

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ
ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Дисциплина: «Языки программирования»

Отчет по лабораторной работе №16
Работа с данными формата JSON в языке Python

Выполнил студент группы ИТС-б-о-21-
1

Романов Платон Дмитриевич

«__»_____20__г.

Подпись студента_____

Проверил: Доцент, к.т.н, доцент
кафедры инфокоммуникаций

Воронкин Роман Александрович

(подпись)

Ставрополь, 2022

Цель работы: приобретение навыков по работе с данными формата JSON с помощью языка программирования Python версии 3.x.

Ссылка на репозиторий - <https://github.com/lesnaya1shelupon/3sem6lab>

Ход работы:

Задание

Для своего варианта лабораторной работы 2.8 необходимо дополнительно реализовать сохранение и чтение данных из файла формата JSON. Необходимо также проследить за тем, чтобы файлы генерируемый этой программой не попадали в репозиторий лабораторной работы.

```
elif command.startswith('load '):

    parts = command.split(' ', maxsplit=1)
    with open(parts[1], 'r', encoding="utf-8") as f:
        spisok = json.load(f)
        #validate(instance=list_shop, schema=schema)

elif command.startswith('save '):
    parts = command.split(' ', maxsplit=1)
    with open(parts[1], 'w') as f:
        json.dump(spisok, f)

elif command == 'help':
    print('Список команд:\n')
    print('add - добавить пользователя.')
    print('list - вывести список пользователей.')
    print('find <Знак зодиака> - запросить пользователей по знаку Зодиака.')
    print("load <имя файла> - загрузить данные из файла;")
    print("save <имя файла> - сохранить данные в файл;")
    print('help - Справочник.')
    print('exit - Завершить работу программы.')
else:
    print(f'Команда <{command}> не существует.', file=sys.stderr)
    print('Введите <help> для просмотра доступных команд')
```

Рисунок 1. Код лабораторной работы 2.8 с внесенными изменениями.

```
>>> load werwer.json
>>> list
```

№	Дата рождения	Фамилия	Имя	Знак Зодиака
1	2000 марта 14	Иванов	Иван	рак

```
>>>
```

Рисунок 2. Результат выполнения программой

```
lzd.py x werwer.json x
[{"surname": "Иванов", "name": "Иван", "post": "рак", "data": "2000 марта 14"}]
```

Рисунок 3. Файл werwer.json

Задание повышенной сложности

Очевидно, что программа в примере 1 и в индивидуальном задании никак не проверяет правильность загружаемых данных формата JSON. В следствие чего, необходимо после загрузки из файла JSON выполнять валидацию загруженных данных. Валидацию данных необходимо производить с использованием спецификации JSON Schema, описанной на сайте <https://json-schema.org/>. Одним из возможных вариантов работы с JSON Schema является использование пакета `jsonschema`, который не является частью стандартной библиотеки Python. Таким образом, необходимо реализовать валидацию загруженных данных с помощью спецификации JSON Schema.

```
schema = {
    "surname": {"type": "string"},
    "name": {"type": "string"},
    "post": {"type": "string"},
    "data": {"type": "string"},
}
```

```
elif command.startswith('load '):

    parts = command.split(' ', maxsplit=1)
    with open(parts[1], 'r', encoding="utf-8") as f:
        spisok = json.load(f)
        validate(instance=spisok, schema=schema)
```

Рисунок 4. Реализация валидации загруженных данных с помощью спецификации JSON Schema

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. 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

4. 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.

5. 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

6. 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.

7. 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.

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

Параметр `ensure_ascii`

Вывод: приобрел навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.