

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФГАОУ высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

**Отчет по лабораторной работе №7
Работа со словарями в языке Python**

Выполнил студент группы

ИТС-б-з-22-1

Потеев А.С. « » _____ 2023г.

Подпись студента _____

Работа защищена « » _____ 2023г.

Проверил доцент, кандидат технических
наук, доцент кафедры инфокоммуникаций

Воронкин Роман Александрович

(подпись)

Ставрополь, 2023 г.

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

Ход работы:

Ссылка на репозиторий – (<https://github.com/poteevprog/Lab7>)

Пример 1. Использовать словарь, содержащий следующие ключи: фамилия и инициалы работника; название занимаемой должности; год поступления на работу. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в список, состоящий из заданных словарей;
- записи должны быть размещены по алфавиту;
- вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;
- если таких работников нет, вывести на дисплей соответствующее сообщение.

```
primer1.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5  from datetime import date
6
7  if __name__ == '__main__':
8      # Список работников.
9      workers = []
10
11     # Организовать бесконечный цикл запроса команд.
12     while True:
13         # Запросить команду из терминала.
14         command = input(">>> ").lower()
15
16         # Выполнить действие в соответствие с командой.
17         if command == 'exit':
18             break
19
20         elif command == 'add':
21             # Запросить данные о работнике.
22             name = input("Фамилия и инициалы? ")
23             post = input("Должность? ")
24             year = int(input("Год поступления? "))
25
26             # Создать словарь.
27             worker = {
28                 'name': name,
29                 'post': post,
30                 'year': year,
31             }
32
33             # Добавить словарь в список.
34             workers.append(worker)
35             # Отсортировать список в случае необходимости.
36             if len(workers) > 1:
37                 workers.sort(key=lambda item: item.get('name', ''))
38
39         elif command == 'list':
40             # Заголовок таблицы.
41             line = '+-{}-+-{}-+-{}-+-{}-+'.format(

```

```

42         '-' * 4,
43         '-' * 30,
44         '-' * 20,
45         '-' * 8
46     )
47     print(line)
48     print(
49         '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
50             "No",
51             "Ф.И.О.",
52             "Должность",
53             "Год"
54         )
55     )
56     print(line)
57
58     # Вывести данные о всех сотрудниках.
59     for idx, worker in enumerate(workers, 1):
60         print(
61             '| {:>4} | {:<20} | {:<15} | {:<11} |'.format(
62                 idx,
63                 worker.get('name', ''),
64                 worker.get('post', ''),
65                 worker.get('year', 0)
66             )
67         )
68     print(line)
69
70     elif command.startswith('select '):
71         # Получить текущую дату.
72         today = date.today()
73
74         # Разбить команду на части для выделения номера года.
75         parts = command.split(' ', maxsplit=1)
76         # Получить требуемый стаж.
77         period = int(parts[1])
78
79         # Инициализировать счетчик.
80         count = 0
81
82         # Проверить сведения работников из списка.

```

```

82         for worker in workers:
83             if today.year - worker.get('year', today.year) >= period:
84                 count += 1
85                 print(
86                     '{:>4}: {}'.format(count, worker.get('name', ''))
87                 )
88
89         # Если счетчик равен 0, то работники не найдены.
90         if count == 0:
91             print("Работники с заданным стажем не найдены.")
92
93     elif command == 'help':
94         # Вывести справку о работе с программой.
95         print("Список команд:\n")
96         print("add - добавить работника;")
97         print("list - вывести список работников;")
98         print("select <стаж> - запросить работников со стажем;")
99         print("help - отобразить справку;")
100        print("exit - завершить работу с программой.")
101    else:
102        print(f"Неизвестная команда {command}", file=sys.stderr)
103

```

```

82         for worker in workers:
83             if today.year - worker.get('year', today.year) >= period:
84                 count += 1
85                 print(
86                     '{:>4}: {}'.format(count, worker.get('name', ''))
87                 )
88
89         # Если счетчик равен 0, то работники не найдены.
90         if count == 0:
91             print("Работники с заданным стажем не найдены.")
92
93     elif command == 'help':
94         # Вывести справку о работе с программой.
95         print("Список команд:\n")
96         print("add - добавить работника;")
97         print("list - вывести список работников;")
98         print("select <стаж> - запросить работников со стажем;")
99         print("help - отобразить справку;")
100        print("exit - завершить работу с программой.")
101    else:
102        print(f"Неизвестная команда {command}", file=sys.stderr)
103

```

Рисунок 1 – Окно программы примера 1

Индивидуальное задание 1.

12. Использовать словарь, содержащий следующие ключи: фамилия, имя; номер телефона; дата рождения (список из трех чисел). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть размещены по алфавиту; вывод на экран информации о людях, чьи дни рождения приходятся на месяц, значение которого введено с клавиатуры; если таких нет, выдать на дисплей соответствующее сообщение.

```
primer1.py  individ1.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      # Список работников.
8      people = []
9
10     # Организовать бесконечный цикл запроса команд.
11     while True:
12         # Запросить команду из терминала.
13         command = input(">>> ").lower()
14
15         # Выполнить действие в соответствие с командой.
16         if command == 'exit':
17             break
18
19         elif command == 'add':
20             # Запросить данные о человеке.
21             name = input("Фамилия, Имя: ")
22             number = input("Номер телефона: ")
23             birthday = input("Дата рождения: ")
24
25             # Создать словарь.
26             human = {
27                 'name': name,
28                 'number': number,
29                 'birthday': birthday,
30             }
31
32             # Добавить словарь в список.
33             people.append(human)
34             # Отсортировать список в случае необходимости.
35             if len(people) > 1:
36                 people.sort(key=lambda item: item.get('name', ''))
37
38         elif command == 'list':
39             # Заголовок таблицы.
```

```

40 line = '+-{}--{}-+-{}--{}-+'.format(
41     '-' * 7,
42     '-' * 30,
43     '-' * 20,
44     '-' * 20
45 )
46 print(line)
47 print(
48     '| {:^7} | {:^30} | {:^20} | {:^20} |'.format(
49         "№",
50         "Фамилия, Имя",
51         "Номер телефона",
52         "Год рождения"
53     )
54 )
55
56 print(line)
57
58 # Вывести данные о всех людях.
59 for idx, human in enumerate(people, 1):
60     print(
61         '| {:>7} | {:<30} | {:<20} | {:<20} |'.format(
62             idx,
63             human.get('name', ''),
64             human.get('number', ''),
65             human.get('birthday', 0)
66         )
67     )
68
69 print(line)
70
71 elif command.startswith('select '):
72     # Разбить команду на части для выделения Фамилии.
73     parts = command.split(' ', maxsplit=1)
74     # Получить требуемый стаж.
75     month = parts[1]
76     search_month = []

```



```

78     # Проверить сведения работников из списка.
79     for human in people:
80         dr = human["birthday"].split(" ")[1].lower()
81         if dr == month:
82             search_month.append(human)
83
84     if len(search_month) > 0:
85         line_new = '+-{}-+-{}-+-{}-+-{}-+'.format(
86             '-' * 7,
87             '-' * 30,
88             '-' * 20,
89             '-' * 20
90         )
91         print(line_new)
92
93         print(
94             '| {:^7} | {:^30} | {:^20} | {:^20} | '.format(
95                 "№",
96                 "Фамилия, Имя",
97                 "Номер телефона",
98                 "Дата рождения"
99             )
100         )
101         print(line_new)
102
103     for idx_new, spisok_new in enumerate(search_month, 1):
104         print(
105             '| {:>7} | {:<30} | {:<20} | {:<20} | '.format(
106                 idx_new,
107                 spisok_new.get('name', ''),
108                 spisok_new.get('number', ''),
109                 spisok_new.get('birthday', '')
110             )
111         )
112
113     print(line_new)

```

```

115         else:
116             print("Работники с заданной датой не найдены.")
117
118     elif command == 'help':
119         # Вывести справку о работе с программой.
120         print("Список команд:\n")
121         print("add - добавить человека;")
122         print("list - вывести список людей;")
123         print("select <месяц> - запросить данные о человеке;")
124         print("help - отобразить справку;")
125         print("exit - завершить работу с программой.")
126     else:
127         print(f"Неизвестная команда {command}", file=sys.stderr)
128

```

```

"C:\Users\user\Desktop\учеба\1 курс\2 семестр\Основы кроссплатформенного программирования\Lab7prog\
>>> add
Фамилия, Имя: Пупкин Вася
Номер телефона: 65767889066
Дата рождения: 12 март 2003
>>> add
Фамилия, Имя: Котлетова Анфиса
Номер телефона: 65780123454
Дата рождения: 20 февраль 1999
>>> add
>>> Неизвестная команда add
add
Фамилия, Имя: Замахов Алексей
Номер телефона: 54654356435
Дата рождения: 20 март 2009
>>> list
+-----+-----+-----+-----+
|  №   | Фамилия, Имя | Номер телефона | Год рождения |
+-----+-----+-----+-----+
|    1 | Замахов Алексей | 54654356435 | 20 март 2009 |
|    2 | Котлетова Анфиса | 65780123454 | 20 февраль 1999 |
|    3 | Пупкин Вася | 65767889066 | 12 март 2003 |
+-----+-----+-----+-----+
>>> select март
+-----+-----+-----+-----+
|  №   | Фамилия, Имя | Номер телефона | Дата рождения |
+-----+-----+-----+-----+
|    1 | Замахов Алексей | 54654356435 | 20 март 2009 |
|    2 | Пупкин Вася | 65767889066 | 12 март 2003 |
+-----+-----+-----+-----+
>>> |

```

Рисунок 2 – Окно программы индивидуального задания 1

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! Функция *len()* возвращает длину (количество элементов) в объекте.

3. Какие методы обхода словарей Вам известны?

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 же 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?

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?

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?

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()` 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?

`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: Я на лабораторной работе приобрел навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.