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 №4**

Дисциплина: «Разработка профессиональных приложений»

**«Файлы и словари»**

Вариант №4

Выполнил:

студент группы ИВТАСбд-22

Данилин Н.Г.

Проверил:

преподаватель кафедры

«Вычислительная техника»

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 os

lst = os.listdir("D://3lab") # директория

number\_files = len(lst)

print("\n")

print (number\_files)

print("файлов в папке")

print("\n")

class row():

idx = 0

def \_\_init\_\_(self, idx: int):

self.idx = idx

def get\_idx(self):

return self.idx

def set\_idx(self, val):

self.idx = val

class model(row):

idx = 0

number = 0

date = ""

fio = ""

stip = ""

kuda = ""

def \_\_init\_\_(self, idx: int, number: int, date: str, time: str, summ: int, opis: str):

super().\_\_init\_\_(idx)

self.idx = idx

self.number = number

self.date = date

self.time = time

self.summ = summ

self.opis = opis

def \_\_str\_\_(self):

return f"Запись №{self.idx}: номер = {self.number}, дата = {self.date}, время = {self.time}, summ = {self.summ}, описание = {self.opis}"

def \_\_repr\_\_(self):

return f"model(idx={self.idx},number={self.number},date={self.date},time={self.time},summ={self.summ},opis={self.opis})"

def \_\_setattr\_\_(self, \_\_name, \_\_value):

self.\_\_dict\_\_[\_\_name] = \_\_value

class Data():

file\_path = ""

data = {}

pointer = 0

def \_\_init\_\_(self, file):

self.file\_path = file

self.data = self.parse(file)

def \_\_str\_\_(self):

d\_str = '\n'.join([str(rm) for rm in self.data])

return f"Контейнер хранит в себе следущее:\n{d\_str}"

def \_\_repr\_\_(self):

return f"Data({[repr(rm) for rm in self.data]})"

def \_\_iter\_\_(self):

return self

def \_\_next\_\_(self):

if self.pointer >= len(self.data):

self.pointer = 0

raise StopIteration

else:

self.pointer += 1

return self.data[self.pointer - 1]

def \_\_getitem\_\_(self, item):

if not isinstance(item, int):

raise TypeError("Индекс должен быть целым числом.")

if 0 <= item < len(self.data):

return self.data[item]

else:

raise IndexError("Неверный индекс.")

def as\_generator(self):

self.pointer = 0

while self.pointer < len(self.data):

yield self.data[self.pointer]

self.pointer += 1

@staticmethod

def parse(file):

parsed = []

with open(file, "r", encoding='utf-8') as raw\_csv:

for line in raw\_csv:

(idx, number, date, time, summ, opis) = line.replace("\n", "").split(",")

parsed.append(model(int(idx), int(number), date, time, summ, opis))

return parsed

def sorted\_by\_kuda(self):

return sorted(self.data, key=lambda f: f.opis)

def sorted\_by\_number(self):

return sorted(self.data, key=lambda f: f.number)

def value(self, value):

r = []

for d in self.data:

if (d.number > value):

r.append(d)

return r

def add\_new(self, number, date, time, summ, opis):

self.data.append(model(len(self.data) + 1, number, date, time, summ, opis))

self.save(self.file\_path, self.data)

@staticmethod

def save(file, new\_data):

with open(file, "w", encoding='utf-8') as f:

for r in new\_data:

f.write(f"{r.idx},{r.number},{r.date},{r.time},{r.summ},{r.opis}\n")

def print(self):

for r in self.data:

print(f"Запись №{r.idx}: номер = {r.number}, дата= {r.date},время = {r.time}, сумма = {r.summ}, описание = {r.opis}")

def printd(self, d):

for r in d:

print(f"Запись №{r.idx}: номер = {r.number}, дата= {r.date},время = {r.time}, сумма = {r.summ}, описание = {r.opis}")

def get\_files\_count\_in\_directory(file):

(loc, dirs, files) = next(os.walk(file))

return len(files)

data = Data("D:\\3lab\\data.csv.txt")

# \_\_repr\_\_()

print(repr(data), "\n")

# \_\_str\_\_()

print(data, "\n")

# Итератор

for item in iter(data):

print(item)

print("-" \* 64)

# Генератор

for item in data.as\_generator():

print(item)

print("-" \* 64)

print("\n")

data.printd(data.sorted\_by\_number())#сортировка по номеру

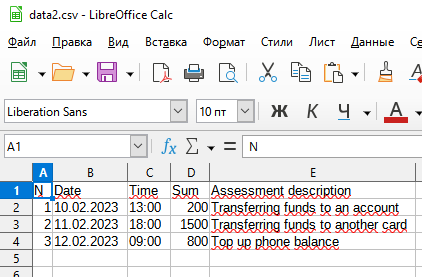
print("\n")

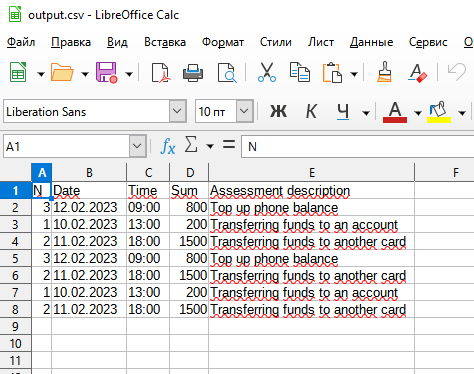
data.printd(data.sorted\_by\_kuda())#сортировка по имени

print("\n")

data.printd(data.value(12))#номер больше 12

**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.