

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
**ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**  
**ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Кафедра инфокоммуникаций**

**Отчет**  
**по лабораторной работе №4**  
**«Работа со словарями в языке Python»**  
**по дисциплине:**  
**«Введение в системы искусственного интеллекта»**

**Вариант 9**

Выполнил: студент группы ИВТ-б-о-18-1 (2)  
Полещук Константин Сергеевич

\_\_\_\_\_ (подпись)

Проверил:  
Воронкин Роман Александрович

\_\_\_\_\_ (подпись)

Ставрополь, 2022 г.

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

### **Задание №1**

**Использовать словарь, содержащий следующие ключи:** название начального пункта маршрута; название конечного пункта маршрута; номер маршрута. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по номерам маршрутов; вывод на экран информации о маршруте, номер которого введен с клавиатуры; если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

```
5 import os
6 #Список маршрутов
7 from os import sep
8 from turtle import clear
9
10
11 list_route = []
12
13 while True:
14
15     print("Заполнить список >> [1]")
16     print("Вывод списка >> [2]")
17     print("Выход >> [3]")
18
19     command = int(input(">>"))
20
21     if command == 1:
22
23         #Ввод данных с клавиатуры
24         start_point = input("Введите начальный пункт маршрута: ")
25         end_point = input("Введите конечный пункт маршрута: ")
26         number_route = int(input("Введите номер маршрута: "))
27
28         print("Маршрут добавлен!\n")
29
30         #Создание словаря "Маршрут"
31         route = {
32             'start_point': start_point,
33             'end_point': end_point,
34             'number_route': number_route
35         }
36
37         #Добавление маршрута в список
38         list_route.append(route)
39
40     elif command == 2:
41
42         if len(list_route) == 0:
43             print("Ваш список пуст! Сначала добавьте маршруты в список!\n")
44
45     else:
46         #Сортировка списка по номеру маршрутов
47         list_route.sort(key=lambda item: item.get('number_route'))
```

Рисунок 1.1 – Листинг программы

```

49         #Вывод
50         for i, route in enumerate(list_route, 1):
51             print(i,".",
52                   " Начальный пункт: ",route.get('start_point', ''),";",
53                   " Конечный пункт: ",route.get('end_point', ''),";",
54                   " Номер маршрута: ",route.get('number_route', 0),";",
55                   sep=''
56             )
57         print()
58
59
60     elif command == 3:
61         break
62     else:
63         print(f"Неизвестная команда: {command}\n")
64

```

Рисунок 1.2 – Листинг программы

```

Заполнить список >> [1]
Вывод списка >> [2]
Выход >> [3]
>>2
Ваш список пуст! Сначала добавьте маршруты в список!

Заполнить список >> [1]
Вывод списка >> [2]
Выход >> [3]
>>1
Введите начальный пункт маршрута: dw
Введите конечный пункт маршрута: df
Введите номер маршрута: 3
Маршрут добавлен!

Заполнить список >> [1]
Вывод списка >> [2]
Выход >> [3]
>>1
Введите начальный пункт маршрута: gr
Введите конечный пункт маршрута: ds
Введите номер маршрута: 1
Маршрут добавлен!

Заполнить список >> [1]
Вывод списка >> [2]
Выход >> [3]
>>2
1. Начальный пункт: gr; Конечный пункт: ds; Номер маршрута: 1;
2. Начальный пункт: dw; Конечный пункт: df; Номер маршрута: 3;

Заполнить список >> [1]
Вывод списка >> [2]
Выход >> [3]
>>3

```

Рисунок 2 – Результат программы

Файл Lab\_4 с решением находится на **Github**:  
<https://github.com/Scratchykaktus/Python.git>

### **Ответы на вопросы:**

1. Что такое словари в языке Python?

Словарь – это изменяемый тип данных. Следовательно, как и список он передается в функцию по ссылке. Поэтому иногда, чтобы избежать нежелательного изменения глобального словаря его копируют. Это делают и с другими целями.

Метод `fromkeys()` позволяет создать словарь из списка, элементы которого становятся ключами.

Применять метод можно как классу `dict`, так и к его объектам:

```
>>> 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}
```

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

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.

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

```
{'Exchange': 8, 'Overflow': 8}
```

Или переписать с помощью генераторного выражения.

```
>>> dict((name, len(name)) for name in ('Stack', 'Overflow', 'Exchange') if  
len(name) > 6)
```

```
{'Exchange': 8, 'Overflow': 8}
```

3. Какие методы обхода словарей Вам известны?

- 1) Если в цикле используются и ключи, и значения словаря, то нужно использовать метод `.items()`;
  - 2) Если в цикле используются только значения словаря, а ключи не важны, то нужно использовать метод `.values()`;
  - 3) Если в цикле нужны ключи словаря и ничего больше, то нужно использовать метод `.keys()`.
4. Какими способами можно получить значения из словаря по ключу?

Стандартный способ доступа к значению словаря – через квадратные скобки. Как видим, если ключ представлен в виде числа, то его пишем без кавычек. Если обратимся к несуществующему ключу, то получим в ответ ошибку `KeyError`.

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

В Python есть много встроенных **структур данных**, используемых для хранения разных типов информации. Словарь (`dict`) — одна из таких структур, которая хранит данные в формате пар ключ-значение. Получить доступ к значениям словаря Python можно с помощью ключей. Этот материал посвящен подробному обсуждению словаря.

Для создания словаря в Python необходимо передать последовательность элементов внутри фигурных скобок `{ }`, разделив их запятыми `( , )`. Каждый элемент имеет ключ и значение, выраженное парой «ключ: значение».

Значения могут быть представлять собой любые типы данных и повторяться, но ключи обязаны быть уникальными.

## 6. Что такое словарь включений?

Списковые включения в Python являются краткими синтаксическими конструкциями. Их можно использовать для создания списков из других списков, применяя функции к каждому элементу в списке.

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.

В Python есть несколько встроенных функций, которые позволяют перебирать данные. Одна из них — zip. Функция zip() в Python создает итератор, который объединяет элементы из нескольких источников данных.

У функции zip() множество сценариев применения. Например, она пригодится, если нужно создать набор словарей из двух массивов, каждый из которых содержит имя и номер сотрудника.

Предположим, что есть список имен и номером сотрудников, и их нужно объединить в массив кортежей. Для этого можно использовать функцию zip(). Вот пример программы, которая делает именно это:

```
employee_numbers = [2, 9, 18, 28]
employee_names = ["Дима", "Марина", "Андрей", "Никита"]

zipped_values = zip(employee_names, employee_numbers)
zipped_list = list(zipped_values)

print(zipped_list)
```

Функция zip возвращает следующее:

```
[('Дима', 2), ('Марина', 9), ('Андрей', 18), ('Никита', 28)]
```

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
- `datetime` — хранит дату и время