

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

Институт цифрового развития
Кафедра инфокоммуникаций

Отчет по лабораторной работе №6

По дисциплине основы кроссплатформенного программирования
«Исследования основных возможностей Git и GitHub»

Выполнила:

студентк группы ИТС-б-о-21-1

Яхшибоев Элёр Содикжон угли

Проверил:доцент кафедры

инфокоммуникаций

Воронкин Р. А.

Работа защищена с оценкой:

(подпись)

Ставрополь, 2022

Тема: Работа со словарями в языке Python

Цель работы: исследование процесса установки и базовых возможностей языка Python версии 3.x.

Задание №1.

```
C: > Users > Admin > Documents > GitHub > 7 > 7.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      # Список .
8      train = []
9      # Организовать бесконечный цикл запроса команд.
10     while True:
11         # Запросить команду из терминала.
12         command = input(">>> ").lower()
13         # Выполнить действие в соответствие с командой.
14         if command == 'exit':
15             break
16         elif command == 'add':
17             # Запросить данные .
18             race = input("Название пункта выезда ")
19             race1 = input("Название пункта прибытия ")
20             number = input("Номер маршрута ")
21
22             # Создать словарь.
23             trains = {
24                 'race': race,
25                 'number': number,
26                 'race1': race1,
27             }
28             # Добавить словарь в список.
29             train.append(trains)
30             # Отсортировать список в случае необходимости.
31             if len(train) > 1:
```

```

32         train.sort(key=lambda item: item.get('race', ''))
33     elif command == 'list':
34         # Заголовок таблицы.
35         line = '+-{}-+-{}-+-{}-+-{}-+'.format(
36             '-' * 4,
37             '-' * 30,
38             '-' * 20,
39             '-' * 20
40         )
41         print(line)
42         print(
43             '| {:^4} | {:^30} | {:^20} | {:^20} |'.format(
44                 "No",
45                 "Пункт выезда",
46                 "Пункт прибытия",
47                 "Номер"
48             )
49         )
50     )
51     print(line)
52     # Вывести данные о всех рейсах.
53     for idx, trains in enumerate(train, 1):
54         print(
55             '| {:>4} | {:<30} | {:<20} | {:>20} |'.format(
56                 idx,
57                 trains.get('race', ''),
58                 trains.get('race1', ''),
59                 trains.get('number', '')
60             )
61         )

```

```

60
61     )
62 )
63 print(line)
64 elif command.startswith('select '):
65     parts = command.split(' ', maxsplit=2)
66     sel = (parts[1])
67     count = 0
68     for trains in train:
69         if trains.get('race') == sel:
70             count += 1
71             print(
72                 '{:>4}: {}'.format(count, trains.get('race', ''))
73             )
74             print("Номер маршрута:", trains.get('number', ''))
75
76     # Если счетчик равен 0, то рейсы не найдены.
77     if count == 0:
78         print("Маршрут не найден.")
79 elif command == 'help':
80     # Вывести справку о работе с программой.
81     print("Список команд:\n")
82     print("add - добавить маршрут;")
83     print("list - вывести список маршрутов;")
84     print("select <товар> - информация о маршруте;")
85     print("help - отобразить справку;")
86     print("exit - завершить работу с программой.")
87 else:
88     print("Неизвестная команда {command}", file=sys.stderr)
89

```

1. Что такое словари в языке Python?

Ответ: 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?

Ответ: Да может! Функция len() возвращает длину (количество элементов) в объекте.

3. Какие методы обхода словарей Вам известны?

Ответ: У словаря как класса есть метод items(), который создает особую структуру, состоящую из кортежей. Каждый кортеж включает ключ и значение:

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

Методы словаря keys() и values() позволяют получить отдельно перечни ключей и значений. Так что если, например, надо перебрать только значения или только ключи, лучше воспользоваться одним из этих методов:

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

Так де существуют методы clear(), copy(), fromkeys(), get(), pop(), popitem(), setdefault(), update(). Метод clear() удаляет все элементы словаря, но не удаляет сам словарь. В итоге остается пустой Словарь. Метод fromkeys() позволяет создать словарь из списка, элементы которого становятся ключами. Применять метод можно как классу dict, так и к его объектам. Метод get() позволяет получить элемент по его ключу. Метод pop() удаляет из словаря элемент по указанному ключу и возвращает значение удаленной пары. Метод popitem() не принимает аргументов, удаляет и возвращает произвольный элемент. С помощью setdefault() можно добавить элемент в словарь. С помощью update() можно добавить в словарь другой словарь

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

Ответ: Операция `dict[key]` вернет элемент словаря `dict` с ключом `key`. Операция вызывает исключение `KeyError`, если ключ `key` отсутствует в словаре.

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

Ответ: Операция `d[key] = value` добавит в словарь `dict` новый элемент - пару ключ-значение. Если в словаре существует ключ `key` то эта операция присвоит ключу `key` новое значение `value`.

6. Что такое словарь включений?

Ответ: 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.

Ответ: Функция `zip()` создает итератор кортежей, который объединяет элементы каждой из переданных последовательностей `*iterables`.

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

Ответ: `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:

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

Вывод: в ходе выполнения работ я приобрел навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3