

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

Кафедра инфокоммуникаций

Лабораторная работа 1
Работа со строками в языке Python3

Выполнил студент группы ИВТ-б-о-20-1

Симанский М.Ю « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

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

Пример

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.

Код программы

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

from datetime import date
import sys

if __name__ == '__main__':
    workers = []

    while True:
        command = input(">>> ").lower()
        # Выполнить действие в соответствии с командой.
        if command == 'exit':
            break
        elif command == 'add':
            # Запросить данные о работнике.
            name = input("Фамилия и инициалы? ")
            post = input("Должность? ")
            year = int(input("Год поступления? "))
            # Создать словарь.
            worker = {
                'name': name,
                'post': post,
                'year': year,
            }
```

```

# Добавить словарь в список.
workers.append(worker)
# Отсортировать список в случае необходимости.
if len(workers) > 1:
    workers.sort(key=lambda item: item.get('name', ''))
elif command == 'list':
    # Заголовок таблицы.
    line = '+-{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 4,
        '-' * 30,
        '-' * 20,
        '-' * 8
    )
    print(line)
    print(
        '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
            "№",
            "Ф.И.О.",
            "Должность",
            "Год"
        )
    )
    print(line)
    # Вывести данные 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', 0)
            )
        )
    print(line)

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

```

Результат выполнения

```
C:\Users\rac-e\anaconda3\python.exe "C:/Users/rac-e/YandexDisk/Лабы 3 семестр/Технологии прог
>>> add
Фамилия и инициалы? Симанский Максим Юрьевич
Должность? Начальник
Год поступления? 2019
>>> select 2
1: Симанский Максим Юрьевич
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
>>> list
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Симанский Максим Юрьевич |      Начальник      |      2019     |
+-----+-----+-----+-----+
>>>
```

Рисунок 1 – Работа программы

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.

Код программы

```

# -*- coding: utf-8 -*-

if __name__ == '__main__':
    school= {'1a':14, '3b':29, '6d':12, '3f':39, '5a':34}
    num = input("Добавьте класс ")
    qunt = input("Количество человек в классе ")
    school.setdefault(num, qunt)
    num2 = input("Удалить класс ")
    school.pop(num2)
    num3 = input("Выбрать класс для изменения ")
    num4 = input("Новое количество человек ")
    school[num3] = num4
    summ = 0
    for key, value in school.items():
        print(key, 'is', value)
        summ = summ + int(value)
    print("Количество человек ", summ)

```

Рисунок 2 – Код программы

Результат работы программы

```

Добавьте класс 3s
Количество человек в классе 12
Удалить класс 1a
Выбрать класс для изменения 3s
Новое количество человек 67
3b is 29
6d is 12
3f is 39
5a is 34
3s is 67
Количество человек 181

```

Рисунок 3 – Результат выполнения программы

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.

Код программы

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    alp = {1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}
    swapped = {v: k for k, v in alp.items()}
    print(swapped)
```

Рисунок 4 – Код решения задачи

Результат выполнения программы

```
C:\Users\rac-e\anaconda3\python.exe "C:/Use
{'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}

Process finished with exit code 0
```

Рисунок 5 – Результат выполнения программы

Индивидуальное задание. Вариант 18

Использовать словарь, содержащий следующие ключи: название товара; название магазина, в котором продается товар; стоимость товара в руб. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть размещены в алфавитном порядке по названиям магазинов; вывод на экран информации о товарах, продающихся в магазине,

название которого введено с клавиатуры; если такого магазина нет, выдать на дисплей соответствующее сообщение.

Код программы

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

if __name__ == '__main__':
    shops = []
    while True:
        command = input(">>> ").lower()
        if command == 'exit':
            break
        elif command == 'add':
            name = input("Название магазина ")
            product = input("Товар ")
            price = int(input("Цена "))
            shop = {
                'name': name,
                'product': product,
                'price': price,
            }
            shops.append(shop)
            if len(shops) > 1:
                shops.sort(key = lambda item: item.get('name', ''))
        elif command == 'list':
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                '-' * 4,
                '-' * 30,
                '-' * 20,
                '-' * 8
            )
            print(line)
            print(
                '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
                    "No",
                    "Название.",
                    "Товар",
                    "Цена"
                )
            )
            print(line)
            for idx, shop in enumerate(shops, 1):
                print(
                    '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
                        idx,
                        shop.get('name', ''),
                        shop.get('product', ''),
                        shop.get('price', 0)
                    )
                )
            print(line)
        elif command == 'select':
            sname = input("Название магазина ")
            cout = 0
            for shop in shops:
                if (shop.get('name') == sname):
                    cout = 1
```

```

        print(
            ' | {:<5} | {:<5} '.format(
                shop.get('product', ''),
                shop.get('price', 0),
            )
        )
    elif cout == 0: print("Такого магазина нет")
elif command == 'help':
    print("Список команд:\n")
    print("add - добавить магазин;")
    print("select - показать товары из заданного магазина;")
    print("list - вывести список магазинов;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")
else:
    print(f"Неизвестная команда {command}", file=sys.stderr)

```

Результат выполнения программы

```

>>> add
Название магазина Magnit
Товар kolbasa
Цена 345
>>> add
Название магазина Lenta
Товар moloko
Цена 34
>>> select
Название магазина Lenta
| moloko | 34
>>> help
Список команд:

add - добавить магазин;
select - показать товары из заданного магазина;
list - вывести список магазинов;
help - отобразить справку;
exit - завершить работу с программой.
>>> list
+-----+-----+-----+-----+
| No |      Название.      |      Товар      |      Цена      |
+-----+-----+-----+-----+
|  1 | Lenta                | moloko           | 34              |
+-----+-----+-----+-----+
|  2 | Magnit               | kolbasa          | 345             |
+-----+-----+-----+-----+
>>>

```

Рисунок 6 – Результат выполнения программы

Проверка каталога утилитой flake8


```
Администратор: Anaconda Prompt (anaconda3)
(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 8>flake8 primer.py
primer.py:0:1: E902 FileNotFoundError: [Errno 2] No such file or directory: 'primer.py'

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 8>flake8 Primer 1.1.py
Primer:0:1: E902 FileNotFoundError: [Errno 2] No such file or directory: 'Primer'
1.1.py:0:1: E902 FileNotFoundError: [Errno 2] No such file or directory: '1.1.py'

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 8>flake8 primer1.py

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 8>flake8 main.py
main.py:21:25: E202 whitespace before ')'
main.py:22:28: E201 whitespace after '('
main.py:22:32: E251 unexpected spaces around keyword / parameter equals
main.py:22:34: E251 unexpected spaces around keyword / parameter equals
main.py:22:68: E202 whitespace before ')'
main.py:25:13: E122 continuation line missing indentation or outdented
main.py:26:13: E122 continuation line missing indentation or outdented
main.py:27:13: E122 continuation line missing indentation or outdented
main.py:28:13: E122 continuation line missing indentation or outdented
main.py:40:45: E231 missing whitespace after ','
main.py:62:31: E701 multiple statements on one line (colon)
main.py:64:17: E117 over-indented
main.py:71:17: E117 over-indented
main.py:71:73: W292 no newline at end of file

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 8>flake8 main.py

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 8>flake8

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 8>
```

Рисунок 7 – Результат проверки flake8

Контрольные вопросы.

1. 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. Функция len() возвращает количество элементов(пар) в словаре.
3. При помощи цикла for , с использованием атрибута items или же при помощи списковых включений.
4. Метод get().
5. Метод setdefault().
6. Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка.
7. 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.

8. Этот модуль позволяет управлять датами и временем, представляя их в таком виде, в котором пользователи смогут их понимать. `datetime` включает различные компоненты. Так, он состоит из объектов следующих типов: `date` — хранит дату. `time` — хранит время.