

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
«Работа с данными формата JSON в языке Python»

Отчет по лабораторной работе № 2.16
по дисциплине «Анализ данных»

Выполнил студент группы ИВТ-б-о-22-1

Сумин Никита Сергеевич.

« » _____ 2024 г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2024

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

Порядок выполнения работы:

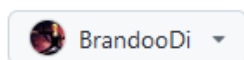
1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

Create a new repository

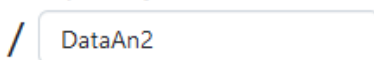
A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *



Repository name *



✔ DataAn2 is available.

Great repository names are short and memorable. Need inspiration? How about [ubiquitous-octo-happiness](#) ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

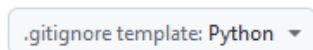
Initialize this repository with:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore



Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license



A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  main as the default branch. Change the default name in your [settings](#).



You are creating a public repository in your personal account.

Create repository

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.
Проработать примеры лабораторной работы.

```

C:\Users\mikeg\anaconda3\python.exe C:\PY\DataAn2\Project\Примеры\1.py
>>> add
Фамилия и инициалы? Иванов В.И.
Должность? Актер
Год поступления? 2000
>>> add
Фамилия и инициалы? Петров Д.В.
Должность? Лётчик
Год поступления? 1990
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Иванов В.И.              |      Актер          |      2000     |
|  2 | Петров Д.В.              |      Лётчик         |      1990     |
+-----+-----+-----+-----+
>>> save primer
>>> load primer
>>> exit

Process finished with exit code 0

```

Рисунок 2 – Результат выполнения примера

6. Выполнить индивидуальные задания.

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.

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.

```
C:\Users\nikit\anaconda3\python.exe "D:\project\mywork2.16\Project\Индивидуальные задания\ind_hard.py"
>>> add
Пункт назначения: Москва
Номер рейса: 123
Тип самолета: грузовой
>>> list
+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Москва | 123 | грузовой |
+-----+-----+-----+-----+
>>> save planes
>>> load planes
Валидация прошла успешно
>>> exit

Process finished with exit code 0
```

Рисунок 6 - Результат выполнения индивидуального задания

Контрольные вопросы:

1. Для чего используется JSON?

JSON (англ. JavaScript Object Notation, обычно произносится как /'dʒeɪsən/ JAY-sən) – текстовый формат обмена данными, основанный на JavaScript.

За счёт своей лаконичности по сравнению с XML формат JSON может быть более подходящим для сериализации сложных структур. Применяется в веб-приложениях как для обмена данными между браузером и сервером (AJAX), так и между серверами (программные HTTP-сопряжения).

2. Какие типы значений используются в JSON? В качестве

значений в JSON могут быть использованы:

запись — это неупорядоченное множество пар ключ:значение, заключённое в фигурные скобки «{ }». Ключ описывается строкой, между ним и значением стоит символ «:». Пары ключ-значение отделяются друг от друга запятыми.

массив (одномерный) — это упорядоченное множество значений. Массив заключается в квадратные скобки «[]». Значения разделяются запятыми. Массив может быть пустым, т.е. не содержать ни одного значения. Значения в пределах одного массива могут иметь разный тип.

число (целое или вещественное).

литералы true (логическое значение «истина»), false (логическое значение «ложь») и null.

строка — это упорядоченное множество из нуля или более символов юникода, заключённое в двойные кавычки. Символы могут быть указаны с использованием escape-последовательностей, начинающихся с обратной косой черты «\» (поддерживаются варианты ' , " , \ , \/, \t, \n, \r, \f и \b), или записаны шестнадцатеричным кодом в кодировке Unicode в виде \uFFFF. **3.**

Как организована работа со сложными данными в JSON?

JSON может содержать другие вложенные объекты в JSON, в дополнение к вложенным массивам. Такие объекты и массивы будут передаваться, как значения назначенные ключам и будут представлять собой связку ключ-значение.

4. Самостоятельно ознакомьтесь с форматом данных

JSON5? В чем отличие этого формата от формата данных JSON?

Формат обмена данными JSON5 — это расширенная JSON-версия, которая призвана смягчить некоторые ограничения JSON, расширив его синтаксис и включив в него некоторые функции из ECMAScript 5.1.

5. Какие средства языка программирования Python могут быть использованы для работы с данными в формате JSON5? Чтобы использовать JSON5, нужно установить пакет json5. json5 становится частью зависимостей в package.json.

6. Какие средства предоставляет язык Python для сериализации данных в формате JSON? json.dump() json.dumps()

7. В чем отличие функций json.dump() и json.dumps()? json.dump() # конвертировать python объект в json и записать в файл json.dumps() # тоже самое, но в строку

8. Какие средства предоставляет язык Python для десериализации данных из формата JSON? json.load() # прочитать json из файла и конвертировать в python объект json.loads() # тоже самое, но из строки с json (s на конце от string/строка)

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

#Для поддержки кириллицы установим ensure_ascii=False json.dump(staff, fout, ensure_ascii=False, indent=4)

10. Самостоятельно ознакомьтесь со спецификацией JSON

Schema? Что такое схема данных? Приведите схему данных для примера 1.

Схема является JSON-объектом, предназначенным для описания какихлибо данных в формате JSON.

Схема для примера 1: schema
= {

```
"type": "array",
"items": {
  "type": "object",
  "minProperties": 3,
  "maxProperties": 3,
  "properties": {
    "name": {"type": "string"},
    "post": {"type": "string"},
    "year": {"type": "integer"}
  },
  "required": ["name", "post", "year"]
}
}
```

Вывод: были приобретены навыки по работе с данными формата JSON при написании программ с помощью языка программирования Python версии 3.x.