**Лабораторная работа №1. Работа со списками, словарями и генераторами.**

## Цель: Закрепить методы работы со списками, словарями и генераторами

**Ход работы**

**Задание 1**

Создать словарь из пар ключ-значение, где ключи - это буквы своей фамилии, а значения - это порядковый номер буквы в алфавите, от которого взят основание натурального логарифма ex (лето).

На рисунке 1 приведен код, который в цикле выполняет проход по исходному словарю (ключи это буквы фамилии, а значения номер буквы алфавита) и вычисляет натуральный логарифм от каждого значения, перезаписывая исходный словарь.

*import* math

*#1 e^x*

dct = {'П': 17, 'Е': 6, 'Т': 20, 'Р': 18, 'О': 16, 'В':3}

*for* i *in* dct:

    dct[i] = math.exp(dct[i])

*print*("словарь с основанием натурального логарифма от букв фамилии ", dct)

Рисунок 1 ­– Листинг кода для задания 1

На рисунке 2 при помощи функции print в консольном окне напечатан созданный словарь.



Рисунок 2 ­– Вывод словаря в консольном окне

**Задание 2**

Отсортировать его по ключу в алфавитном порядке и сохранить в файл.

На рисунке 3 приведён листинг кода для задания 2. Сортировка выполняется при помощи встроенного метода sorted. Результаты записываются в файл *'sort\_keys.txt'*.

dct\_sort1 = {}

keys\_sort =  *sorted*(dct, key=dct.get)*#сортирует и возращают значения #  в*

*for* k *in* keys\_sort:

    dct\_sort1[k] = dct[k]  *# 3) от меньшего к большему*

*print*("словарь отсортированный по ключу ", dct\_sort1)

FILE = *open*('sort\_keys.txt', 'w')

FILE.write(str(dct\_sort1))

FILE.close()

Рисунок 3 ­– Листинг кода для задания 2

На рисунке 4 приведено содержимое записанного файла. Как видно словарь отсортирован по ключам.

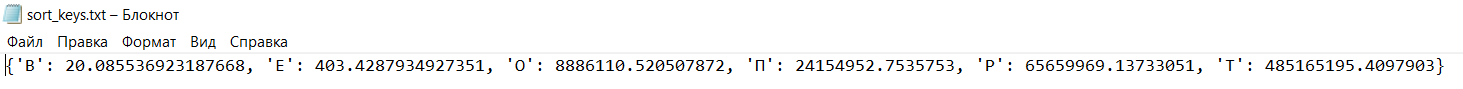


Рисунок 4 ­– Отсортированный словарь по ключам, записанный в файл

**Задание 3**

Отсортировать по значениям от меньшего к большему и сохранить в файл.

На рисунке 5 приведён листинг кода для задания 2. Сортировка выполняется при помощи встроенного метода sorted. Результаты записываются в файл 'values\_keys.txt'.

sort\_dict2 = {}

sort\_values = *sorted*(dct.values(), reverse = False) *#сортирует и возращают значения*

*for* i *in* sort\_values:

*for* j *in* dct.keys():

*if* dct[j]==i:

      sort\_dict2 [j]= dct[j]

*break*

*print*("словарь отсортированный по значению  ", sort\_dict2)

FILE = *open*('sort\_values.txt', 'w')

FILE.write(str(sort\_dict2))

FILE.close()

Рисунок 5 ­– Листинг кода для задания 3

На рисунке 6 приведено содержимое записанного файла. Как видно словарь отсортирован по значениям.

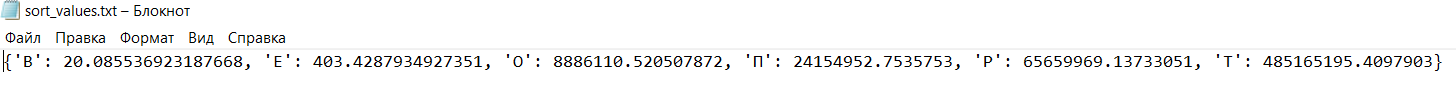


Рисунок 6 ­– Отсортированный словарь по значениям, записанный в файл

**Задание 4**

Для задания 1 написать функцию-генератор.

Листинг кода для задания 4 приведён на рисунке 7.

*#Функция генератор*

dct = {'П': 17, 'Е': 6, 'Т': 20, 'Р': 18, 'О': 16, 'В':3}

*def* x(n):

*if* n == 0:

*return* 1

*yield* math.exp(n)

*for* i *in* dct:

    dct[i] = (*next*(x(dct[i])))

*print*("словарь, созданный при помощи генератора", dct)

Рисунок 7 ­– Листинг кода для задания 4

На рисунке 8 при помощи функции print в консольном окне напечатан созданный словарь.



Рисунок 8 ­– Вывод словаря, созданного при помощи генератора

**Задание 5**

Создать список из *значений* словаря и разделить его на два: один из значений меньше среднего по списку, второй - среднее и выше.

Листинг кода для задания 5 приведён на рисунке 9.

dct = {'П': 17, 'Е': 6, 'Т': 20, 'Р': 18, 'О': 16, 'В':3}

list = [] *#создаем список и записываем туда все значения*

*for* i *in* dct:

  list.append(dct[i])

sr = *sum*(list) / *len*(list) *#среднее арифметическое*

list1 = [] *# список для < среднего*

list2 = [] *#список для >= среднего*

*for* i *in* list: *# перебираем значения и записываем их в соответствующий список*

*if* i < sr:

    list1.append(i)

*elif* i>= sr:

    list2.append(i)

*print*("список значений меньше среднего ", list1)

*print*("список значений больше среднего ", list2)

Рисунок 9 ­– Листинг кода для задания 5

На рисунке 10 при помощи функции print в консольном окне напечатаны два списка.

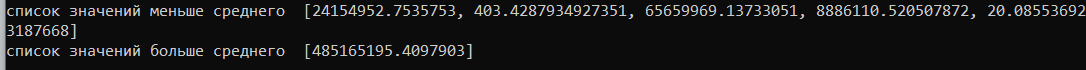


Рисунок 10 ­– Вывод списков со значениями меньше среднего и больше среднего

## Вывод: В результате выполнения лабораторной работы получены практические навыки работы со списками, словарями и генераторами. Изучены основные методы работы, применяемы в python.