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 №4**

Выполнил:

студент группы ИВТАПбд-21

Галацков И.А.

Проверил(а):

преподаватель кафедры «ВТ»

Исхаков И. И.

Ульяновск

2023

**Постановка задачи**

Необходимо реализовать предложенный вариант задания. Номер задания студента определяется порядковым номером студента в журнале старосты.

Финансовые переводы: №, дата и время, сумма, описание транзакции

Должны быть соблюдены 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.

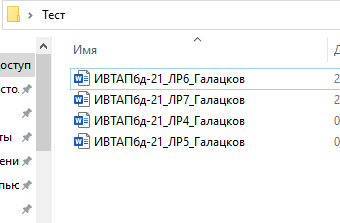
**Описание реализации**

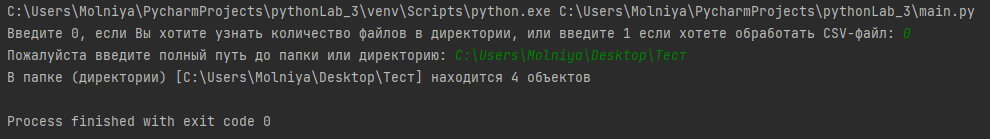
Импортируем библиотеки csv, datatime, pathlib. Была реализована функция пользовательского ввода userInput(): пользователю предлагается ввести 0, для просмотра количества файлов в директории или 1, для обработки сsv-файла. Был реализован подсчет количества файлов в директории: в переменную path, после происходит проверка на существование данного пути, если путь существует, то при помощи функции iterdir() подсчитываем количество файлов. Была реализована обработка csv-файла, сначала считываются данные из файла, далее с помощью функции sorted() мы сортируем данные согласно требованию работы, и записываем их в файл.

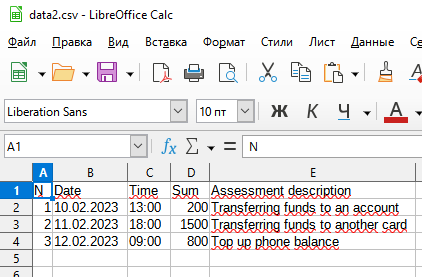
**Исходный код реализации**

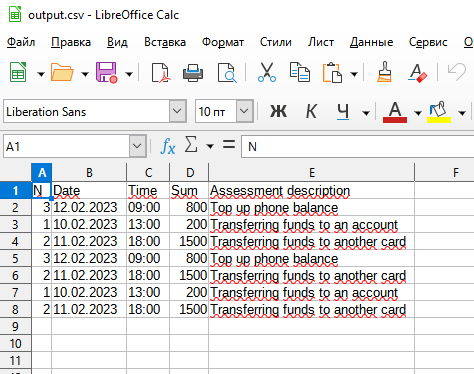
import csv  
import datetime  
from pathlib import Path  
  
def userInput(): #Функция ввода пользователя  
 while True:  
 try:  
 a = int(input("Введите 0, если Вы хотите узнать количество файлов в директории, или введите 1 если хотете обработать CSV-файл: "))  
 except ValueError:  
 print("Пожалуйста введите целое число из предложенных вариантов")  
 else:  
 if ((a != 0) and (a != 1)): #обработка ошибки ввода  
 print("Пожалуйста введите целое число из предложенных вариантов")  
 else:  
 return a  
 break  
  
a = userInput()  
  
  
if (a == 0):  
 path = input("Пожалуйста введите полный путь до папки или директорию: ")  
 directory = Path(path)  
  
 if not directory.is\_dir():  
 print(ValueError(f"[{directory}] не существует или не является директорией!"))  
 else:  
 print(f"В папке (директории) [{path}] находится {len(list(directory.iterdir()))} объектов")  
 #print(f"В папке (директории) [{path}] и в её папках (поддиректориях) находится {len(list(directory.rglob('\*')))} объектов")  
  
  
elif(a == 1):  
 fI = open("C:\\Users\\Molniya\\Desktop\\data2.csv", 'r')  
  
 reader = csv.DictReader(fI, fieldnames=None, restkey=None, restval=None, dialect="excel") #переменная для чтения файла формата csv  
  
 l = list(reader) #список содержащий данные из файла  
  
 print(l)  
  
 sSL = sorted(l, key=lambda d: str(d['Assessment description'])) # Сортировка по строковомы значению  
 nSL = sorted(l, key=lambda d: str(d['N']), reverse=True) #Сортировка по числовому значению  
  
 print(sSL)  
 print(nSL)  
  
 tSL = []  
  
 # Вывод информации по времени  
 for line in l:  
 time = datetime.datetime.strptime(line['Time'], '%H:%M')  
 if time.hour > 14:  
 tSL.append(line)  
  
  
 with open("C:\\Users\\Molniya\\Desktop\\output.csv", 'w', newline='') as csvfile:  
  
 writer = csv.DictWriter(csvfile, fieldnames = reader.fieldnames) #переменная для записи данных в файл  
 writer.writeheader()  
  
 for row in sSL:  
 writer.writerow(row)  
  
 for row in nSL:  
 writer.writerow(row)  
  
 for row in tSL:  
 writer.writerow(row)

**Тестирование**









**Описание возникших затруднений**

Основные затруднения при выполнении данной лабораторной работой были связаны со знакомством с форматом .csv и Python-библиотеки csv для импорта и экспорта для электронных таблиц и баз данных, а также с сортировкой списка словарей через лямбда-функции.

**Вывод**

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