

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №1  
Работа со словарями в языке Python.**

**по дисциплине «Технологии программирования и алгоритмизации»**

Выполнила студентка группы ИВТ-б-о-20-1

Новикова В.С. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2021

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

**Ход работы:**

Ссылка на репозиторий: [https://github.com/Valentina1502/LABA\\_1.git](https://github.com/Valentina1502/LABA_1.git)

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 (рис. 1).

Код:

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

from datetime import date
import sys

if __name__ == '__main__':
    # список работников
    workers = []

    # организация бесконечного цикла запроса команд
    while True:
        # запросить команду из терминала
        command = input(">>>").lower()

        # выполнить действие в соответствии с командой
        if command == 'exit':
            break

        elif command == 'add':
            # запрос данных пользователя
            name = input("Фамилия и инициалы? ")
            post = input("Должность? ")
            year = int(input("Год поступления? "))

            # создать словарь
            worker = {
                'name': name,
                'post': post,
                'year': year,
            }

            # добавление словаря в список
            workers.append(worker)
            # сортировка списка в случае необходимости
```

```

        if len(workers) > 1:
            workers.sort(key=lambda item: item.get('name', ''))

    elif command == 'list':
        line = '+-{}-+-{}-+-{}-+-{}-+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 8
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
                "№",
                "Ф.И.О.",
                "Должность",
                "Год"
            )
        )
        print(line)

        # вывод данных о всех сотрудниках
        for idx, worker in enumerate(workers, 1):
            print(
                '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
                    idx,
                    worker.get('name', ''),
                    worker.get('post', ''),
                    worker.get('year', '')
                )
            )

        print(line)

    elif command.startswith('select '):
        # получить текущую дату
        today = date.today()

        # разобрать команду на части для выделения номера года
        parts = command.split(' ', maxsplit=1)
        # Получить требуемый стаж
        period = int(parts[1])

        # Инициализировать счетчик
        count = 0
        # Проверить сведения работников из списка
        for worker in workers:
            if today.year - worker.get('year', today.year) >= period:
                count += 1
            print(
                '{:>4}: {}'.format(count, worker.get('name', ''))
            )

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

    elif command == 'help':
        # Вывести справку о работе с программой
        print("Список команд:\n")
        print("add - добавить работника;")
        print("list - вывести список работников;")
        print("select <стаж> - запросить работников со стажем;")
        print("help - отобразить справку;")

```

```
print("exit - завершить работу с программой.")

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

  

```
C:\Users\Валентина\AppData\Local\Programs\Python\Python39\python.exe C:/Users
>>>add
Фамилия и инициалы? Иванов И.И.
Должность? студент
Год поступления? 2020
>>>add
Фамилия и инициалы? Петров П.П.
Должность? студент
Год поступления? 2018
>>>select 3
1: Петров П.П.
>>>list
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
| 1 |      Петров П.П.      |      студент      |      2018      |
| 2 |      Иванов И.И.      |      студент      |      2020      |
+-----+-----+-----+-----+
>>>exit

Process finished with exit code 0
```

Рисунок 1 – Пример 1

Задача 1 (рис. 2): 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:

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

import sys

if __name__ == '__main__':
    school = []
    # заполняем словарь
    school = {
        '1a': 20,
        '1b': 21,
        '2a': 21,
        '2c': 20,
```

```

    '3a': 19,
    '3b': 23,
}
# вывод списка
for key, value in school.items():
    print('в классе ', key, ' учатся ', value)

print('---- Изменения списка -----')

school['1b'] = '25'
del school['3a']
school['4a'] = '22'

for key, value in school.items():
    print('в классе ', key, ' учатся ', value)

# кол-во учеников в школе
s = 0
for i in school:
    s = s + int(school[i])
print("В школе учится: ", s)

```

```

C:\Users\Валентина\AppData\Local\Programs
в классе 1a учатся 20
в классе 1b учатся 21
в классе 2a учатся 21
в классе 2c учатся 20
в классе 3a учатся 19
в классе 3b учатся 23
---- Изменения списка -----
в классе 1a учатся 20
в классе 1b учатся 25
в классе 2a учатся 21
в классе 2c учатся 20
в классе 3b учатся 23
в классе 4a учатся 22
В школе учится: 131

Process finished with exit code 0

```

Рисунок 2 – Задача 1

Задача 2 (рис. 3): 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:

```

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

if __name__ == '__main__':
    dic = {1: 'Каждый', 2: 'Охотник', 3: 'Желает', 4: 'Знать'}
    print(dic)

```

```

for key, value in dic.items():
    print(key, ' - ', value)
swapped = dict(map(reversed, dic.items()))

for key, value in swapped.items():
    print(key, ' - ', value)
print(swapped)

```

```

C:\Users\Валентина\AppData\Local\Programs\Python\Python39
{1: 'Каждый', 2: 'Охотник', 3: 'Желает', 4: 'Знать'}
1 - Каждый
2 - Охотник
3 - Желает
4 - Знать
Каждый - 1
Охотник - 2
Желает - 3
Знать - 4
{'Каждый': 1, 'Охотник': 2, 'Желает': 3, 'Знать': 4}

```

Рисунок 3 – Задача 2

#### Вариант 11

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 датам рождения; вывод на экран информации о человеке, номер телефона которого введен с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение (рис. 4).

```

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

from datetime import date
import sys

if __name__ == '__main__':
    # список работников
    workers = []

    # организация бесконечного цикла запроса команд
    while True:
        # запросить команду из терминала
        command = input(">>>").lower()

        # выполнить действие в соответствии с командой
        if command == 'exit':
            break

        elif command == 'add':
            # запрос данных пользователя
            name = input("Имя: ")
            fam = input("Фамилия: ")
            year = input("Дата рождения (yyyy.mm.dd): ")
            tel = input("Телефон: (x-xxx-xxx-xx-xx): ")

```

```

# создать словарь
worker = {
    'name': name,
    'fam': fam,
    'year': year,
    'tel': tel,
}

# добавление словаря в список
workers.append(worker)
# сортировка списка в случае необходимости
if len(workers) > 1:
    workers.sort(key=lambda item: item.get('year', ''))

elif command.startswith('found '):

    # разобрать команду на части для выделения номера
    parts = command.split(' ', maxsplit=1)
    # Получить требуемый номер
    phone = (parts[1])

    # Инициализировать счетчик
    count = 0
    # Проверить сведения работников из списка
    for worker in workers:
        if worker.get('tel', '') == phone:
            count += 1
            print(
                '{:>4}: {:>15} | {:>15} | {:>15}'.format(count,
worker.get('name', ''), worker.get('fam', ''), worker.get('tel', ''))
            )

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

elif command == 'all':
    line = '+-{}-+-{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 4,
        '-' * 20,
        '-' * 20,
        '-' * 12,
        '-' * 20
    )
    print(line)
    print(
        '| {:^4} | {:^20} | {:^20} | {:^12} | {:^20} |'.format(
            "№",
            "Фамилия",
            "Имя",
            "Год",
            "Телефон"
        )
    )
    print(line)

    for idx, worker in enumerate(workers, 1):
        print(
            '| {:^4} | {:^20} | {:^20} | {:^12} | {:^20} |'.format(
                idx,
                worker.get('name', ''),
                worker.get('fam', ''),
                worker.get('year', ''),

```

```

        worker.get('tel', '')
    )

    print(line)

elif command == 'help':
    # Вывести справку о работе с программой
    print("Список команд:\n")
    print("add - добавить работника;")
    print("all - вывести список работников;")
    print("found <х-xxx-xxx-xx-xx> - найти работника по номеру;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")

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

```

```

C:\Users\Валентина\AppData\Local\Programs\Python\Python39\python.exe C:/Users/Валентина/LABA_
>>>add
Имя: Валентина
Фамилия: Новикова
Дата рождения (yyyy.mm.dd): 2000.02.15
Телефон: (х-xxx-xxx-xx-xx): 8-888-765-43-21
>>>add
Имя: Ольга
Фамилия: Иванова
Дата рождения (yyyy.mm.dd): 2000.01.09
Телефон: (х-xxx-xxx-xx-xx): 8-432-321-23-76
>>>found 8-888-765-43-21
1:      Валентина |      Новикова | 8-888-765-43-21
>>>all
+-----+-----+-----+-----+-----+
| № |      Фамилия      |      Имя      |      Год      |      Телефон      |
+-----+-----+-----+-----+-----+
| 1 |      Ольга      |      Иванова   | 2000.01.09 | 8-432-321-23-76 |
| 2 |      Валентина   |      Новикова   | 2000.02.15 | 8-888-765-43-21 |
+-----+-----+-----+-----+-----+

```

Рисунок 4 – Индивидуальное задание

### 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() быть использована при работе со словарями.



### 3. Какие методы обхода словарей Вам известны?

Все элементы в словаре можно обходить с помощью оператора for.

```
Nums={1: 'one', 2: 'two', 3: 'three'}
```

```
for i in nums:
```

```
    print(i)
```

Однако "по умолчанию" извлекаются только ключи. Но по ключам всегда можно получить значения:

```
for i in nums:
```

```
    print(nums[i])
```

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

Метод словаря keys() позволяет получить отдельно перечни ключей.

Через цикл For.

Обратиться на прямую к элементу через его ключ.

Метод get() позволяет получить элемент по его ключу.

### 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.

Основной пример:

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

```
{1: 1, 2: 4, 3: 9, 4: 16}
```

Еще один способ написания:

```
>>> dict((x, x * x) for x in (1, 2, 3, 4))
```

```
{1: 1, 2: 4, 3: 9, 4: 16}
```

Как и в случае со списком, мы можем использовать условный оператор внутри словаря включения, чтобы получить только элементы словаря, удовлетворяющие заданному критерию.

```
>>> {name: len(name) for name in ('Stack', 'Overflow', 'Exchange') if  
len(name) > 6}  
{'Exchange': 8, 'Overflow': 8}
```

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.

В Python есть несколько встроенных функций, которые позволяют перебирать данные. Одна из них — `zip`.

Пример работы функция `zip()` в Python

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()`. Вот пример программы, которая делает именно это:

```
employee_numbers = [2, 9, 18, 28]  
employee_names = ["Дима", "Марина", "Андрей", "Никита"]  
zipped_values = zip(employee_names, employee_numbers)  
zipped_list = list(zipped_values)  
print(zipped_list)
```

Функция `zip` возвращает следующее. Эта программа создала массив из кортежей, каждый из которых содержит имя и номер сотрудника.:

```
[('Дима', 2), ('Марина', 9), ('Андрей', 18), ('Никита', 28)]
```

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. Позволяет производить самые разные операции с датой и временем.

`datetime` включает различные компоненты. Так, он состоит из объектов следующих типов:

`date` — хранит дату

`time` — хранит время

`datetime` — хранит дату и время

**Вывод:** при выполнении заданий были приобретены навыки по работе со словарями при написании программ с помощью языка программирования Python.