

**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

Клочко Никита Александрович

« » _____ 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/NikitaKloch/labarnaya7>

Задание 1.

11. Использовать словарь, содержащий следующие ключи: фамилия, имя; номер телефона; дата рождения (список 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 человеке, номер телефона которого введен с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.

```
1  import sys
2  from datetime import datetime, timedelta
3  import datetime
4
5
6  # Фамилия
7  # Имя
8  # Номер телефона
9  # Дата рождения (ДД ММ ГГГГ)
10
11  listing = []
12
13  while True:
14      command = input(">>> ").lower()
15
16
17      if command == 'exit':
18          break
19
20
21      elif command == 'add':
22          surname = input('Фамилия: ')
23          name = input('Имя: ')
24          phone = input('Телефон: ')
25
26          day, month, year = input('Дата рождения (ДД ММ ГГГГ): ').split(' ')
27          dates = f'{day} {month} {year}'
28
29          time_list = {
30              'surname': surname,
31              'name': name,
32              'phone': phone,
33              'data': dates
34          }
35
36          listing.append(time_list)
37
38      if len(listing) > 1:
39          listing.sort(key=lambda item: item.get('data', ''))
```

```

47         '- ' * 30
48     )
49     print(table)
50
51     print(
52         '| {:^4} | {:^30} | {:^15} | {:^30} | {:^30} | '.format(
53             "№",
54             "Фамилия",
55             "Имя",
56             "Телефон",
57             "Дата рождения"
58         )
59     )
60
61
62     print(table)
63
64     for idx, spisok_new in enumerate(listing, 1):
65         print(
66             '| {:>4} | {:<30} | {:<15} | {:<30} | {:<30} | '.format(
67                 idx,
68                 spisok_new.get('surname', ''),
69                 spisok_new.get('name', ''),
70                 spisok_new.get('phone', ''),
71                 spisok_new.get('data', 0)
72             )
73         )
74
75     print(table)
76
77     elif command == 'phone':
78         search_phone = input('Введите номер телефона: ')
79         new_listing = []
80         for phone_item in listing:
81             if search_phone == phone_item['phone']:
82                 new_listing.append(phone_item)
83
84         if len(new_listing) > 0:
85             line_new = '+-{}-+-{}-+-{}-+-{}-+'.format(
86                 '- ' * 4,
87                 '- ' * 30,
88                 '- ' * 15,
89                 '- ' * 30,
90                 '- ' * 30

```

```

91             )
92             print(line_new)
93             print(
94                 '| {:^4} | {:^30} | {:^15} | {:^30} | {:^30} | '.format(
95                     "№",
96                     "Фамилия",
97                     "Имя",
98                     "Телефон",
99                     "Дата рождения"
100                 )
101             )
102         )
103         print(line_new)
104         for idx_new, spisok_new_new in enumerate(new_listing, 1):
105             print(
106                 '| {:>4} | {:<30} | {:<30} | {:<20} | {:<30} | '.format(
107                     idx_new,
108                     spisok_new_new.get('surname', ''),
109                     spisok_new_new.get('name', ''),
110                     spisok_new_new.get('phone', ''),
111                     spisok_new_new.get('data', 0)

```

```

115         print(line_new)
116     else:
117         print('Такого номера не найдено!', file=sys.stderr)
118 elif command == 'help':
119     print('Список команд:\n')
120     print('add - добавить пользователя.')
121     print('list - вывести список пользователей.')
122     print('find <Номер телефона> - запросить пользователей по номеру телефона.')
123     print('help - Справочник.')
124     print('exit - Завершить работу программы.')
125 else:
126     print(f'Команда <{command}> не существует.', file=sys.stderr)
127     print('Введите <help> для просмотра доступных команд')
128

```

Рисунок 1. Код 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: приобрел навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.