

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное

образовательное учреждение высшего

образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №7

Работа со словарями в языке Python

Выполнил студент группы ИТС-б-о-21-1

Романов Платон Дмитриевич

« » _____ 20__ г.

Подпись студента _____

Проверил: Доцент, к.т.н, доцент

кафедры инфокоммуникаций

Воронкин А. В.

Работа защищена с оценкой: _____

(подпись)

Ставрополь, 2022

Цель работы: приобретение навыков по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.

Ссылка на репозиторий - <https://github.com/lesnaya1shelupon/7lab>

Задание 1. Использовать словарь, содержащий следующие ключи: фамилия, имя, знак Зодиака, дата рождения (массив из трех чисел). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры, записи должны быть упорядочены по датам рождения, вывод на экран информации о людях, родившихся под знаком, название которого введено с клавиатуры, если таких нет, выдать на дисплей соответствующее сообщение.

```
import sys
from datetime import datetime, timedelta
import datetime

if __name__ == '__main__':
    spisok = []
    new_spisok = []

    while True:
        command = input(">>> ").lower()

        if command == 'exit':
            break

        elif command == 'add':
            surname = input("Фамилия: ")
            name = input("Имя: ")
            post = input("Знак Зодиака: ")
            day, month, year = input("Дата рождения: ").split(" ")
            datas = f'{year} {month} {day}'
            spisok_new = {
                'surname': surname,
                'name': name,
                'post': post,
                'data': datas,
            }

            spisok.append(spisok_new)

            if len(spisok) > 1:
                spisok.sort(key=lambda item: item.get('data', ''))

        elif command == 'list':
            line = '+-{}-+-{}-+-{}-+-{}-+-{}-+'.format(
                '-' * 4,
                '-' * 15,
                '-' * 30,
                '-' * 20,
                '-' * 15,
            )
            print(line)
            print(
                '| {:^4} | {:^15} | {:^30} | {:^20} | {:^15} |'.format(
                    "№",
                    "Дата рождения",
                    "Фамилия",
                    "Имя",
                    "Знак Зодиака"
                )
            )
            print(line)

            # Вывести данные о всех сотрудниках.
            for idx, spisok_new in enumerate(spisok, 1):
                print(
                    '| {:>4} | {:<15} | {:<30} | {:<20} | {:<15} |'.format(
```

Рисунок 1. Код задачи

```

57         '| {:>4} | {:<15} | {:<30} | {:<20} | {:<15} | '.format(
58             idx,
59             spisok_new.get('data', ''),
60             spisok_new.get('surname', ''),
61             spisok_new.get('name', ''),
62             spisok_new.get('post', 0)
63         )
64     )
65 )
66
67 print(line)
68 elif command == 'find':
69     find = input("Введите знак Зодиака: ")
70     for find_item in spisok:
71         if find == find_item['post']:
72             new_spisok.append(find_item)
73             # print(find_item)
74
75     if len(new_spisok) > 0:
76         line_new = '+-{}-+-{}-+-{}-+-{}-+-{}-+'.format(
77             '-' * 4,
78             '-' * 15,
79             '-' * 30,
80             '-' * 20,
81             '-' * 15
82         )
83         print(line_new)
84         print(
85             '| {:^4} | {:^15} | {:^30} | {:^20} | {:^15} | '.format(
86                 "№",
87                 "Дата рождения",
88                 "Фамилия",
89                 "Имя",
90                 "Знак Зодиака"
91             )
92         )
93     )
94     print(line_new)
95
96     for idx_new, spisok_new_new in enumerate(new_spisok, 1):
97         print(
98             '| {:>4} | {:<15} | {:<30} | {:<20} | {:<15} | '.format(
99                 idx_new,
100                 spisok_new_new.get('data', ''),
101                 spisok_new_new.get('surname', ''),
102                 spisok_new_new.get('name', ''),
103                 spisok_new_new.get('post', 0)
104             )
105         )
106     )
107
108     print(line_new)
109 else:
110     print('Таких пользователей не найдено', file=sys.stderr)
111 elif command == 'help':
112     print('Список команд:\n')
113     print('add - добавить пользователя ')

```

Рисунок 2. Код задачи

```

106         )
107
108         print(line_new)
109     else:
110         print('Таких пользователей не найдено', file=sys.stderr)
111 elif command == 'help':
112     print('Список команд:\n')
113     print('add - добавить пользователя.')
114     print('list - вывести список пользователей.')
115     print('find <Знак зодиака> - запросить пользователей по знаку Зодиака.')
116     print('help - Справочник.')
117     print('exit - Завершить работу программы.')
118 else:
119     print(f'Команда <{command}> не существует.', file=sys.stderr)
120     print('Введите <help> для просмотра доступных команд')
121

```

Рисунок 3. Код задачи

```

D:\Trash\venv\Scripts\python.exe "C:/git/7lab/ind zad 15 var.py"
>>> add
Фамилия: Романов
Имя: Платон
знак Зодиака: Весы
Дата рождения: 27 09 2001
>>> add
Фамилия: Антонов
Имя: Антон
знак Зодиака: Рак
Дата рождения: 0 07 2000
>>> list
+-----+-----+-----+-----+-----+
| № | Дата рождения | Фамилия | Имя | Знак Зодиака |
+-----+-----+-----+-----+-----+
| 1 | 2000 07 0 | Антонов | Антон | Рак |
| 2 | 2001 09 27 | Романов | Платон | Весы |
+-----+-----+-----+-----+-----+
>>>

```

Рисунок 4. Окно вывода задачи

```
Дата рождения: 27 09 2001
>>> add
Фамилия: Пушкин
Имя: Александр
знак Зодиака: Весы
Дата рождения: 20 09 2001
>>> add
Фамилия: Антонов
Имя: Антон
знак Зодиака: Дева
Дата рождения: 01 01 2000
>>> find
Введите знак Зодиака: Весы
+-----+-----+-----+-----+
| № | Дата рождения | Фамилия | Имя | Знак Зодиака |
+-----+-----+-----+-----+
| 1 | 2001 09 20 | Пушкин | Александр | Весы |
| 2 | 2001 09 27 | Романов | Платон | Весы |
+-----+-----+-----+-----+
>>>
```

Рисунок 5. Окно вывода задачи

Ответы на контрольные вопросы:

1. Что такое словари в языке Python?

Словарь (dict) представляет собой структуру данных (которая ещё называется ассоциативный массив), предназначенную для хранения произвольных объектов с доступом по ключу.

2. Может ли функция *len()* быть использована при работе со словарями?

Да может! Функция *len()* возвращает длину (количество элементов) в объекте.

3. Какие методы обхода словарей Вам известны?

У словаря как класса есть метод *items()*, который создает особую структуру, состоящую из кортежей. Каждый кортеж включает ключ и значение:

```
>>> n = nums.items()
>>> n
dict_items([(1, 'one'), (2, 'two'), (3, 'three')])
```

Методы словаря *keys()* и *values()* позволяют получить отдельно перечни ключей и значений. Так что если, например, надо перебрать только значения или только ключи, лучше воспользоваться одним из этих методов:

```
>>> v_nums = []
>>> for v in nums.values():
...     v_nums.append(v)
...
>>> v_nums
['one', 'two', 'three']
```

Так же существуют методы *clear()*, *copy()*, *fromkeys()*, *get()*, *pop()*, *popitem()*, *setdefault()*, *update()*.

Метод *clear()* удаляет все элементы словаря, но не удаляет сам словарь. В итоге остается пустой Словарь. Метод *fromkeys()* позволяет создать словарь из списка, элементы которого становятся ключами. Применять метод можно как классу *dict*, так и к его объектам. Метод *get()* позволяет получить элемент по его ключу. Метод *pop()* удаляет из словаря элемент по указанному ключу и возвращает значение удаленной пары. Метод *popitem()* не принимает аргументов, удаляет и возвращает произвольный элемент. С помощью *setdefault()* можно добавить элемент в словарь. С помощью *update()* можно добавить в словарь другой словарь.

4. Какими способами можно получить значения из словаря по ключу?

Операция `dict[key]` вернет элемент словаря `dict` с ключом `key`. Операция вызывает исключение `KeyError`, если ключ `key` отсутствует в словаре.

5. Какими способами можно установить значение в словаре по ключу?

Операция `d[key] = value` добавит в словарь `dict` новый элемент - пару ключ-значение.

Если в словаре существует ключ `key` то эта операция присвоит ключу `key` новое значение `value`.

6. Что такое словарь включений?

Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка. Как и в случае со списком, мы можем использовать условный оператор внутри словаря включения, чтобы получить только элементы словаря, удовлетворяющие заданному критерию.

7. Самостоятельно изучите возможности функции *zip()* приведите примеры ее использования.

Функция *zip()* создает итератор кортежей, который объединяет элементы

каждой из переданных последовательностей `*iterables`.

8. Самостоятельно изучите возможности модуля *datetime*. Каким функционалом по работе с датой и временем обладает этот модуль?

Datetime — важный элемент любой программы, написанной на Python. Этот модуль позволяет управлять датами и временем, представляя их в таком виде, в котором пользователи смогут их понимать.

Datetime включает различные компоненты:

- `date` — хранит дату
- `time` — хранит время
- `datetime` — хранит дату и время

Вывод: приобрели навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.