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

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.6

Работа со словарями в языке Python

Выполнил студент группы

ИВТ-б-о-21-1

Мальцев Н.А. « »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил доцент
Кафедры инфокоммуникаций, старший преподаватель
Воронкин Р.А.

_____
(подпись)

Ставрополь 2022

Работа со словарями в языке Python.

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

Порядок выполнения работы:

1. Проработка примеров:

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, введённое с клавиатуры;

- если таких работников нет, вывести на дисплей соответствующее сообщение.

Код программы:

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

import sys
from datetime import date

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

    while True:
        command = input(">>> ").lower()

        if command == 'exit':
            break
        elif command == 'add':
            name = input("Фамилия и инициалы? ")
            post = input("Должность? ")
            year = int(input("Год поступления? "))

            worker = {
                'name': name,
                'post': post,
                'year': year,

            }

            workers.append(worker)

            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)

            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 '):
            today = date.today()
            parts = command.split('', maxsplit=1)
            period = int(parts[1])

            count = 0
            for worker in workers:
                if today.year - worker.get('year', today.year) >= period:
                    count += 1
                    print(
                        '{:>4}: {}'.format(count, worker.get('name', ''))
                    )
            if count == 0:
                print("Работники с заданным стажем не найдены.")

        elif command == 'help':
            print("Сисок комманд:\n")
            print("add - добавить работников;")
            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 комманда {command}", file=sys.stderr)
```

Результат работы программы:

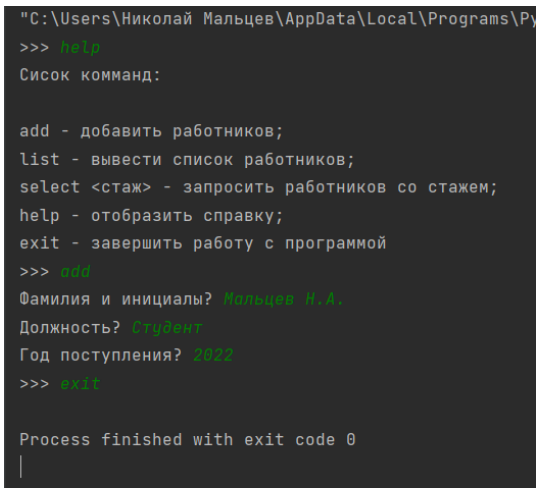

Рисунок 1. Результат работы программы из примера 1

2. Выполнение задачи 1:

Задача: 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:

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

import sys

if __name__ == '__main__':
    school = {'1А': 24, '1Б': 32, '1В': 16, '2А': 21, '2Б': 15, '3А': 10}
    print("Программа начала работу...")
    while True:
        command = input(">> ").lower()

        if command == 'exit':
            break
        # Добавление класса в словарь
        elif command == 'add':
            cl_add = input("Класс: ")
            count_add = input("Количество учеников: ")

            school[cl_add] = count_add
        # Вывод списка комманд
        elif command == 'help':
            print(
                "Список команд: \n",
                "add - добавить класс\n",
                "delite - удалить класс\n",
                "disp - вывести классы\n",
                "help - список команд\n",
                "summ - общее количество учащихся в школе\n",
                "exit - выход из программы"
```

```
                )
        # Удаление класса из словаря
        elif command == 'delite':
            key = input("Класс, который необходмо удалить: ")
            school.pop(key)
        # Вывод словаря
        elif command == 'displ':
            print(school)
        #Вывод общего количества учащихся
        elif command == 'summ':
            count = sum(school.values())
            print("Количество учеников в школе: ", count)
        # Вывод ошибки в случае ввода неправильной команды
        else:
            print("Неизвестная команда {command}", file=sys.stderr)
```

Результат работы программы:

```
"C:\Users\Николай Мальцев\AppData\Local\Programs\Python\Python310\python.
Программа начала работу...
>> help
Список команд:
 add - добавить класс
 delite - удалить класс
 disp - вывести классы
 help - список команд
 summ - общее количество учащихся в школе
 exit - выход из программы
>> summ
Количество учеников в школе:  118
>> add
Класс: 5
Количество учеников: 15
>> list
>> Неизвестная команда {command}
displ
{'1А': 24, '1Б': 32, '1В': 16, '2А': 21, '2Б': 15, '3А': 10, '5': '15'}
>> delite
Класс, который необходмо удалить: 2Б
>> displ
{'1А': 24, '1Б': 32, '1В': 16, '2А': 21, '3А': 10, '5': '15'}
>>
```

Рисунок 2. Результат работы программы к заданию 1

3. Выполнение задачи 2:

Задача: создайте словарь, где ключами являются числа, а значениями – строки. Примените к нему метод items(), с с помощью полученного объекта dict_items создайте новый словарь, "обратный" исходному, т. е. ключами являются строки, а значениями – числа.

Листинг программы:

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

if __name__ == '__main__':
    num = {'two': 2, 'three': 3, 'four': 4}
    print(num)

    dict_items = num.items()
```

```
    dict_inv = (lambda d: {v: k for k, v in d})
    print(dict_inv(dict_items))
```

Результат работы программы:

```
{'two': 2, 'three': 3, 'four': 4}
{2: 'two', 3: 'three', 4: 'four'}
```

Рисунок 3. Результат работы программы к заданию 2

4. Выполнение индивидуального задания (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.

Листинг программы:

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

import sys
import datetime

if __name__ == '__main__':
    # Список .
    manslist = []

    # Организовать бесконечный цикл запроса команд.
    while True:
        # Запросить команду из терминала.
        command = input(">>> ").lower()

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

        elif command == 'add':
            # Запросить данные .
            name = input("Имя:  ")
            number = input("Номер телефона ")
            date = input("Дата рождения  ")

            # Создать словарь.
            man = {
                'name': name,
                'number': number,
                'date': date,
            }
```

```
            # Добавить словарь в список.
            manslist.append(man)
            # Отсортировать список.
            if len(manslist) > 1:
                manslist.sort(key=lambda item:
datetime.datetime.strptime(item.get('date', ''), '%d.%m.%Y'))

        elif command == 'list':
            # Заголовок таблицы.
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                '-' * 4,
                '-' * 30,
                '-' * 20,
                '-' * 20
            )
            print(line)
            print(
                '| {:^4} | {:^30} | {:^20} | {:^20} |'.format(
                    "No",
                    "Имя",
                    "Номер телефона",
                    "Дата рождения"
                )
            )
            print(line)

            # Вывести данные о человеке.
            for idx, man in enumerate(manslist, 1):
                print(
                    '| {:>4} | {:<30} | {:<20} | {:>20} |'.format(
                        idx,
                        man.get('name', ''),
                        man.get('number', ''),
                        man.get('date', '')
                    )
                )

            print(line)

        elif command.startswith('select '):
            parts = command.split(' ', maxsplit=1)
            sel = parts[1]

            count = 0
            for man in manslist:
                if man.get('number') == sel:
                    count += 1
                    print(
                        '{:>4}: {}'.format(count, man.get('name', ''))
                    )
                    print('Номер телефона:', man.get('number', ''))
                    print('Дата рождения:', man.get('date', ''))

            # Если счетчик равен 0, то человек не найден.
            if count == 0:
                print("Человек не найден.")

        elif command == 'help':
            # Вывести справку о работе с программой.
            print("Список команд:\n")
            print("add - добавить человека;")
            print("list - вывести список людей;")
            print("select <товар> - информация о человеке;")
```

```
            print("help - отобразить справку;")
            print("exit - завершить работу с программой.")

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

Результат выполнения программы:

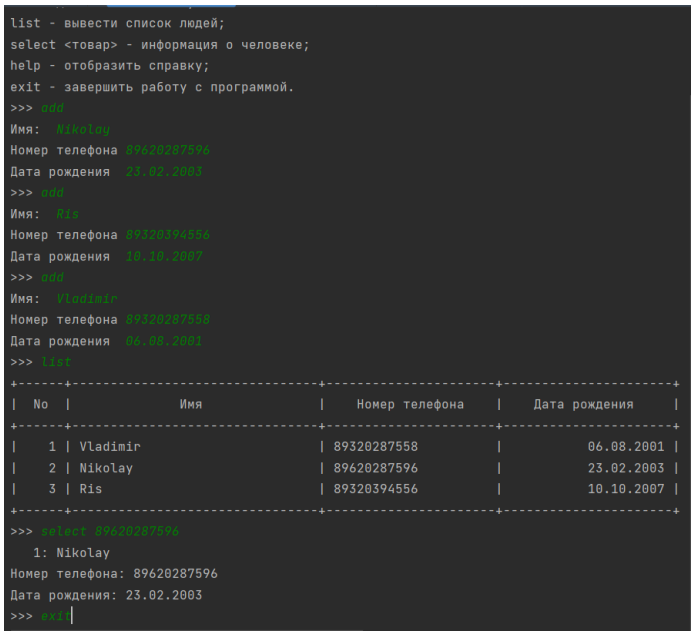

Рисунок 4. Результат выполнения программы к индивидуальному заданию

Ответы на вопросы:

1. Что такое словари в языке Python?

Это изменяемый неупорядоченный набор элементов «ключ-значение».

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

Да, может. В данном случае она возвращает количество пар ключ-значение в словаре.

3. Какие методы обхода словарей вам известны?

С помощью цикла for.

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

С помощью метода get().

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

Пример: x[ключ] = значение

6. Что такое словарь включений?

Словарь включений аналогичен списковым включениям, но он создаёт объект словаря вместо списка.

7. Самостоятельно изучите возможности функции zip() приведите примеры её использования.

Функция zip() в Python создаёт итератор , который объединяет элементы из нескольких источников данных. Например, имея два массива, можно создать на их основе словарь.

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

При помощи модуля datetime можно получить текущую дату и время в разных форматах, с разными часовыми поясами.

Вывод: в ходе работы были изучены словари в языке программирования Python, проработаны примеры их использования.