

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

Кафедра инфокоммуникаций

**Отчет по лабораторной работе №2.16
Работа с данными формата JSON в языке Python
по дисциплине «Технологии программирования»**

Выполнил студент группы ИВТ-б-о-20-1

Ищенко Т.С. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверила Воронкин Р.А. _____

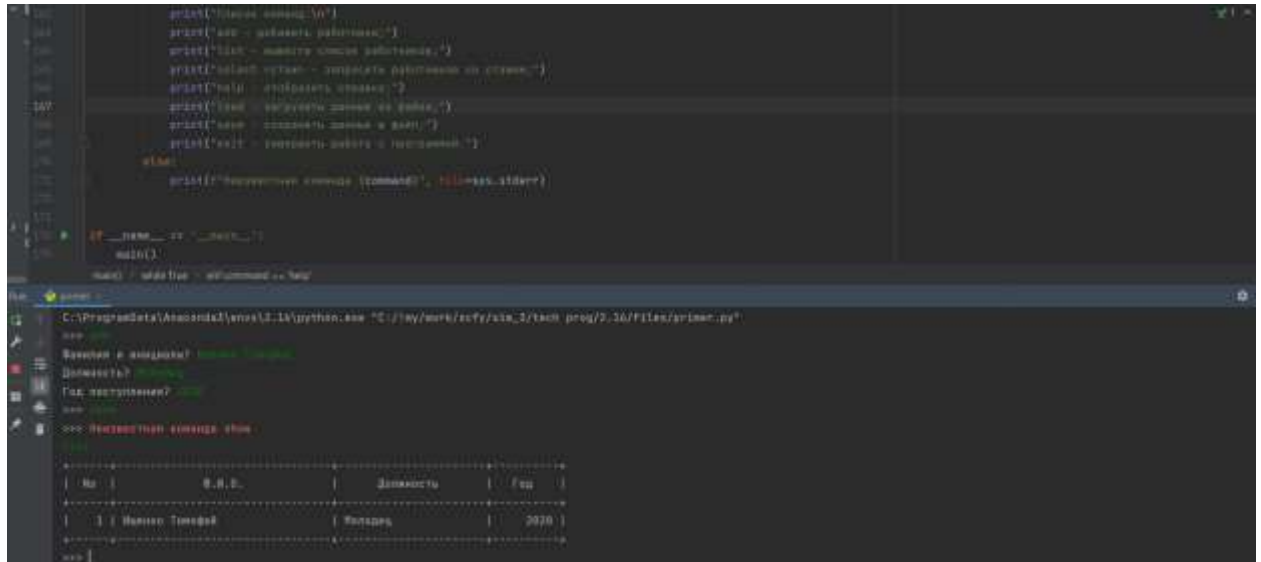
(подпись)

Ставрополь 2021

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

Ход работы:

1. Произвёл отработку примера, согласно методическим рекомендациям



```
140: print("Список команд:\n")
141: print("add - добавить станцию.")
142: print("show - вывести список станций.")
143: print("select <название> - запросить станции по названию.")
144: print("help - отобразить справку.")
145: print("load - загрузить данные из файла.")
146: print("save - сохранить данные в файл.")
147: print("exit - завершить работу с программой.")
148:
149: while True:
150:     print(f">>> {command}")
151:     if command == "add":
152:         station = input("Название станции: ")
153:         stations.append(station)
154:         print("Станция добавлена.")
155:     elif command == "show":
156:         for station in stations:
157:             print(station)
158:     elif command == "select":
159:         name = input("Название станции: ")
160:         for station in stations:
161:             if station == name:
162:                 print(station)
163:     elif command == "help":
164:         print("Список команд:\n")
165:         print("add - добавить станцию.")
166:         print("show - вывести список станций.")
167:         print("select <название> - запросить станции по названию.")
168:         print("help - отобразить справку.")
169:         print("load - загрузить данные из файла.")
170:         print("save - сохранить данные в файл.")
171:         print("exit - завершить работу с программой.")
172:     elif command == "load":
173:         file = input("Путь к файлу: ")
174:         with open(file, "r") as f:
175:             data = json.load(f)
176:             stations = data["stations"]
177:             for station in stations:
178:                 print(station)
179:     elif command == "save":
180:         file = input("Путь к файлу: ")
181:         with open(file, "w") as f:
182:             json.dump(stations, f)
183:     elif command == "exit":
184:         break
185:
186: if __name__ == "__main__":
187:     main()
```

С:\ProgramData\Anaconda3\envs\3.11\python.exe "C:/my/work/scf/sia_2/task prog/2.16/files/prim.py"

>>> add

Название станции? Москва Тимирязевская

Данные? [{"station": "Москва Тимирязевская"}]

Где сохранение? 1

>>> show

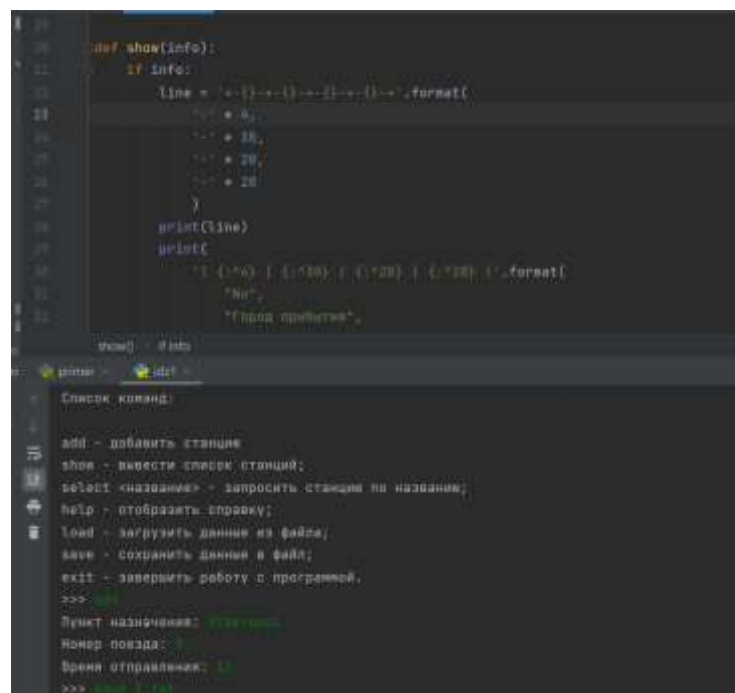
Список команд:

№	Станция	Данные	Файл
1	Москва Тимирязевская	[{"station": "Москва Тимирязевская"}]	1

>>>

Рисунок 1 – Результат выполнения примера

2. Произвёл выполнение индивидуального задания



```
19: def show(info):
20:     if info:
21:         line = "{:10} | {:10} | {:10} | {:10} |".format(
22:             "№", "Станция", "Данные", "Файл"
23:         )
24:         print(line)
25:         for station in stations:
26:             print(station)
27:     else:
28:         print("No")
29:         print("Формат: {station} | {data} | {file}")
```

show() = 0 info

Список команд:

add - добавить станцию

show - вывести список станций

select <название> - запросить станции по названию

help - отобразить справку

load - загрузить данные из файла

save - сохранить данные в файл

exit - завершить работу с программой

>>> show(1, 1)

Пункт назначения: Москва

Номер поезда: 1

Время отправления: 11

>>>

Рисунок 2 – Результат выполнения индивидуального задания

3. Выполнил задание повышенной сложности, согласно методическим рекомендациям

```
75
76
77 def cheking_file(file):
78     with open('params.json') as params:
79         schema = json.load(params)
80         checker = jsonschema.Draft7Validator(schema)
81     try:
82         if not checker.validate(file):
83             print('Ошибка нет')
84     except jsonschema.exceptions.ValidationError:
85         print('Ошибка', file=sys.stderr)
86         exit(1)
87
88
89 cheking_file()  except jsonschema.exceptions.Va...
primer x idz1 x idz2 x
C:\ProgramData\Anaconda3\envs\2.16\python.exe "C:/!my/work/scfy/sim_3/tech prog/2.16/files/idz2/idz2.py"
>>> from example import
Ошибка нет
>>> show
+-----+-----+-----+-----+
| No | Город прибытия | Номер поезда | Вреня отправления |
+-----+-----+-----+-----+
| 1 | stavoropol | 5 | 11 |
+-----+-----+-----+-----+
>>> |
```

Рисунок 3 – Результат выполнения задания повышенной сложности

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.

5. 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 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 выполнения лабораторной работы приобрел навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.