)  return mylist |

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

Листинг 2. Функция сортировки

|  |
| --- |
| def printsortlist(mylist, data, reversed=0, integ=0):  if integ:  mylist.sort(key=lambda x: int(x[data]), reverse=reversed)  else:  mylist.sort(key=lambda x: x[data], reverse=reversed)  printlist(mylist)  return mylist |

Вывод элементов таблицы, с учётом определённых условий был осуществлён циклом for, в котором перебираются все элементы и проверяется не является элемент определённого заголовка больше заданного значения. После чего выводится требуемая нам строка.

Листинг 3. Функция вывода с условием

|  |
| --- |
| def printiflist(mylist, data, value):  for x in mylist:  if int(x[data]) > value:  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 4. Функция вывода в файл

|  |
| --- |
| def outputlist(filename, mylist):  with open(filename, 'w', newline='') as w\_file:  writer = csv.DictWriter(w\_file, ["num", "Date", "Cost", "Name"], delimiter=";")  writer.writeheader()  writer.writerows(mylist)  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.

**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

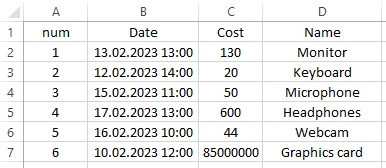


Рис 1. Входные данные

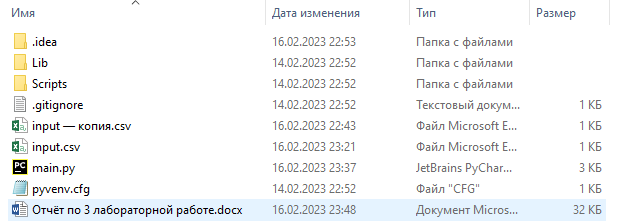


Рис 2. Входные данные

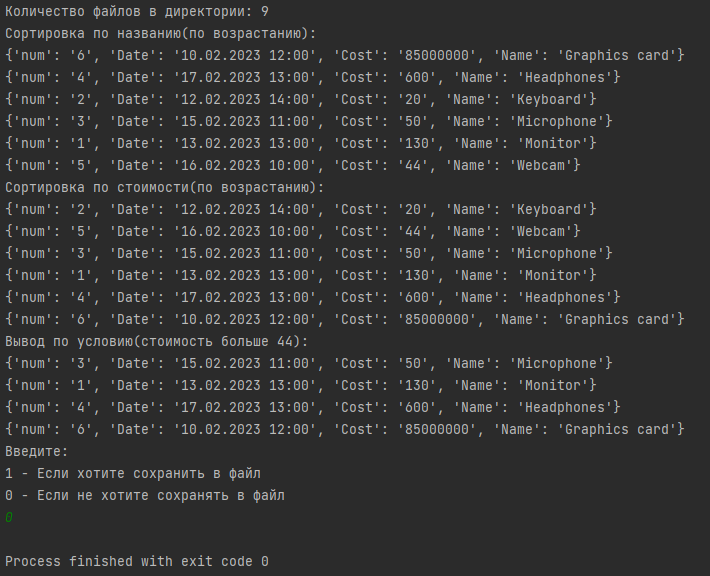


Рис 3. Результат выполнения программы без записи в файл

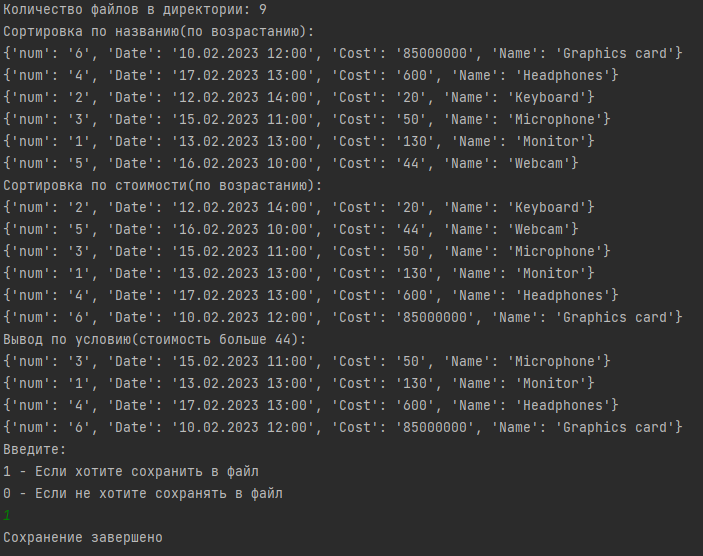


Рис 4. Результат выполнения программы с записью в файл

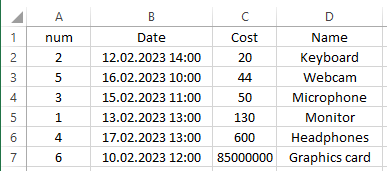


Рис 5. Результат выполнения программы с записью в файл

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 inputlist(filename):  
 mylist = []  
 with open(filename, newline='') as r\_file:  
 file\_reader = csv.DictReader(r\_file, delimiter=";")  
 for row in file\_reader:  
 mylist.append(row)  
 return mylist  
  
  
def outputlist(filename, mylist):  
 with open(filename, 'w', newline='') as w\_file:  
 writer = csv.DictWriter(w\_file, ["num", "Date", "Cost", "Name"], delimiter=";")  
 writer.writeheader()  
 writer.writerows(mylist)  
 print("Сохранение завершено")  
  
  
def printlist(mylist):  
 for x in mylist:  
 print(x)  
  
  
def printsortlist(mylist, data, reversed=0, integ=0):  
 if integ:  
 mylist.sort(key=lambda x: int(x[data]), reverse=reversed)  
 else:  
 mylist.sort(key=lambda x: x[data], reverse=reversed)  
 printlist(mylist)  
 return mylist  
  
  
def printiflist(mylist, data, operation, value):  
 for x in mylist:  
 if (operation == '>'):  
 if int(x[data]) > value:  
 print(x)  
 elif (operation == '<'):  
 if int(x[data]) < value:  
 print(x)  
 elif (operation == '!='):  
 if x[data] != value:  
 print(x)  
 elif (operation == '>='):  
 if int(x[data]) >= value:  
 print(x)  
 elif (operation == '<='):  
 if int(x[data]) <= value:  
 print(x)  
 elif (operation == '=='):  
 if x[data] == value:  
 print(x)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 print("Количество файлов в директории:", len(os.listdir(path=".")))  
  
 mylist = inputlist("input.csv")  
  
 print("Сортировка по названию(по возрастанию):")  
 mylist = printsortlist(mylist, "Name", 0)  
  
 print("Сортировка по стоимости(по возрастанию):")  
 mylist = printsortlist(mylist, "Cost", 0, 1)  
  
 print("Вывод по условию(стоимость больше 44):")  
 printiflist(mylist, "Cost", ">", 44)  
  
 print("Введите:\n1 - Если хотите сохранить в файл\n0 - Если не хотите сохранять в файл")  
 check = int(input())  
 if check == 1:  
 outputlist("input.csv", mylist)  
 else:  
 exit()