

Министерство науки и высшего образования Российской  
Федерации Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**  
**дисциплины «Основы кроссплатформенного**  
**программирования»**

Выполнил:

Боженко Александр Иванович

1 курс, группа ИТС-б-о-21-1,

11.03.02 «Инфокоммуникационные  
технологии и системы связи»,

направленность (профиль)

«Инфокоммуникационные системы и сети»,

очная форма обучения

\_\_\_\_\_  
(подпись)

Руководитель практики:

Воронкин Р.А., канд. техн. наук, доцент

кафедры инфокоммуникаций

\_\_\_\_\_  
(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

## Тема: Работа со словарями в языке Python

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

### Ход работы:

#### Задание 1.

Решите задачу: создайте словарь, связав его с переменной `school`, и наполните данными, которые бы отражали количество учащихся в разных классах (1а, 1б, 2б, 6а, 7в и т. п.). Внесите изменения в словарь согласно следующему: а) в одном из классов изменилось количество учащихся, б) в школе появился новый класс, с) в школе был расформирован (удален) другой класс. Вычислите общее количество учащихся в школе

```
C: > Users > user > job-7 > school.py
You, 16 минут назад | 1 author (You)
1 school = {"1a": 23, "1б": 22, "2а": 21, "2б": 20, "3а": 19, "3б": 20, "4а": 18, "4б": 24, "5а": 23, "5б": 17,
2 "6а": 20, "6б": 15, "7а": 18, "7б": 25, "8а": 25, "8б": 26, "9а": 29, "9б": 17, "10а": 24, "10б": 22,
3 "11а": 24, "11б": 23} # Словарь классов с количеством учащихся
4
5 school["2а"] = 19 # Во 2а классе изменилось количество учащихся
6
7 school["1в"] = 15 # В школе появился 1в класс
8
9 del school["10б"] # Класс 10б был удален
10
11 print(f'Всего учеников в школе: {sum(school.values())}') # Выводим на экран общее число учащихся в школе
```

```
Всего учеников в школе: 466
PS C:\Users\user>
```

Рис 1.

#### Задание 2.

Решите задачу: создайте словарь, где ключами являются числа, а значениями – строки. Примените к нему метод `items()`, с помощью полученного объекта `dict_items` создайте новый словарь, "обратный" исходному, т. е. ключами являются строки, а значениями – числа.

```
C: > Users > user > job-7 > dict_items.py > dict_items
You, 18 минут назад | 1 author (You)
1 dict_items = { 1: 'a', 2: 'b' }.items()
2 inv_dict = lambda d: { v: k for k, v in d }
3 print(inv_dict(dict_items))

{'a': 1, 'b': 2}
PS C:\Users\user>
```

Рис 2.

### Задание 3.

8. Использовать словарь, содержащий следующие ключи: название пункта назначения; номер поезда; время отправления. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по номерам поездов; вывод на экран информации о поезде, номер которого введен с клавиатуры; если таких поездов нет, выдать на дисплей соответствующее сообщение.

```
Alex, 23 часа назад | 1 author (Alex)
1 import sys
2 if __name__ == '__main__':
3     # Список .
4     train = []
5     # Организовать бесконечный цикл запроса команд.
6     while True:
7         # Запросить команду из терминала.
8         command = input(">>> ").lower()
9         # Выполнить действие в соответствие с командой.
10        if command == 'exit':
11            break
12        elif command == 'add':
13            # Запросить данные .
14            race = input("Название пункта назначения ")
15            number = input("Номер поезда ")
16            time = input("время отправления ")
17            # Создать словарь.
18            trains = {
19                'race': race,
20                'number': number,
21                'time': time,
22            }
23            # Добавить словарь в список.
24            train.append(trains)
25            # Отсортировать список в случае необходимости.
26            if len(train) > 1:
27                train.sort(key=lambda item: item.get('race', ''))
28        elif command == 'list':
29            # Заголовок таблицы.
30            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
31                '-' * 4,
32                '-' * 30,
33                '-' * 20,
34                '-' * 20
35            )
36            print(line)
37            print(
```

```
>>> add
Название пункта назначения Ставрополь
Номер поезда 15
время отправления 12:00
>>> list
+-----+-----+-----+-----+
| No |          Пункт          | Номер | Время отправления |
+-----+-----+-----+-----+
|  1 |   Ставрополь   |    15 |         12:00     |
+-----+-----+-----+-----+
>>> 
```

Рис 3.

1. Что такое словари в языке Python?

Ответ: 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?

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

3. Какие методы обхода словарей Вам известны?

Ответ: 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 де 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?

Ответ: 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?

Ответ: 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?

Ответ: 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.

Ответ: 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?

Ответ: *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:** в ходе выполнения работ я приобрел навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.