

**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

Тимофеева Марина Сергеевна

« » _____ 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/AlisaL1sa/primer7>

Задание 1.

17. Использовать словарь, содержащий следующие ключи: название товара; название магазина, в котором продается товар; стоимость товара в руб. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть размещены в алфавитном порядке по названиям товаров; вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, выдать на дисплей соответствующее сообщение.

```
1  import sys
2
3  # Название магазина (фильтр 2х)
4  # Название товара
5  # Стоимость
6
7  list_shop = []
8  spisok_new = []
9
10 while True:
11     command = input('>>> ').lower()
12
13     if command == 'exit':
14         break
15
16     elif command == 'add':
17         name_shop = input('Название магазина: ')
18         name_prodyckt = input('Название товара: ')
19         prise = input('Стоимость товара: ')
20
21         list_shop_new = {
22             'name_shop': name_shop,
23             'name_prodyckt': name_prodyckt,
24             'prise': prise
25         }
26
27         list_shop.append(list_shop_new)
28
29         if len(list_shop) > 1:
30             list_shop.sort(key=lambda item: item.get('name_shop', ''))
```

```

32 elif command == 'list':
33     line = '+-{}--{}--{}--{}--+'.format(
34         '-' * 6,
35         '-' * 20,
36         '-' * 30,
37         '-' * 20
38     )
39
40     print(line)
41     print(
42         '| {:^6} | {:^20} | {:^30} | {:^20} | '.format(
43             "№",
44             "Название магазина",
45             "Название товара",
46             "Стоимость"
47         )
48     )
49     print(line)
50
51     for idx, spisok_new in enumerate(list_shop, 1):
52         # print(spisok_new)
53         print(
54             '| {:>6} | {:<20} | {:<30} | {:<20} | '.format(
55                 idx,
56                 spisok_new.get('name_shop', ''),
57                 spisok_new.get('name_prodyckt', ''),
58                 spisok_new.get('prise', 0)
59             )
60         )

```

```

62     print(line)
63
64
65 elif command == 'prodyckt':
66     shop_sear = input('Введите название товара: ')
67     search_shop = []
68     for shop_sear_itme in list_shop:
69         if shop_sear == shop_sear_itme['name_prodyckt']:
70             search_shop.append(shop_sear_itme)
71
72     if len(search_shop) > 0:
73         line_new = '+-{}--{}--{}--{}--+'.format(
74             '-' * 6,
75             '-' * 20,
76             '-' * 30,
77             '-' * 20
78         )
79         print(line_new)
80
81         print(
82             '| {:^6} | {:^20} | {:^30} | {:^20} | '.format(
83                 "№",
84                 "Название магазина",
85                 "Название товара",
86                 "Стоимость"
87             )
88         )
89         print(line_new)
90

```

```

91         for idx_new, spisok_new_new in enumerate(search_shop, 1):
92             print(
93                 '| {:>6} | {:<20} | {:<30} | {:<20} | '.format(
94                     idx_new,
95                     spisok_new_new.get('name_shop', ''),
96                     spisok_new_new.get('name_prodyckt', ''),
97                     spisok_new_new.get('prise', '')
98                 )
99             )
100
101         print(line_new)
102     else:
103         print('Такого товара не найдено', file=sys.stderr)
104
105
106     elif command == 'help':
107         print('Список команд:\n')
108         print('add - добавить магазин.')
109         print('list - вывести список магазинов.')
110         print('prodyckt <Название> - запросить информацию о товаре.')
111         print('help - Справочник.')
112         print('exit - Завершить работу программы.')
113     else:
114         print(f'Команда <{command}> не существует.', file=sys.stderr)
115         print('Введите <help> для просмотра доступных команд')

```

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