Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

**Работа с данными формата JSON в языке Python**

|  |
| --- |
| Выполнил:  Боженко Александр Иванович  2 курс, группа ИТС-б-о-21-1,  11.03.02 «Инфокоммуникационные технологии и системы связи», направленность (профиль)  «Инфокоммуникационные системы и сети», очная форма обучения  (подпись) |
| Руководитель практики:  Воронкин Р.А, канд. техн. наук, доцент кафедры инфокоммуникаций  (подпись) |

Отчет защищен с оценкой Дата защиты

Ставрополь, 2022 г.

**Цель работы**: приобретение навыков по работе с данными формата

JSON с помощью языка программирования Python версии 3.x. Ссылка на репозиторий https://github.com/danilusikov0913/YPlr6

**Ход работы:**

**1. Пример**

Для примера 1 лабораторной работы 2.8 добавьте возможность сохранения списка в файл формата JSON и чтения данных из файла JSON.

Решение: введем следующие команды для работы с файлом формата JSON в интерактивном режиме:

* load - загрузить данные из файла, имя файла должно отделяться от команды load пробелом. Например: load data.json
* save - сохранить сделанные изменения в файл, имя файла должно отделяться от команды save пробелом. Например: save data.json Напишем программу для решения поставленной задачи.

Код задания 1

import sys

import json

spisok\_new = []

def table():

    line = '+-{}-+-{}-+-{}-+-{}-+'.format(

        '-' \* 6,

        '-' \* 20,

        '-' \* 30,

        '-' \* 20

    )

    return line

def table\_name():

    post = '| {:^6} | {:^20} | {:^30} | {:^20} | '.format(

        "№",

        "пункт назначения",

        "номер",

        "время"

    )

    return post

def table\_name\_fil(names):

    post = []

    for idx\_new, spisok\_new\_new in enumerate(names, 1):

        post.append(

            '| {:>6} | {:<20} | {:<30} | {:<20} | '.format(

                idx\_new,

                spisok\_new\_new.get('name', ''),

                spisok\_new\_new.get('namber', ''),

                spisok\_new\_new.get('time', '')

            )

        )

    return post

def save\_list\_shop(file\_name, staff):

    with open(file\_name, "w", encoding="utf-8") as fout:

        json.dump(staff, fout, ensure\_ascii=False, indent=4)

def load\_list\_shop(file\_name):

    with open(file\_name, "r", encoding="utf-8") as fin:

        return json.load(fin)

def main():

    list\_shop = []

    while True:

        command = input('>>> ').lower()

        if command == 'exit':

            break

        elif command == 'add':

            name = input('Пункт назначения: ')

            namber = input('Номер поезда: ')

            time = input('время: ')

            list\_shop\_new = {

                'name': name,

                'namber': namber,

                'time': time

            }

            list\_shop.append(list\_shop\_new)

            if len(list\_shop) > 1:

                list\_shop.sort(key=lambda item: item.get('name\_shop', ''))

        elif command == 'list':

            print(table())

            print(table\_name())

            print(table())

            for item\_n in table\_name\_fil(list\_shop):

                print(item\_n)

            print(table())

        elif command == 'product':

            shop\_sear = input('Введите пункт назначения: ')

            search\_shop = []

            for shop\_sear\_itme in list\_shop:

                if shop\_sear == shop\_sear\_itme['name']:

                    search\_shop.append(shop\_sear\_itme)

            if len(search\_shop) > 0:

                print(table())

                print(table\_name())

                print(table())

                for item\_f in table\_name\_fil(search\_shop):

                    print(item\_f)

                print(table())

            else:

                print('Такого рейса нет', file=sys.stderr)

        elif command.startswith("save "):

            parts = command.split(maxsplit=1)

            file\_name = parts[1]

            save\_list\_shop(file\_name, list\_shop)

        elif command.startswith("load "):

            parts = command.split(maxsplit=1)

            file\_name = parts[1]

            list\_shop = load\_list\_shop(file\_name)

        elif command == 'help':

            print('Список команд:\n')

            print('add - добавить магазин.')

            print('list - вывести список магазинов.')

            print('product <Название> - запросить информацию о товаре.')

            print('help - Справочник.')

            print("load <Название файла без скобок> - загрузить данные из файла;")

            print("save <Название файла без скобок> - сохранить данные в файл;")

            print('exit - Завершить пработу программы.')

        else:

            print(f'Команда <{command}> не существует.', file=sys.stderr)

            print('Введите <help> для просмотра доступных команд')

if \_\_name\_\_ == '\_\_main\_\_':

    main()

результат программы

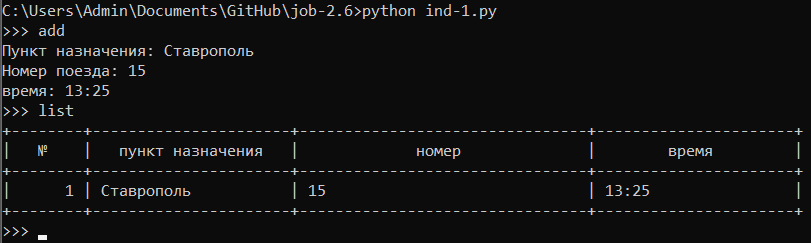


Рис 1.

**Контрольные вопросы:**

* 1. Для чего используется JSON?

JSON представляет собой хорошую альтернативу XML и требует куда меньше форматирования контента. Это информативное руководство поможет вам быстрее разобраться с данными, которые вы можете использовать с JSON и основной структурой с синтаксисом этого же формата.

* 1. Какие типы значений используются в JSON?

Запись, массив, число, литералы, строка

* 1. Как организована работа со сложными данными в JSON?
  2. Самостоятельно ознакомьтесь с форматом данных JSON5? В чем отличие этого формата от формата данных JSON?

JSON5 — предложенное расширение формата json в соответствии с синтаксисом ECMAScript 5, вызванное тем, что json используется не только для общения между программами, но и создаётся/редактируется вручную. Файл JSON5 всегда является корректным кодом ECMAScript 5. JSON5 обратно совместим с JSON

* 1. Какие средства языка программирования Python могут быть использованы для работы с данными в формате JSON5?

|  |  |
| --- | --- |
| [JSON5](https://json5.org/) расширяет формат обмена данными [JSON,](http://www.json.org/) чтобы сделать его немного более удобным в качестве языка конфигурации: | |
|  | * Комментарии в стиле JavaScript (как однострочные, так и многострочные) являются законными. * Ключи объектов могут быть без кавычек, если они являются законными идентификаторами ECMAScript * Объекты и массивы могут заканчиваться запятыми. * Строки могут заключаться в одинарные кавычки, и допускаются многострочные строковые литералы. |

* 1. Какие средства предоставляет язык Python для сериализации данных в формате JSON?

Модуль **json** предоставляет удобный метод dump() для записи данных в файл. Существует также метод dumps() для записи данных в обычную строку. Типы данных Python кодируются в формат JSON в соответствии с интуитивно понятными правилами преобразования

* 1. В чем отличие функций json.dump() и json.dumps()?

dump отличается от dumps тем, что dump записывает объект Python в

файл JSON, а dumps сериализует объект Python и хранит его в виде строки.

* 1. Какие средства предоставляет язык Python для десериализации данных из формата JSON?

В модуле **json** определены методы load() и loads(), предназначенные для преобразования кодированных в формате JSON данных в объекты Python.

Подобно операции *сериализации*, также существует таблица преобразования типов, определяющая правила для обратного *декодирования* данных.

* 1. Какие средства необходимо использовать для работы с данными формата JSON, содержащими кириллицу?

Параметр ensure\_ascii

**Вывод:** в ходе лабораторной работы приобретены навыки по работе с данными формата JSON с помощью языка программирования Python версии

3.x.