

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе №9 по дисциплине:  
основы программной инженерии**

Выполнила:

студент группы ПИЖ-б-о-20-1

Лазарева Дарья Олеговна

Проверил:

доцент кафедры инфокоммуникаций

Романкин Р.А.

Ставрополь, 2021 г.

## Ход работы:

### 1. Представление словаря

```
>>> a = {'cat': 'кошка', 'dog': 'собака', 'bird': 'птица'}  
>>> a  
{'cat': 'кошка', 'dog': 'собака', 'bird': 'птица'}
```

### 2. Доступ к определенному элементу

```
>>> a['cat']  
'кошка'  
>>> a['bird']  
'птица'
```

### 3. Простейшие операции со словарями

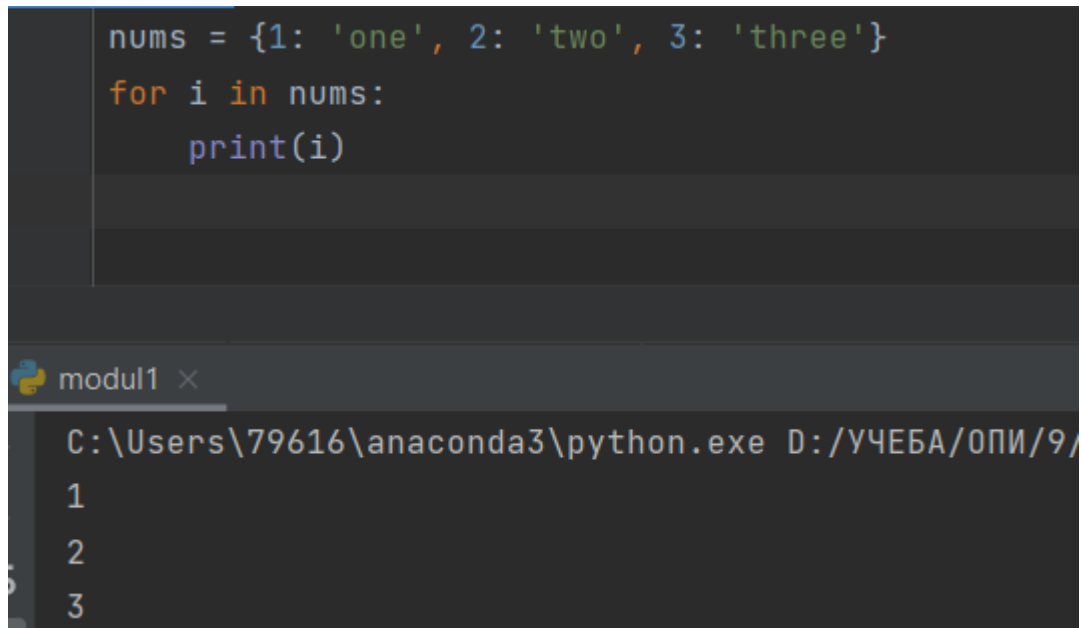
```
>>> a['elephant'] = 'бегемот'  
>>> a['table'] = 'стол'  
>>> a  
{'cat': 'кошка', 'dog': 'собака', 'bird': 'птица', 'elephant': 'бегемот', 'table': 'стол'}  
>>> a['elephant'] = 'слон'  
>>> del a['table']  
>>> a  
{'cat': 'кошка', 'dog': 'собака', 'bird': 'птица', 'elephant': 'слон'}
```

### 4. Словари с одинаковыми значениями, но с разными ключами

```
>>> nums = {1: 'one', 2: 'two', 3: 'three'}  
>>> person = {'name': 'Tom', 1: [30, 15, 16], 2: 2.34, ('ab', 100): 'no'}
```

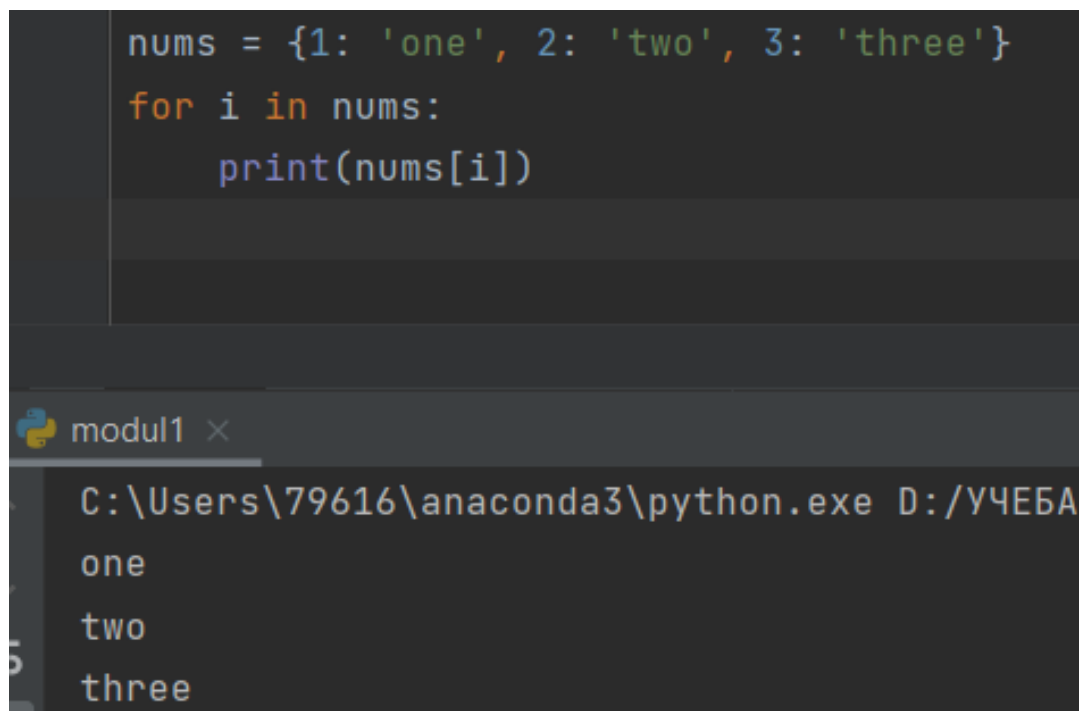
## 5. Перебор элементов словаря в цикле for

```
nums = {1: 'one', 2: 'two', 3: 'three'}
for i in nums:
    print(i)
```



The screenshot shows a Python IDE window titled 'modul1'. The code defines a dictionary 'nums' with keys 1, 2, and 3, each mapped to a string ('one', 'two', 'three'). A for loop iterates over the dictionary, printing each key. The output in the console is '1', '2', and '3' on separate lines.

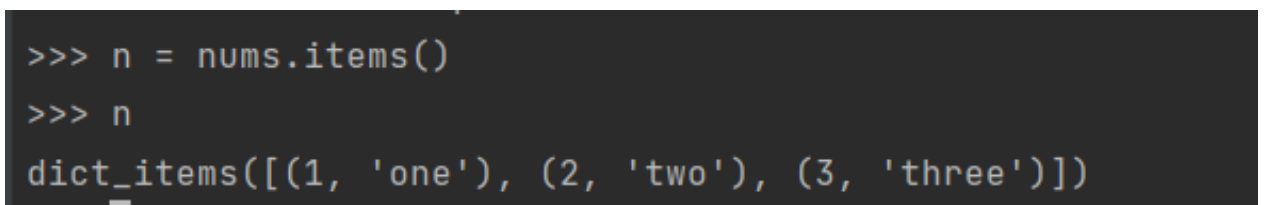
```
nums = {1: 'one', 2: 'two', 3: 'three'}
for i in nums:
    print(nums[i])
```



The screenshot shows a Python IDE window titled 'modul1'. The code defines a dictionary 'nums' with keys 1, 2, and 3, each mapped to a string ('one', 'two', 'three'). A for loop iterates over the dictionary, printing the value corresponding to each key. The output in the console is 'one', 'two', and 'three' on separate lines.

## 6. Метод items()

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



The screenshot shows a Python REPL session. The user enters 'n = nums.items()' and then 'n'. The output is 'dict\_items([(1, 'one'), (2, 'two'), (3, 'three')])'.

## 7. Извлечение ключа и его значения

```
nums = {1: 'one', 2: 'two', 3: 'three'}

for key, value in nums.items():
    print(key, 'is', value)
```

modul1 ×

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА

1 is one  
2 is two  
3 is three

## 8. Использование метода values()

```
nums = {1: 'one', 2: 'two', 3: 'three'}
v_nums = []
for v in nums.values():
    v_nums.append(v)
```

modul1 ×

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/9/

['one', 'two', 'three']

## 9. Метод clear() – удаление всех элементов словаря

```
>>> a
{'cat': 'кошка', 'dog': 'собака', 'bird': 'птица', 'elephant': 'слон'}
>>> a.clear()
>>> a
{}
```

## 10. Применение метода copy()

```
>>> nums2 = nums.copy()
>>> nums2[4] = 'four'
>>> nums
{1: 'one', 2: 'two', 3: 'three'}
>>> nums2
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

## 11. Использование метода fromkeys()

```
>>> a = [1, 2, 3]
>>> c = dict.fromkeys(a)
>>> c
{1: None, 2: None, 3: None}
>>> d = dict.fromkeys(a, 10)
>>> d
{1: 10, 2: 10, 3: 10}
>>> c
{1: None, 2: None, 3: None}
```

## 12. Получение элемента по его ключу – метод get()

```
>>> nums.get(1)
'one'
```

### 13. Результат выполнения метода pop() и popitem()

```
>>> nums.pop(1)
'one'
>>> nums
{2: 'two', 3: 'three'}
>>> nums.popitem()
(3, 'three')
>>> nums
{2: 'two'}
```

### 14. Добавление элемента в словарь с помощью метода setdefault()

```
>>> nums.setdefault(4, 'four')
'four'
>>> nums
{2: 'two', 4: 'four'}
```

### 15. Добавление в словарь другого словаря – метода update()

```
>>> nums.update({6: 'six', 7: 'seven'})
>>> nums
{2: 'two', 4: 'four', 6: 'six', 7: 'seven'}
```

### 16. Словарь включений

```
>>> {x: x * x for x in (1, 2, 3, 4)}
{1: 1, 2: 4, 3: 9, 4: 16}
>>> dict((x, x * x) for x in (1, 2, 3, 4))
{1: 1, 2: 4, 3: 9, 4: 16}
```

```
>>> {name: len(name) for name in ('Stack', 'Overflow', 'Exchange') if len(name) > 6}
{'Overflow': 8, 'Exchange': 8}
```

## 17.Объединение словарей

```
>>> dict1 = {'w':1, 'x': 1}
>>> dict2 = {'x': 2, 'y':2, 'z':2}
>>> {k: v for d in [dict1, dict2] for k, v in d.items()}
{'w': 1, 'x': 2, 'y': 2, 'z': 2}
```

## Выполнение индивидуального задания (вариант 13)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import datetime
from datetime import date

import sys

if __name__ == '__main__':
    workers = []

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

        if command == 'exit':
            break

        elif command == 'add':
            surname = input("Фамилия: ")
            name = input("Имя: ")
            number = int(input("Номер телефона: "))
            date_obj = input("Дата рождения: ").split('.')

            worker = {
                'surname': surname,
                'name': name,
                'number': number,
                'date_obj': date_obj,
            }

            workers.append(worker)
            if len(workers) > 1:
                workers.sort(key=lambda item: item.get('name', ''))

        elif command == 'list':
            for num, elem in enumerate(workers):
                print(f"{num+1}.\n{str(elem['surname'])}\n{str(elem['name'])}\n"
                    f"Номер телефона: {str(elem['number'])}\nДата рождения: {elem['date_obj']}")

        elif command.startswith('select'):
            surname = input("Введите фамилию: ")
            for elem in workers:
                if elem['surname'] == surname:
                    print(f"Имя {str(elem['name'])}\nНомер телефона: {str(elem['number'])}\n"
                        f"Дата рождения: {elem['date_obj']}")
                    break
            else:
                print("Фамилии не найдено")

        elif command == 'help':
            print("Список команд:\n")
            print("add - добавить человека;")
            print("list - вывести список всех людей;")
            print("select - найти данные по фамилии;")
            print("help - отобразить справку;")
            print("exit - завершить работу с программой.")

        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)
```



Список команд:

```
add - добавить человека;  
list - вывести список всех людей;  
select - найти данные по фамилии;  
help - отобразить справку;  
exit - завершить работу с программой.
```

Рисунок 1 – Вывод всех команд

```
>>> add  
Фамилия: petrov  
Имя: petr  
Номер телефона: 4555  
Дата рождения: 23.06.2005
```

Рисунок 2 – Ввод данных с помощью команды «add»

```
>>> list  
1.  
  lazareva darya  
Номер телефона: 8999  
Дата рождения: ['10', '11', '2002']  
2.  
  petrov petr  
Номер телефона: 4555  
Дата рождения: ['23', '06', '2005']
```

Рисунок 3 – Вывод всех людей, внесенные в список

```
>>> select
Введите фамилию: lazareva
Имя darya
Номер телефона: 8999
Дата рождения: ['10', '11', '2002']
```

Рисунок 4 – Результат вывода информации о пользователе по фамилии

```
>>> select
Введите фамилию: mirzaeva
Фамилии не найдено
```

Рисунок 5 – Результат написания фамилии, которой нет в списке

Ответы на 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. Словарь – изменяемый неупорядоченный набор элементов «ключ: значение»

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

Да, ведь Python Len() – встроенная функция, которая возвращает целое число, представляющее количество элементов в объекте, переданном ему аргументом. Функция Len (Dict) возвращает количество клавиш: пары значения в этом словаре

3. Какие методы обхода словарей Вам известны?

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

```
>>> for i in nums:
...     print(nums[i])
...
one
two
three
```

```
>>> for key, value in nums.items():
...     print(key, 'is', value)
...
1 is one
2 is two
3 is three
```

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

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

```
>>> for key, value in nums.items():
...     print(key, 'is', value)
...
1 is one
2 is two
3 is three
```

6. Что такое словарь включений?

Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка.

```
>>> {x: x * x for x in (1, 2, 3, 4)}
{1: 1, 2: 4, 3: 9, 4: 16}
```

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

Функция `zip()` в Python создает итератор, который объединяет элементы из нескольких источников данных. Эта функция работает со списками, кортежами, множествами и словарями для создания списков или кортежей, включающих все эти данные.

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. Так, он состоит из объектов следующих типов: `date` — хранит дату, `time` — хранит время.