

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**«Работа со словарями в языке Python»**

**ОТЧЕТ**  
**по лабораторной работе №9**  
**дисциплины**  
**«Основы программной инженерии»**

Выполнил:

Сотников Андрей Александрович  
2 курс, группа ПИЖ-б-о-21-1,  
09.03.04 «Программная  
инженерия», направленность  
(профиль) «Разработка и  
сопровождение программного  
обеспечения», очная форма  
обучения

---

(подпись)

Проверил:

---

(подпись)

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

Ставрополь, 2022 г.

## Проработка примера из лабораторной работы:

```
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 соответствии 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.
```

Рисунок 1.1 – Код примера

```

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                 "№",
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} | {:<30} | {:<20} | {:>8} |'.format(
62                     idx,
63                     worker.get('name', ''),
64                     worker.get('post', ''),
65                     worker.get('year', 0)
66                 )
67             )
68         print(line)
69

```

Рисунок 1.2 – Код примера

```

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     # 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.")

```

Рисунок 1.3 – Код примера

```

102 else:
103     print(f"Неизвестная команда {command}", file=sys.stderr)
104

```

Рисунок 1.4 – Код примера

```

>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Фамилия и инициалы? Евгенийев И.В.
Должность? Директор
Год поступления? 1944
>>> add
Фамилия и инициалы? Мизин Г.Е.
Должность? Попуск
Год поступления? 2021
>>> list
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |   Год   |
+-----+-----+-----+-----+
|  1 | Евгенийев И.В.          | Директор            |  1944   |
|  2 | Мизин Г.Е.             | Попуск              |  2021   |
+-----+-----+-----+-----+
>>> select 1
    1: Евгенийев И.В.
    2: Мизин Г.Е.
>>> seect 1
>>> Неизвестная команда seect 1
exit

Process finished with exit code 0

```

Рисунок 1.4 – Результат работы примера

**Задание №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.

```
1  ▶ #!/usr/bin/env python3
2  ▶ #- coding: utf-8 -*-
3
4  ▶ if __name__ == '__main__':
5
6      # Source Dictionary
7      school = {
8          '1a': 22,
9          '1б': 23,
10         '1б': 25,
11         '3a': 30,
12         '3e': 24,
13         '7в': 27,
14         '10в': 17,
15         '9a': 31,
16         '3д': 24,
17     }
18
19     # Infinite Command Request Loop
20     while True:
21         command = input("Enter command:")
22         if command == "change":
23             while True:
24                 cc = input("Enter class you want change the num of students: ")
25                 nn = input("Enter the number of students students: ")
26                 if cc in school:
27                     school[cc] = nn
28                     break
29                 else:
30                     print("This class does not exist")
31             print(school)
32
33         elif command == "delete":
34             while True:
35                 cc = input("Enter the disbanded class: ")
36                 if cc in school:
37                     school.pop(cc)
38                     break
39                 else:
40                     print("This class does not exist")
41             print(school)
42
43         elif command == "add":
44             while True:
45                 cc = input("Enter new class: ")
46                 nn = input("Enter number of students: ")
47                 if cc in school:
48                     print("This class already exist")
49                 else:
```

Рисунок 2.1 – Код программы задания №1

```
elif command == "delete":
    while True:
        cc = input("Enter the disbanded class: ")
        if cc in school:
            school.pop(cc)
            break
        else:
            print("This class does not exist")
    print(school)

elif command == "add":
    while True:
        cc = input("Enter new class: ")
        nn = input("Enter number of students: ")
        if cc in school:
            print("This class already exist")
        else:
            school[cc] = nn
        break
    print(school)

elif command == "count":
    summa = 0
    for i in school:
        summa += int(school.get(i))
    print(summa)

elif command == "exit":
    break

else:
    print("Wrong command!")
```

Рисунок 2.2 – Код программы задания №1

```

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\labor-9\PyCharm\1_task.py
Enter command:change
Enter class you want change the num of students: 1a
Enter the number of students students: 33
{'1a': '33', '1б': 23, '1в': 25, '3a': 30, '3e': 24, '7в': 27, '10в': 17, '9a': 31, '3д': 24}
Enter command:delete
Enter the disbanded class: 10в
{'1a': '33', '1б': 23, '1в': 25, '3a': 30, '3e': 24, '7в': 27, '9a': 31, '3д': 24}
Enter command:add
Enter new class: 10г
Enter number of students: 15
{'1a': '33', '1б': 23, '1в': 25, '3a': 30, '3e': 24, '7в': 27, '9a': 31, '3д': 24, '10г': '15'}
Enter command:exit

```

Рисунок 2.4 – Результат работы кода задания №1

**Задание №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.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5
6      # Dictionary with keys "numbers" and values "words"
7      first_dict = {
8          1: 'one',
9          2: 'two',
10         3: 'three',
11         4: 'four',
12         5: 'five'
13     }
14
15     # Using the .items function
16     dict_items = first_dict.items()
17
18     # Creating a new dictionary in which keys and values are changed
19     new_dict = {v: k for k, v in dict_items}
20
21     # Output
22     print(new_dict)
23

```

2\_task × Individual ×

```

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\la
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}

```

Рисунок 3 – Код и результат работы программы задания №2



**Индивидуальное задание:** 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 упорядочены по трем первым цифрам номера телефона; вывод на экран информации о человеке, чья фамилия введена с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
from datetime import date

if __name__ == '__main__':
    # Список работников.
    people = []

    # Организовать бесконечный цикл запроса команд.
    while True:
        # Запрос команды
        command = input(">>> ").lower()

        # Выполнить действие в соответствие с командой.
        if command == 'exit':
            break

        elif command == 'add':
            # Запросить данные о работнике.
            name = input("Фамилия и инициалы? ")
            post = input("Телефон? ")
            year = int(input("Год рождения? "))

            # Создать словарь.
            man = {
                'name': name,
                'tel': post,
                'date': year,
            }

            # Добавить словарь в список.
            people.append(man)
            # Отсортировать список в случае необходимости.
            if len(people) > 1:
                people.sort(key=lambda item: item.get('tel', ''))

        elif command == 'list':
            # Заголовок таблицы.
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                '-' * 4,
                '-' * 30,
                '-' * 20,
                '-' * 12
            )
```

Рисунок 4.1 – Код программы индивидуального задания

```

# Заголовок таблицы.
line = '+-{}-+-{}-+-{}-+-{}-+'.format(
    '-' * 4,
    '-' * 30,
    '-' * 20,
    '-' * 12
)
print(line)
print(
    '| {:^4} | {:^30} | {:^20} | {:^12} |'.format(
        "№",
        "Ф.И.О.",
        "Телефон",
        "Год рождения"
    )
)
print(line)

# Вывести данные о всех сотрудниках.
for idx, man in enumerate(people, 1):
    print(
        '| {:>4} | {:<30} | {:<20} | {:>12} |'.format(
            idx,
            man.get('name', ''),
            man.get('tel', ''),
            man.get('date', 0)
        )
    )
print(line)

elif command.startswith('select'):

    # Разбить команду на части.
    parts = command.split(' ', maxsplit=1)
    # Получить имя.
    period = parts[1]
    count = 0
    # Проверить сведения работников из списка.
    for man in people:
        if man.get('name', period).lower() == period.lower():
            count += 1
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                '-' * 4,
                '-' * 30,
                '-' * 20,

```

Рисунок 4.2 – Код программы индивидуального задания

```

        '-' * 30,
        '-' * 20,
        '-' * 12
    )
    print(line)
    print(
        '| {:^4} | {:^30} | {:^20} | {:^12} |'.format(
            "№",
            "Ф.И.О.",
            "Телефон",
            "Год рождения"
        )
    )
    print(line)
    print(
        '| {:>4} | {:<30} | {:<20} | {:>12} |'.format(
            count,
            man.get('name', ''),
            man.get('tel', ''),
            man.get('date', 0)
        )
    )
    print(line)

    # Если счетчик равен 0, то работники не найдены.
    if count == 0:
        print("Люди с заданным именем не найдены.")

elif command == 'help':
    # Вывести справку о работе с программой.
    print("Список команд:\n")
    print("add - добавить человека;")
    print("list - вывести список людей;")
    print("select <имя> - запросить людей с этим именем;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")

else:
    print(f"Неизвестная команда {command}", file=sys.stderr)

```

Рисунок 4.3 – Код программы индивидуального задания

```

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\labor-9\PyCharm\Individual.py
>>> add
Фамилия и инициалы? Сотников А
Телефон? 89283480000
Год рождения? 2003
>>> add
Фамилия и инициалы? Мизин Г
Телефон? 89620002300
Год рождения? 2003
>>> add
Фамилия и инициалы? Человек
Телефон? 89113506852
Год рождения? 1950
>>> select Сотников А
+-----+-----+-----+
| № |          Ф.И.О.          |      Телефон      | Год рождения |
+-----+-----+-----+
|  1 | Сотников А              | 89283480000       |      2003   |
+-----+-----+-----+
>>> list
+-----+-----+-----+
| № |          Ф.И.О.          |      Телефон      | Год рождения |
+-----+-----+-----+
|  1 | Человек                 | 89113506852       |      1950   |
|  2 | Сотников А              | 89283480000       |      2003   |
|  3 | Мизин Г                 | 89620002300       |      2003   |
+-----+-----+-----+

```

Рисунок 4.4 – Результат работы программы индивидуального задания

## 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. Данные в словаре хранятся в формате ключ – значение.

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

Функция len() широко используется для определения размера объектов в Python. В нашем случае передача объекта словаря этой функции вернет размер словаря, то есть количество пар ключ-значение, присутствующих в словаре.

### 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.

С другой стороны у словаря как класса есть метод items(), который создает особую структуру, состоящую из кортежей. Каждый кортеж включает ключ и значение.

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.

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

```
>>> for i in nums:
...     print(nums[i])
...
one
two
three
```

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

С помощью метода `setdefault()`, при непосредственном обращении к ключу словарю.

## 6. Что такое словарь включений?

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

Основной пример:

```
>>> {x: x * x for x in (1, 2, 3, 4)}  
{1: 1, 2: 4, 3: 9, 4: 16}
```

## 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. Предположим, что есть список имен и номером сотрудников, и их нужно объединить в массив кортежей. Для этого можно использовать функцию `zip()`.

## 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. Поддерживается и стандартный способ представления времени, однако больший упор сделан на простоту манипулирования датой, временем и их частями.