

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 №2.6**  
**дисциплины «Программирование на Python»**

Выполнил:  
Магдаев Даламбек Магомедович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р.А., доцент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

## Тема: Работа со словарями в языке Python

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

### Порядок выполнения работы:

1. Создал новый репозиторий, клонировал его, в нем создал ветку developer и перешел на нее.
2. Проработал пример лабораторной работы:

```
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
>>> list
+-----+-----+-----+-----+
| № |           Ф.И.О.           |      Должность      |      Год      |
+-----+-----+-----+-----+
>>> add
Фамилия и инициалы? Магдаев Д.М.
Должность? Студент
Год поступления? 2022
>>> Иванов И.И.
Неизвестная команда иванов и.и.
>>> add
Фамилия и инициалы? Иванов И.И.
Должность? Директор
Год поступления? 2005
>>> list
+-----+-----+-----+-----+
| № |           Ф.И.О.           |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Иванов И.И.                |      Директор       |      2005     |
+-----+-----+-----+-----+
|  2 | Магдаев Д.М.               |      Студент        |      2022     |
+-----+-----+-----+-----+
>>> select 10
1: Иванов И.И.
>>>
```

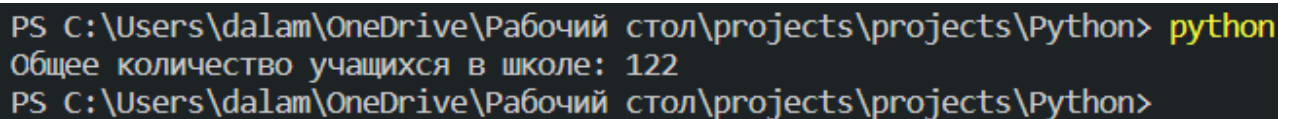
Рисунок 1. Несколько запусков программы примера

3. Решил задачу 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.

### Код программы:

```
#!/usr/bin/env
python3# -*- coding:
utf-8 -*-
if __name__ == "__main__":
    school = {
        '1a': 25,
        '1б': 27,
        '2б': 20,
        '6a': 30,
        '7в': 22}
    # а) изменилось количество учащихся в одном из классов
    school['1a'] = 26
    # б) появился новый класс
    school['8г'] = 18
    # в) расформирован (удален) другой класс
    del school['2б']
    total_students = sum(school.values())
    print("Общее количество учащихся в школе:", total_students)
```



```
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python> python
Общее количество учащихся в школе: 122
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python>
```

Рисунок 2. Вывод программы task\_1

4. Решил задачу 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.

### Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    # Создание исходного словаря
    original_dict = {1: "one", 2: "two", 3: "three"}
    reversed_dict = {v: k for k, v in original_dict.items()}
    print("Исходный словарь:", original_dict)
    print("Новый \"обратный\" словарь:", reversed_dict)
```

```
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python> python
Исходный словарь: {1: 'one', 2: 'two', 3: 'three'}
Новый "обратный" словарь: {'one': 1, 'two': 2, 'three': 3}
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python>
```

Рисунок 3. Вывод программы task\_2

5. Выполнил индивидуальное задание вариант 9: 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.

#### **Код программы:**

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    routes = []
    while True:
        command = input(">>> ").lower()
        line = '+-{}-+-{}-+-{}-+'.format(
            '_' * 30,
            '_' * 20,
            '_' * 8
        )
        text = (
            '| {:^30} | {:^20} | {:^8} |'.format(
                "Начало",
                "Конец",
                "Номер"
            )
        )
        if command == 'exit':
            break
        elif command == 'add':
            start = input("Введите начальный пункт: ")
            end = input("Введите конечный пункт: ")
            count = int(input("Введите номер маршрута: "))
            route = {
```

```

'начальный пункт': start,
'конечный пункт': end,
'номер маршрута': count
}
if route not in routes:
bisect.insort(
routes, route, key=lambda item: item.get('номер маршрута'))
else:
print("данный маршрут уже добавлен")
elif command == 'list':
print(line)
print(text)
print(line)
for route in routes:
print(
'| {:<30} | {:<20} | {:>8} |'.format(
route.get('начальный пункт', ""),
route.get('конечный пункт', ""),
route.get('номер маршрута', "")
)
)
print(line)
elif (m := re.match(r'select (.+)', command)):
name_punct = m.group(1)
found = False
for route in routes:
if str(route['номер маршрута']) == name_punct:
if not found:
print(line)
print(text)
print(line)
print(
'| {:<30} | {:<20} | {:>8} |'.format(
route.get('начальный пункт', ""),
route.get('конечный пункт', ""),
route.get('номер маршрута', "")
)
)
found = True
if not found:
print("Маршрутов с данным номером не найдено.")
else:
print(line)
elif command == 'help':
# Вывести справку о работе с программой.

```

```

print("Список команд:\n")
print("add - добавить маршрут;")
print("list - вывести список маршрутов;")
print("select <номер маршрута> - запросить маршруты, которые имеют
данный номер")
print("help - отобразить справку;")
print("exit - завершить работу с программой.")
else:
print(f"Неизвестная команда {command}", file=sys.stderr)

```

```

PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Python> python -u "c:\Users\dalam\OneDrive\
>>> add
Введите начальный пункт: A
Введите конечный пункт: Б
Введите номер маршрута: 3
>>> add
Введите начальный пункт: Б
Введите конечный пункт: A
Введите номер маршрута: 5
>>> add
Введите начальный пункт: В
Введите конечный пункт: A
Введите номер маршрута: 3
>>> list
+-----+-----+-----+
|          Начало          |          Конец          |          Номер          |
+-----+-----+-----+
| A                        | Б                        | 3                       |
| В                        | А                        | 3                       |
| Б                        | А                        | 5                       |
+-----+-----+-----+
>>> select 3
+-----+-----+-----+
|          Начало          |          Конец          |          Номер          |
+-----+-----+-----+
| A                        | Б                        | 3                       |
| В                        | А                        | 3                       |
+-----+-----+-----+

```

Рисунок 4. Несколько запусков программы индивидуального задания

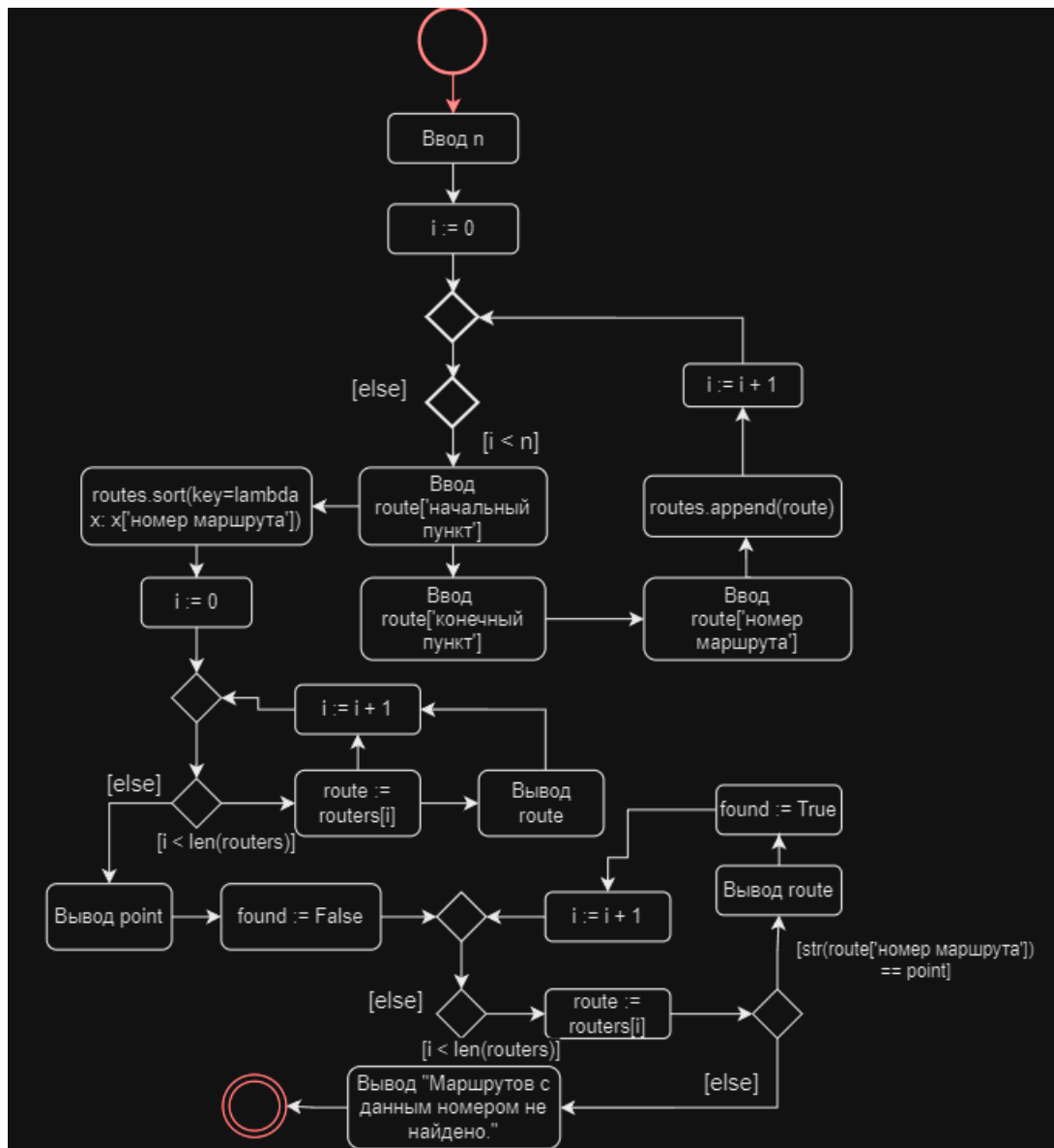


Рисунок 5. UML-диаграмма индивидуального задания

## 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, и выведет количество ключей.

3. Какие методы обхода словарей Вам известны?

Цикл for, методы items(), keys() и values().

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

В словаре доступ к значениям осуществляется по ключам, которые заключаются в квадратные скобки или с помощью метода get().

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

С помощью setdefault() можно добавить элемент в словарь. С помощью квадратных скобок: `my_dict['a'] = 1`

С помощью метода update(): `my_dict.update({'a': 1})`

6. Что такое словарь включений?

Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка (краткая синтаксическая конструкция, предназначенная для создания словаря).

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

Функция zip() принимает итерируемый объект, например, список, кортеж, множество или словарь в качестве аргумента. Затем она генерирует список кортежей, которые содержат элементы из каждого объекта, переданного в функцию.

Пример:

```
keys = ['a', 'b', 'c'] values = [1, 2, 3]
```

```
my_dict = dict(zip(keys, values)) print(my_dict)
```

Данный код выведет: {'a': 1, 'b': 2, 'c': 3}



8. Самостоятельно изучите возможности модуля `datetime`. Каким функционалом по работе с датой и временем обладает этот модуль?

Модуль `datetime` в Python позволяет работать с датой и временем, предоставляя классы и функции для работы с текущей датой и временем, форматирования даты и времени, а также вычисления разницы между двумя датами или временем с помощью класса `timedelta`.

- `date` – хранит дату
- `time` – хранит время
- `datetime` – хранит дату и время

**Вывод:** в результате выполнения работы были приобретены навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.