

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

Выполнила:  
Кубанова Ксения Олеговна  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А.

---

(подпись)

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

Ставрополь, 2024 г.

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

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

### Порядок выполнения работы

#### Пример 1.

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

```
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
load - загрузить данные из файла;
save - сохранить данные в файл;
exit - завершить работу с программой.
>>> add
Фамилия и инициалы? gdvgv
Должность? sfcdsc
```

Рисунок 1 – результат работы примера 1

Выполненный код примера находится в файле *prim1.py*.

#### Индивидуальное задание 1.

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.

```
>>> add
Номер поезда? 6
Пункт назначения? New-York
Время отправления? 6
>>> save first.json
>>> load first.json
>>> list
```

№	Номер поезда	Пункт назначения	Время отправления
1	1	London	3
2	6	New-York	6

```
>>> 
```

## Рисунок 2 – результат работы ИДЗ1

Выполненный код находится в файле *ind1.py*.

### Задание повышенной сложности.

Необходимо использовать модуль *jsonschema* для валидации данных.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import json
5  from jsonschema import validate
6
7  if __name__ == "__main__":
8      schema = {
9          "type": "object",
10         "properties": {
11             "punkt": {"type": "string"},
12             "nomer": {"type": "string"},
13             "time": {"type": "number"},
14         },
15     }
16
17     with open("first.json", "r", encoding="utf-8") as fin:
18         a = json.load(fin)
19         for i in a:
20             validate(instance=i, schema=schema)
```

## Рисунок 3 – задание повышенной сложности

**Вывод:** в ходе выполнения лабораторной работы были приобретены навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.