

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/YPIr6>

**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('number', ''),
                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: ')
            number = input('Номер поезда: ')
            time = input('время: ')

            list_shop_new = {
                'name': name,
                'number': number,
                '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 работу программы.')
else:
    print(f'Команда <{command}> не существует.', file=sys.stderr)
    print('Введите <help> для просмотра доступных команд')

if __name__ == '__main__':
    main()

```

результат программы

```

C:\Users\Admin\Documents\GitHub\job-2.6>python ind-1.py
>>> add
Пункт назначения: Ставрополь
Номер поезда: 15
время: 13:25
>>> list
+-----+-----+-----+-----+
| №      | пункт назначения | номер | время |
+-----+-----+-----+-----+
| 1      | Ставрополь      | 15    | 13:25 |
+-----+-----+-----+-----+
>>> _

```

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.

2. Какие типы значений используются в JSON?

Запись, массив, число, литералы, строка

3. Как организована работа со сложными данными в JSON?

4. 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

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

JSON5 расширяет формат обмена данными JSON, 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.

6. 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

7. 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.

8. 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.

9. 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.