

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 «Программирование на Python»

Выполнил студент группы ИВТ-б-о-21-1

Стригалов Дмитрий.

«27» ноября 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

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

Порядок выполнения работы:

Проработать примеры лабораторной работы.

```
>>> add
Фамилия и инициалы? Иванов И.И.
Должность? Учитель
Год поступления? 2017
>>> add
Фамилия и инициалы? Гришин А.А.
Должность? Бухгалтер
Год поступления? 2015
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|   1 | Гришин А.А.              | Бухгалтер           |      2015      |
|   2 | Иванов И.И.              | Учитель             |      2017      |
+-----+-----+-----+-----+
>>> save primer
>>> load primer
>>> exit
```

Рисунок 1 – Результат выполнения примера

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.

```

help - список всех команд
>>> add
Введите название для товара: new balance
Введите название магазина: krossovki new balance
Стоимость товара: 5600.434
>>> list
+-----+-----+-----+-----+
|  №   | Название товара |      Цена      | Название магаз |
+-----+-----+-----+-----+
|    1 | new balance     | 5600.43        | krossovki new  |
+-----+-----+-----+-----+
>>> save products
>>> load products
>>> |

```

Рисунок 2 - Результат выполнения индивидуального задания

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.

```

>>> load ind2
Валидация прошла успешно
>>> list
+-----+-----+-----+-----+
|  №  | Название товара |      Цена      | Название магазина |
+-----+-----+-----+-----+
|   1 | new balance    |    323.32     |      new balance   |
+-----+-----+-----+-----+
>>> select
Товар в наличии: new balance
Цена: 323.32

```

Рисунок 3 - 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.