

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.6

Работа со словарями в языке Python3.

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Малышев А.Ю. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверила Воронкин Р.А. _____

(подпись)

Ставрополь 2021

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

Ход работы: <https://github.com/AlexandrM333/labrab2.6.git>

Ссылка на репозиторий:

Выполнил пример из лабораторной:

```
primer
C:\Games\envs\labrab2.6\python.exe C:/Users/Александр/projects/labrab2.6/primer.py
>>> add
Фамилия и инициалы? Грушин М.В.
Должность? Охранник
Год поступления? 2019
>>> add
Фамилия и инициалы? Петров И.И.
Должность? Секретарь
Год поступления? 2015
>>> add
Фамилия и инициалы? Распутин С.А.
Должность? Слесарь
Год поступления? 2009
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          | Должность | Год |
+-----+-----+-----+-----+
| 1 | Грушин М.В.             | Охранник  | 2019 |
| 2 | Петров И.И.             | Секретарь  | 2015 |
| 3 | Распутин С.А.           | Слесарь    | 2009 |
+-----+-----+-----+-----+
>>> 10
>>> Неизвестная команда 10
select 10
1: Распутин С.А.
>>>
```

Рисунок 1. Пример задания.

2. Выполнил две общих задания:

```
C:\Games\envs\labrab2.6\python.exe C:/Users/Александр/projects/labrab2.6/z1.py
>>> change
Выберите класс: 00
Новое количество учащихся?: 15
>>> add
Новый класс: 3ю
Количество учащихся?: 14
>>> delete
Выберите класс: 10н
Класс удален
>>> total
437 учащихся
>>> list
10г: 15
11а: 17
11б: 15
1а: 23
1б: 22
2а: 19
2б: 20
3в: 22
3г: 20
3ю: 14
4а: 23
4г: 24
5б: 19
5в: 18
```

Рисунок 2. Первое задание

```
C:\Games\envs\labrab2.6\python.exe C:/Users/Александр/projects/labrab2.6/z2.py
{'ball': 1, 'bird': 2, 'three': 3, 'mouse': 4, 'cat': 5, 'dog': 6, 'blink': 7}

Process finished with exit code 0
```

Рисунок 3. Второе задание.

3. Выполнил индивидуальное задание:

```
C:\Games\envs\labrab2.6\python.exe C:/Users/Александр/projects/labrab2.6/indiv.py
>>> add
Пункт назначения: Москва
Номер поезда: 5
Время отправления: 11:15
>>> add
Пункт назначения: Воронеж
Номер поезда: 7
Время отправления: 14:20
>>> add
Пункт назначения: Ставрополь
Номер поезда: 1
Время отправления: 17:00
>>> list
+-----+-----+-----+
| Пункт назначения | № | Время |
+-----+-----+-----+
| Воронеж         | 7 | 14:20 |
| Москва          | 5 | 11:15 |
| Ставрополь      | 1 | 17:00 |
+-----+-----+-----+
>>> select
Выберите номер маршрута: 5
    №5 Москва 11:15
>>> select
Выберите номер маршрута: 1
    №1 Ставрополь 17:00
```

Рисунок 4. Индивидуальное задание.

4. Проверил задания на flake8:

```
Anaconda Powershell Prompt (Games)
(base) PS C:\Users\Александр> cd projects
(base) PS C:\Users\Александр\projects> cd labrab2.6
(base) PS C:\Users\Александр\projects\labrab2.6> flake8
(base) PS C:\Users\Александр\projects\labrab2.6>
```

Рисунок 5. Проверка заданий.

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?

Можно, при этом функция `len()` предоставит количество связей ключ+значение.

3. Какие методы обхода словарей Вам известны?

Элементы словаря перебираются в цикле `for` также, как элементы других сложных объектов. Однако "по умолчанию" извлекаются только ключи. Но по ключам всегда можно получить значения.

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

В словаре доступ к значениям осуществляется по ключам, которые заключаются в квадратные скобки: `dict[key]`. Также метод `get()` позволяет получить элемент по его ключу: `dict.get(key)`.

Равносильно `dict[key]`, если ключ присутствует в словаре. Если ключ отсутствует в словаре, то выражение `словарь[ключ]` приведет к возникновению исключительной ситуации, тогда как выражение `словарь.get(ключ, значение)` в этом случае вернет значение, по умолчанию значение равно `None`.

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

Конструкцией `dict[key] = value` можно установить новую связку ключ+значение. Также с помощью `setdefault()` можно добавить элемент в словарь: `dict.setdefault(key, value)`

6. Что такое словарь исключений?

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.

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.

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. 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.

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

Модуль `datetime` предоставляет классы для обработки времени и даты разными способами. С её помощью можно получить текущую дату, найти разницу между датами, получить старые и будущие даты, проводить операции над датами.

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