

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

Отчет по лабораторной работе № 2.16
Работа с данными формата JSON в языке Python

Выполнил студент группы ИВТ-б-о-20-1

Ищенко М.А.

Работа защищена « » _____ 20__ г.

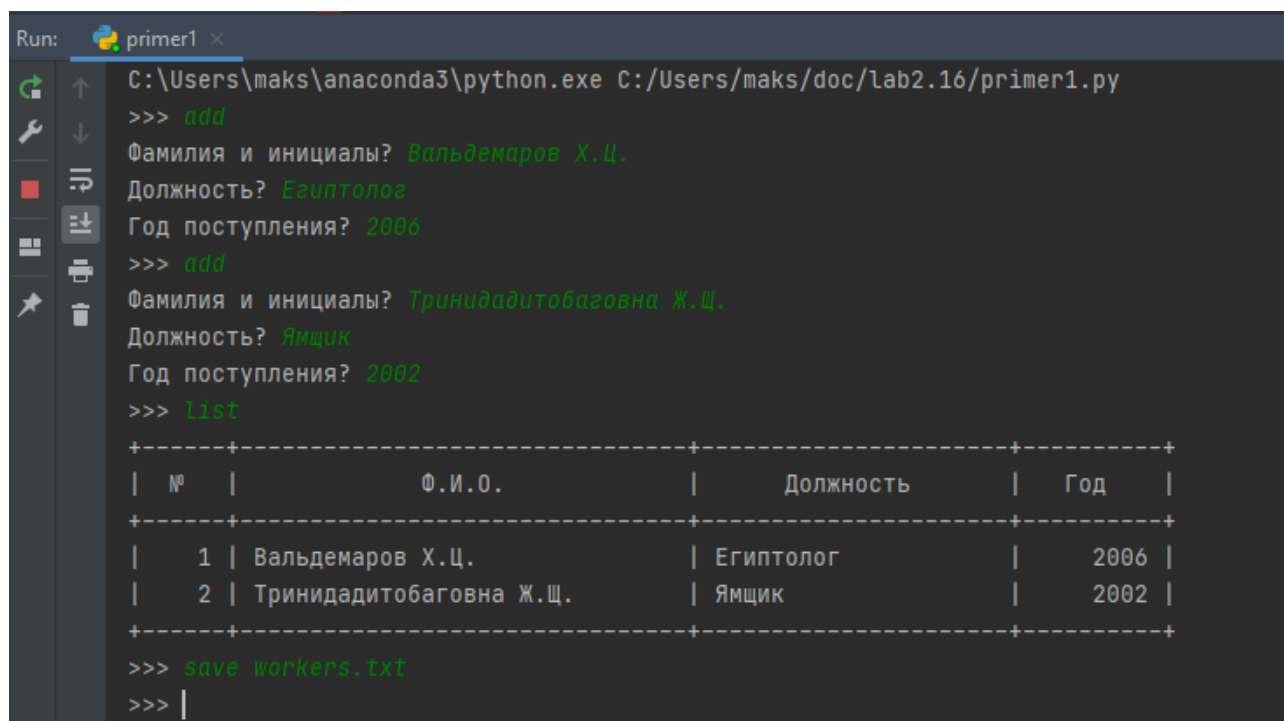
Проверил(а) _____

Ставрополь 2021

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

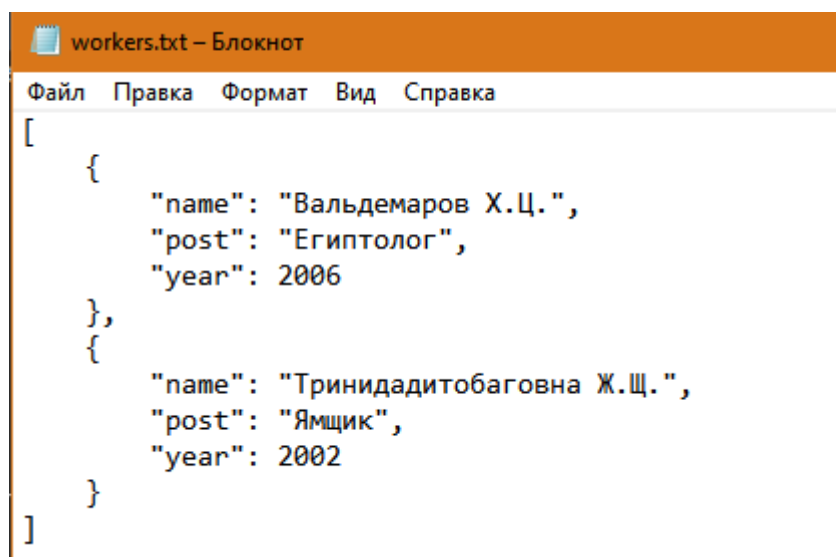
Создан общедоступный репозиторий на GitHub. Дополнен файл .gitignore необходимыми правилами для работы с IDE PyCharm.

Проработан пример из лабораторной работы, рис. 1-3



```
Run: primer1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.16/primer1.py
>>> add
Фамилия и инициалы? Вальдемаров Х.Ц.
Должность? Египтолог
Год поступления? 2006
>>> add
Фамилия и инициалы? Тринидадитобаговна Ж.Щ.
Должность? Ямщик
Год поступления? 2002
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Вальдемаров Х.Ц. | Египтолог | 2006 |
| 2 | Тринидадитобаговна Ж.Щ. | Ямщик | 2002 |
+-----+-----+-----+-----+
>>> save workers.txt
>>> |
```

Рисунок 1 – Пример



```
workers.txt – Блокнот
Файл Правка Формат Вид Справка
[
    {
        "name": "Вальдемаров Х.Ц.",
        "post": "Египтолог",
        "year": 2006
    },
    {
        "name": "Тринидадитобаговна Ж.Щ.",
        "post": "Ямщик",
        "year": 2002
    }
]
```

Рисунок 2 – Выгружаемые программой данные

```
Run: primer1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.16/primer1.py
>>> list
Список работников пуст.
>>> load_workers.txt
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Вальдемаров Х.Ц. | Египтолог | 2006 |
| 2 | Тринидадитобаговна Ж.Щ. | Ямщик | 2002 |
+-----+-----+-----+-----+
>>>
```

Рисунок 3 – Загрузка данных json

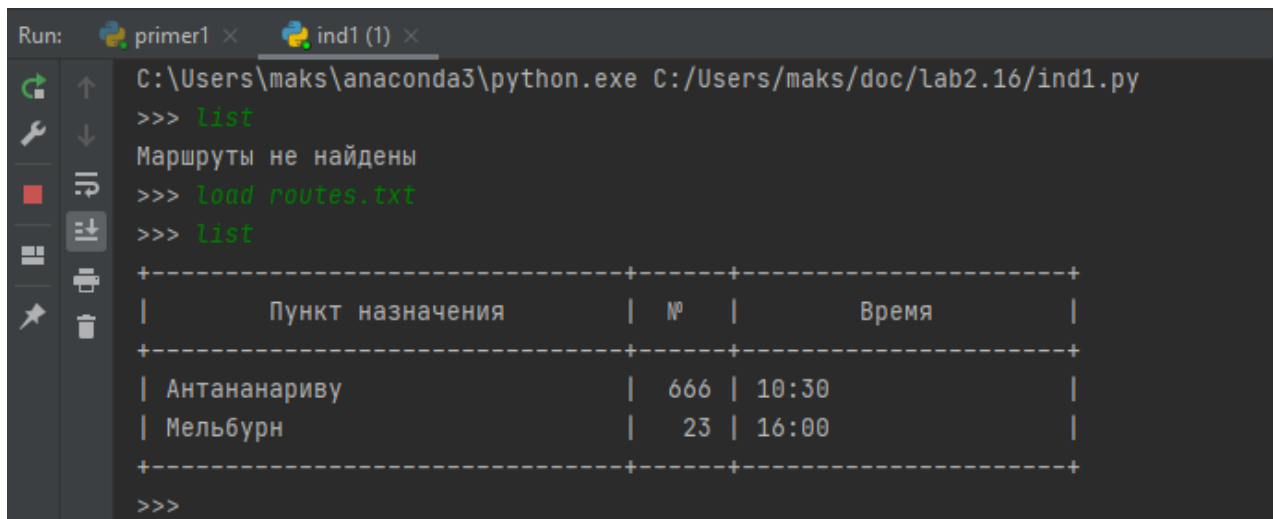
Выполнено индивидуальное задание варианта 6, рис. 4-6

```
Run: primer1 x ind1 (1) x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.16/ind1.py
>>> add
Пункт назначения? Антананариву
Номер поезда? 666
Время отправления?(формат чч:мм) 10:30
>>> add
Пункт назначения? Мельбурн
Номер поезда? 23
Время отправления?(формат чч:мм) 16:00
>>> list
+-----+-----+-----+-----+
| Пункт назначения | № | Время |
+-----+-----+-----+-----+
| Антананариву | 666 | 10:30 |
| Мельбурн | 23 | 16:00 |
+-----+-----+-----+-----+
>>> save_routes.txt
>>>
```

Рисунок 4 – Индивидуальное задание

```
routes.txt – Блокнот
Файл Правка Формат Вид Справка
[
  {
    "destination": "Антананариву",
    "number": "666",
    "time": "10:30"
  },
  {
    "destination": "Мельбурн",
    "number": "23",
    "time": "16:00"
  }
]
```

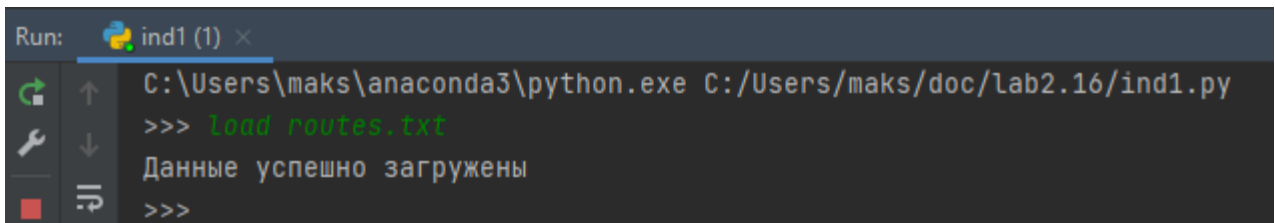
Рисунок 5 – Выгружаемые программой данные



```
Run: primer1 x ind1 (1) x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.16/ind1.py
>>> list
Маршруты не найдены
>>> load routes.txt
>>> list
+-----+-----+-----+
| Пункт назначения | № | Время |
+-----+-----+-----+
| Антананариву    | 666 | 10:30 |
| Мельбурн        | 23  | 16:00 |
+-----+-----+-----+
>>>
```

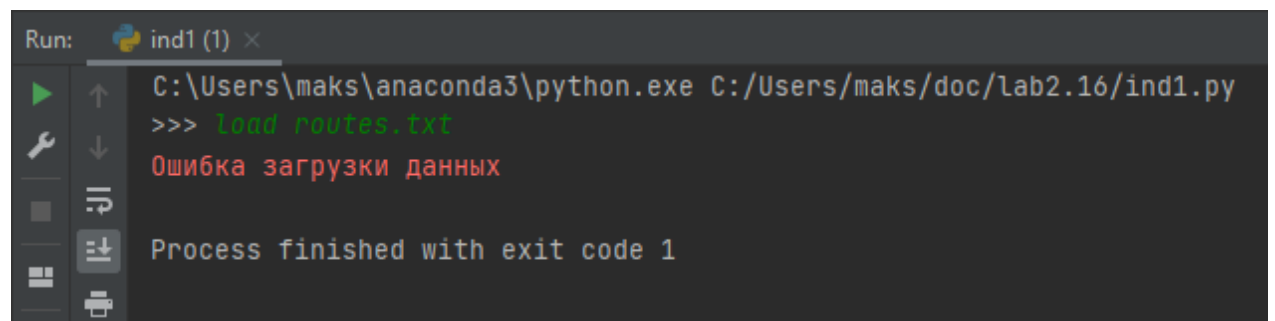
Рисунок 6 – Загрузка данных json

Добавлена часть кода, выполняющая проверку загружаемых данных с помощью json schema, рис. 7-8



```
Run: ind1 (1) x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.16/ind1.py
>>> load routes.txt
Данные успешно загружены
>>>
```

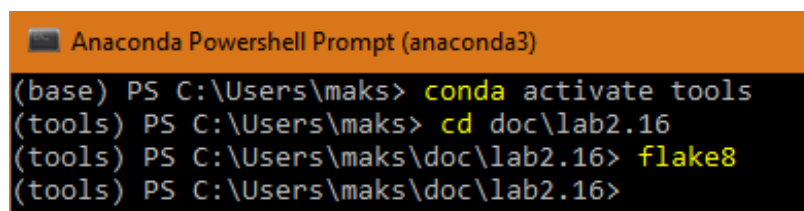
Рисунок 7 – Успешная загрузка



```
Run: ind1 (1) x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.16/ind1.py
>>> load routes.txt
Ошибка загрузки данных
Process finished with exit code 1
```

Рисунок 8 – Ошибка при загрузке

Программы проверены на flake8, рис. 9



```
Anaconda Powershell Prompt (anaconda3)
(base) PS C:\Users\maks> conda activate tools
(tools) PS C:\Users\maks> cd doc\lab2.16
(tools) PS C:\Users\maks\doc\lab2.16> flake8
(tools) PS C:\Users\maks\doc\lab2.16>
```

Рисунок 9 – Проверка заданий

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 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.

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

- `json5.load()`;
- `json5.loads()`;
- `json5.tool()`;
- `json5.dump()`;
- `json5.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 кириллицу?

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 занятия были приобретены навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x