

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Институт цифрового развития**

**ОТЧЁТ**

**по лабораторной работе №2.6**

Дисциплина: «Программирование на Python»

Тема: «Работа со словарями в языке Python»

Выполнил: студент 2 курса  
группы ПИЖ-б-о-21-1  
Пуценко Иван Алексеевич

Ставрополь 2022



## 1. Выполнил задания.

Использовать словарь, содержащий следующие ключи: название пункта назначения рейса; номер рейса; тип самолета. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по возрастанию номера рейса; вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры; если таких рейсов нет, выдать на дисплей соответствующее сообщение.

```
if __name__ == '__main__':
    planes = list()
    while True:
        command = input('введите команду("help" - руководство по командам)\n>>>').lower()

        if command == 'exit':
            print("всего доброго")
            break

        elif command == 'help':
            print("add - добавление рейса\n"
                  "help - помощь по командам\n"
                  "select \"пункт назначения\" - вывод самолетов летящих в п.н.\n"
                  "exit - выход из программы")

        elif command == "add":
            plane = dict()
            plane["destination"] = input("Пункт назначения:\n")
            plane["flight_number"] = int(input("Номер рейса:\n"))
            plane["type_plane"] = input("Тип самолета\n")
            planes.append(plane)
            planes.sort(key=lambda key_plane: key_plane.get("flight_number"))
            print(planes)

        elif command == "select":
            lst = list(map(lambda x: x.get("destination"), planes))
            point = input('выберите нужное вам место\n')
            print("результаты поиска")
            if point in lst:
                print('рейсы в эту точку')
                for i in planes:
                    if point == i["destination"]:
                        print(f"{i['flight_number']}.....{i['type_plane']}")
            else:
                print("рейсов не найдено")

        elif command == "plane_print":
            for i in planes:
                print(f"куда - {i['destination']} номер - {i['flight_number']} самолет - {i['type_plane']}")
            else:
                print('нет такой команды')

if __name__ == '__main__':
    planes = list()
    while True:
        command = input('введите команду("help" - руководство по командам)\n>>>').lower()

        if command == 'exit':
```

```

        print("всего доброго")
        break

    elif command == 'help':
        print("add - добавление рейса\n"
              "help - помощь по командам\n"
              "select \"пункт назначения\" - вывод самолетов летящих в
п.н.\n"
              "exit - выход из программы")

    elif command == "add":
        plane = dict()
        plane["destination"] = input("Пункт назначения:\n")
        plane["flight_number"] = int(input("Номер рейса:\n"))
        plane["type_plane"] = input("Тип самолета\n")
        planes.append(plane)
        planes.sort(key=lambda key_plane: key_plane.get("flight_number"))
        print(planes)

    elif command == "select":
        lst = list(map(lambda x: x.get("destination"), planes))
        point = input('выберите нужное вам место\n')
        print("результаты поиска")
        if point in lst:
            print('рейсы в эту точку')
            for i in planes:
                if point == i["destination"]:
                    print(f"{i['flight_number']}.....{i['type_plane']}")
            else:
                print("рейсов не найдено")

    elif command == "plane_print":
        for i in planes:
            print(f"куда - {i['destination']} номер - {i['flight_number']}
самолет - {i['type_plane']}")

    else:
        print('нет такой команды')

```

## Контр. вопросы и ответы на них:

### 1. Что такое словари в языке Python?

Словари в Python – это изменяемые отображения ссылок на объекты, доступные по ключу.

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

Функция `len()` возвращает длину (количество элементов) в объекте. Аргумент может быть последовательностью, такой как строка, байты, кортеж, список или диапазон или коллекцией (такой как словарь, множество или неизменяемое множество).

### **3. Какие методы обхода словарей Вам известны?**

Самый очевидный вариант обхода словаря — это попытаться напрямую запустить цикл `for` по объекту словаря, так же как мы делаем это со списками, кортежами, строками и любыми другими итерируемыми объектами.

```
for something in currencies:
```

```
    print(something)
```

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

С помощью метода `.get()`

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

С помощью функции `dict.update()`

### **6. Что такое словарь включений?**

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

### **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.

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

Как получить текущие дату и время?

```
import datetime
dt_now = datetime.datetime.now()
print(dt_now)
```

Результат:

```
2022-09-11 15:43:32.249588
```

Получить текущую дату:

```
from datetime import date
current_date = date.today()
print(current_date)
```

Результат:

```
2022-09-11
```

Получить текущее время:

```
import datetime
current_date_time = datetime.datetime.now()
```

```
current_time = current_date_time.time()  
print(current_time)
```

**Результат:**

15:51:05.627643