МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №11
Работа с данными формата JSON в языке Python.
По дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1
Злыгостев И.С. « »_____20__г.
Подпись студента_____
Работа защищена « »_____20__г.
Проверил Воронкин Р. А. _____
(подпись)

Ставрополь 2021

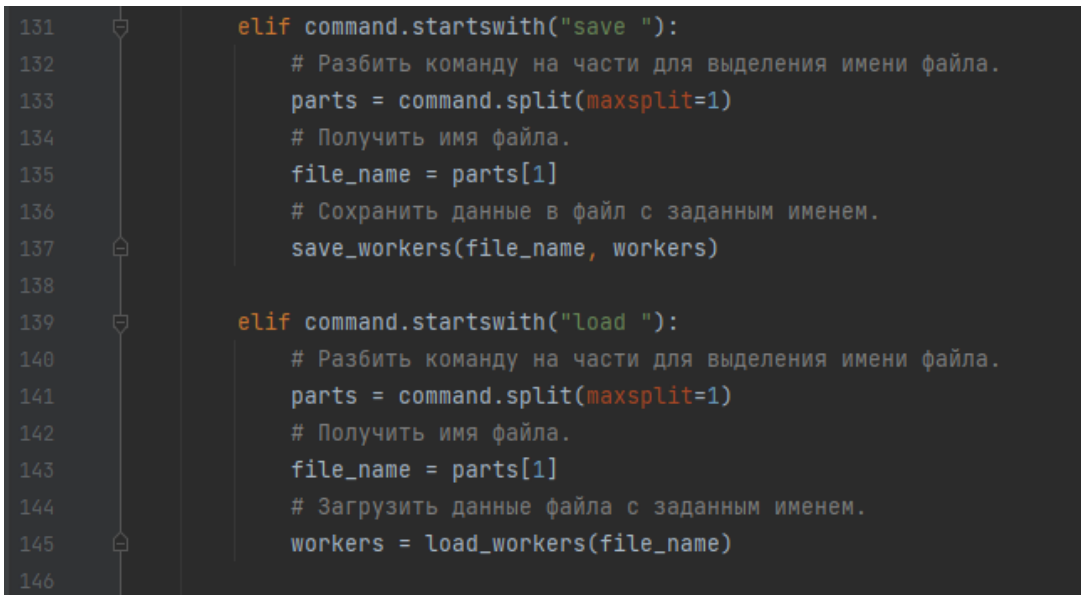
Цель работы: приобретение навыков по работе с данными формата JSON с помощью языка программирования Python версии 3.x.

Ход работы:

1. Изучил теоретический материал и проработал примеры работы с JSON.

```
77  def save_workers(file_name, staff):
78      """
79      Сохранить всех работников в файле JSON
80      """
81      with open(file_name, "w", encoding="utf-8") as fout:
82          json.dump(staff, fout, ensure_ascii=False, indent=4)
83
84
85  def load_workers(file_name):
86      """
87      Загрузить всех работников из файла JSON
88      """
89      with open(file_name, "r", encoding="utf-8") as fin:
90          return json.load(fin)
91
```

Рисунок 1 – Функции сохранения и загрузки работников



```
131         elif command.startswith("save "):
132             # Разбить команду на части для выделения имени файла.
133             parts = command.split(maxsplit=1)
134             # Получить имя файла.
135             file_name = parts[1]
136             # Сохранить данные в файл с заданным именем.
137             save_workers(file_name, workers)
138 
139         elif command.startswith("load "):
140             # Разбить команду на части для выделения имени файла.
141             parts = command.split(maxsplit=1)
142             # Получить имя файла.
143             file_name = parts[1]
144             # Загрузить данные файла с заданным именем.
145             workers = load_workers(file_name)
146 
```

Рисунок 2 – Обращение к функциям по командам

2. Сделал проверку работоспособности загрузки и сохранения файлов. Для этого добавил данные в таблицу. С помощью команды save сохранил свою таблицу в формате json.

Индивидуальное задание.

Вариант 3.

1. В качестве индивидуального задания взял работу из лабораторной работы 2.8. Добавил две функции, отвечающие за сохранение и загрузку файлов.

```python
def saving(file_name, flights):
    with open('../json/'+file_name, "w", encoding="utf-8") as file_out:
        # Выполнить сериализацию данных в формат JSON.
        # Для поддержки кирилицы установим ensure_ascii=False
        print("Файл сохранён")
        json.dump(flights, file_out, ensure_ascii=False, indent=4)


def opening(file_name):
    """Загрузить всех работников из файла JSON."""
    # Открыть файл с заданным именем для чтения.
    with open('../json/'+file_name, "r", encoding="utf-8") as f_in:
        file = json.load(f_in)
        print("Файл загружен")
        with open('../json/check.json') as check:
            schema = json.load(check)
            validator = jsonschema.Draft7Validator(schema)
            try:
                if not validator.validate(file):
                    print("Нет ошибок валидации")
            except jsonschema.exceptions.ValidationError:
                print("Ошибка валидации", list(validator.iter_errors(file)))
                exit()
        return file
```

Рисунок 3 – Объявление функций сохранения и загрузки данных

```python
        elif com.startswith("save "):
            # Разбить команду на части для выделения имени файла.
            parts = com.split(maxsplit=1)
            # Получить имя файла.
            file_name = parts[1]
            # Сохранить данные в файл с заданным именем.
            saving(file_name, flights)
        elif com.startswith("load "):
            # Разбить команду на части для выделения имени файла.
            parts = com.split(maxsplit=1)
            # Получить имя файла.
            file_name = parts[1]
            # Сохранить данные в файл с заданным именем.
            flights = opening(file_name)
```

Рисунок 4 – Обращение к командам save и load

2. Сделал проверку кода. Для этого внес исходные данные в таблицу и сохранил её.

№	Место прибытия	Номер самолёта	Тип
1	LONDON	RF-862333	BOING
2	PARIS	RF-86231	AIRBUS

Рисунок 5 – Сохранение исходной таблицы

3. Проверил наличии файла с расширением json.

Рисунок 6 – Текстовый файл с данными

4. Затем самостоятельно изучил способ валидации файлов через jsonschema и реализовал её в своём коде.

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. Самостоятельно ознакомьтесь с форматом данных JSON5? В чем отличие этого формата от формата данных JSON?

Формат обмена данными JSON5 (JSON5) — это надмножество JSON, которое направлено на смягчение некоторых ограничений JSON путем расширения его синтаксиса для включения некоторых продуктов из ECMAScript 5.1.

JSON5 получил следующие новшества:

– строки могут охватывать несколько строк, экранируя новые символы строк;

– числа могут быть шестнадцатеричными;

– допускаются однострочные и многострочные комментарии;

– ключи объектов могут быть без кавычек, если они являются законными идентификаторами ECMAScript;

– объекты и массивы могут заканчиваться запятыми в конце.

Существует одно заметное отличие от JSON: методы load() и loads() поддерживают выборочную проверку (и отклонение) дубликатов ключей объектов.

4. Какие средства языка программирования Python могут быть использованы для работы с данными в формате JSON5?

- json.load()
- json.loads();
- json.tool();

- json.dump();
- json.dumps().

6. 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().

7. 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.

8. 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().

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

– При записи достаточно передать ensure_ascii=False, чтобы не экранировать не-ascii символы.

10. 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 выполнения лабораторной работы были приобретены навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.