

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №2.16**

**Работа с файлами в языке Python**

**по дисциплине «Технологии программирования и алгоритмизации»**

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2021

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

### Ход работы: Примеры

<https://github.com/Mirror-Shard/L2.16>

1. Изучил теоретический материал и приступил к выполнению примеров:

```
def save_workers(file_name, staff):  
    """  
    Сохранить всех работников в файл JSON.  
    """  
    # Открыть файл с заданным именем для записи.  
    with open(file_name, "w", encoding="utf-8") as fout:  
        # Выполнить сериализацию данных в формат JSON.  
        # Для поддержки кириллицы установим ensure_ascii=False  
        json.dump(staff, fout, ensure_ascii=False, indent=4)  
  
def load_workers(file_name):  
    """  
    Загрузить всех работников из файла JSON.  
    """  
    # Открыть файл с заданным именем для чтения.  
    with open(file_name, "r", encoding="utf-8") as fin:  
        return json.load(fin)
```

Рисунок 1 – Добавление возможности сохранения списка

### Задание 1

2. Добавил возможность сохранения и загрузки списков для своего варианта:

```
>>> list
Список пуст
>>> load s.txt
>>> list
```

No	Ф.И.О.	Группа	Средняя оценка
1	KKK k k	20	5.0
2	AAA а а	19	4.8

```
>>>
```

Рисунок 2 – Чтение файла

- Затем добавил систему валидации данных jsonschema, создав файл со схемой и добавив функцию в основной файл:

```
1  {
2      "type": "array",
3      "items": {
4          "type": "object",
5          "properties": {
6              "name": {
7                  "type": "string",
8                  "minLength": 1,
9                  "maxLength": 40
10             },
11             "group": {
12                 "type": "integer",
13                 "minimum": 1,
14                 "maximum": 99
15             },
16             "average_estimation": {
17                 "type": "number"
18             }
19         },
20         "required": ["name", "group"]
21     }
22 }
```

Рисунок 3 – Схема для словаря

```
def validate(file):  
    with open("schema.json") as f:  
        schema = json.load(f)  
  
    validator = jsonschema.Draft7Validator(schema)  
    try:  
        if not validator.validate(file):  
            print("Валидация прошла успешно!")  
    except jsonschema.exceptions.ValidationError:  
        print("Ошибка валидации!")
```

Рисунок 4 – Функция валидации

### Контрольные вопросы:

1. Для чего используется JSON?
  - JSON используется для обмена данными, которые являются структурированными и хранятся в файле или в строке кода.
2. Какие типы значений используются в JSON?
  - string;
  - number;
  - object;
  - array;
  - boolean;
  - null.
3. Как организована работа со сложными данными в JSON?
  - Данные также могут быть вложены в формате JSON, используя JavaScript массивы, которые передаются как значения. При помощи вложенных массивов и объектов можно создать сложную иерархию данных.
4. Какие средства предоставляет язык Python для сериализации данных в формате JSON?
  - Процесс кодирования данных в необходимый формат называется сериализацией. Для того чтобы записать эти данные в файл с форматом JSON

в Python, используются функция `dump()` и `dumps()`.

5. 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.

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

– Когда есть файл JSON, который необходимо преобразовать в объект Python, тогда проводится десериализация. Для десериализации по аналогии используются две функции: `load()` и `loads()`.

7. Самостоятельно ознакомьтесь со спецификацией JSON Schema? Что такое схема данных? Приведите схему данных для примера 1.

Схема JSON – это словарь, который позволяет аннотировать и проверять документы JSON.

Преимущества:

- описывает ваш существующий формат(ы) данных;
- обеспечивает четкую читаемую документацию для человека и машины;
- проверяет данные, которые полезны для автоматизированного тестирования и обеспечения качества

предоставляемых клиентом данных.

Пример схемы:

```
Schema = {  
    "type": "object", "employees": {  
        "name": {"type": "string"},  
        "post": {"type": "string"},  
        "year": {"type": "string", "format": "date"}  
    }  
}
```

**Вывод:** в ходе работы приобрёл навыки по работе с данными формата JSON при написании программ с помощью языка программирования Python версии 3.