

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 ПРАКТИЧЕСКОЙ РАБОТЕ №2**  
**дисциплины «Анализ данных»**

Выполнил:

Пушкин Никита Сергеевич  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем»

---

(подпись)

Руководитель практики: кандидат тех.  
наук доцент кафедры  
инфокоммуникаций: Воронкин Р.А

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

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

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

**Порядок выполнения работы:**

Для начала устанавливаем виртуальное окружение

```
python -m venv .venv
```

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

**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.

```

55     def add(humans):
56         # Функция добавления новых людей
57         daytime = input('Введите дату рождения: ')
58         zodiac = input('Введите знак зодиака: ')
59         name = input('Введите Ф.И.О.: ')
60         air = {
61             'zodiac': zodiac,
62             'name': name,
63             'daytime': daytime
64         }
65
66         humans.append(air)
67         save_to_json(file_path: 'humans.json', humans) # Сохраняем данных
68         if len(humans) > 1:
69             humans.sort(key=lambda x: x.get('daytime', ''))
70
71
72     1 usage
73     def save_to_json(file_path, data):
74         with open(file_path, 'w') as file:
75             json.dump(data, file)
76
77     1 usage
78     def load_from_json(file_path):
79         try:
80             with open(file_path, 'r') as file:
81                 return json.load(file)
82         except FileNotFoundError:
83             return []

```

```

1 [{"zodiac": "\u0422\u0435\u043b\u0435\u0446", "name": "\u041d\u0430\u0437\u0430\u0440\u043a\u043e\u0432 \u0414\u0430\u0432\u0438\u0434", "daytime": "09/05/2002"},

```

```

C:\Users\Никита\AppData\Local\Programs\Python\Python311\python.exe "C:\Users\Никита\Desktop\Анализ данных\pythonfiles\ind.py"
Список команд:
exit - Завершить работу
add - Добавить человека
list - Показать список людей
select - Выбрать знак зодиака по дате рождения
Введите команду: list
+-----+-----+-----+-----+
| № | Ф.И.О. | Знак зодиака | Дата рождения |
+-----+-----+-----+-----+
| 1 | Назарков Д.А. | Телец | 09/05/2002 |
| 2 | Павленко М.С. | Рак | 11/07/2002 |
| 3 | Пушкин Н.С. | Стрелец | 16/12/2002 |
| 4 | Коновалова Ю.Д. | Козерог | 23/12/2002 |
+-----+-----+-----+-----+
Введите команду: |

```

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.

```
usage
def load_from_json(file_path):
    try:
        with open(file_path, 'r') as file:
            data = json.load(file)
            validate_data(data) # Вызываем функцию валидации
            return data
    except FileNotFoundError:
        return []

def validate_data(data):
    schema = {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "name": {"type": "string"},
                "zodiac": {"type": "string"},
                "daytime": {"type": "string", "format": "date"}
            },
            "required": ["name", "zodiac", "daytime"]
        }
    }

    jsonschema.validate(data, schema)
```

```
1  {
2      "type": "array",
3      "items": {
4          "type": "object",
5          "properties": {
6              "name": {"type": "string"},
7              "zodiac": {"type": "string"},
8              "daytime": {"type": "string", "format": "date"}
9          },
10         "required": ["name", "zodiac", "daytime"]
11     }
12 }
```

## Ответы на вопросы:

1. Для чего используется JSON?
  - JSON используется для обмена данными между приложениями, веб-серверами и клиентами в удобном и легко читаемом формате.
2. Какие типы значений используются в JSON?
  - JSON поддерживает строки, числа, логические значения, массивы, объекты, а также значения null.
3. Как организована работа со сложными данными в JSON?
  - Для работы со сложными данными в JSON используются массивы и вложенные объекты, позволяющие организовать и структурировать информацию.
4. Самостоятельно ознакомьтесь с форматом данных JSON5? В чем отличие этого формата от формата данных JSON?
  - JSON5 расширяет синтаксис JSON, позволяя использовать комментарии, без кавычек для ключей, разрывать строки и использовать дополнительные данные типов.
5. Какие средства языка программирования Python могут быть использованы для работы с данными в формате JSON5?
  - Для работы с JSON5 в Python могут использоваться библиотеки, например, "json5" или "json5-parser".
6. Какие средства предоставляет язык Python для сериализации данных в формате JSON?
  - Python предоставляет модуль json для сериализации данных в JSON формат.
7. В чем отличие функций json.dump() и json.dumps()?
  - json.dump() записывает JSON данные в файл, а json.dumps() возвращает JSON в виде строки.
8. Какие средства предоставляет язык Python для десериализации данных из формата JSON?
  - Для десериализации данных из JSON формата в Python используется метод json.loads().
9. Какие средства необходимо использовать для работы с данными формата JSON, содержащими кириллицу?

- Для работы с данными формата JSON, содержащими кириллицу, необходимо убедиться, что данные корректно кодируются, используя например UTF-8.

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

- Схема данных используется для определения структуры и правил валидации данных в формате JSON. Путем описания свойств объектов JSON, определяется корректность данных в соответствии с требованиями схемы.

- Json для примера 1:

```
{
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "post": { "type": "string" },
    "year": { "type": "number" }
  },
  "required": ["name", "post", "year"]
}
```

**Вывод:** на основе выполненной работы приобрёл навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.