Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
дисциплины «Анализ данных»
Вариант 9

Выполнил:
Дудкин Константин Александрович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»
направление «Программное
обеспечение средств вычислительной
техники и автоматизированных
систем»,
очная форма обучения

_____
(подпись)

Руководитель практики:
Кандидат технических наук, доцент
кафедры инфокомуникаций, доцент
Воронкин Роман Александрович

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2024 г.

Тема: Работа с данными формата JSON в языке Python

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

Порядок выполнения работы

Проработал пример, указанный в методических материалах:

Рисунок 1. Пример (часть 1)

```
38          print(line)
39          print(
40              '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
41                  *args: "№",
42                  "Ф.И.О.",
43                  "Должность",
44                  "Год"
45              )
46          )
47          print(line)
48          # Вывести данные о всех сотрудниках.
49          for idx, worker in enumerate(staff, 1):
50              print(
51                  '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
52                      *args: idx,
53                      worker.get('name', ''),
54                      worker.get('post', ''),
55                      worker.get('year', 0)
56                  )
57              )
58          print(line)
59
60      else:
61          print("Список работников пуст.")
62
63
    1 usage
64  def select_workers(staff, period):
65      """
66      Выбрать работников с заданным стажем.
67      """
68      # Получить текущую дату.
69      today = date.today()
70      # Сформировать список работников.
71      result = []
72      for employee in staff:
73          if today.year - employee.get('year', today.year) >= period:
```

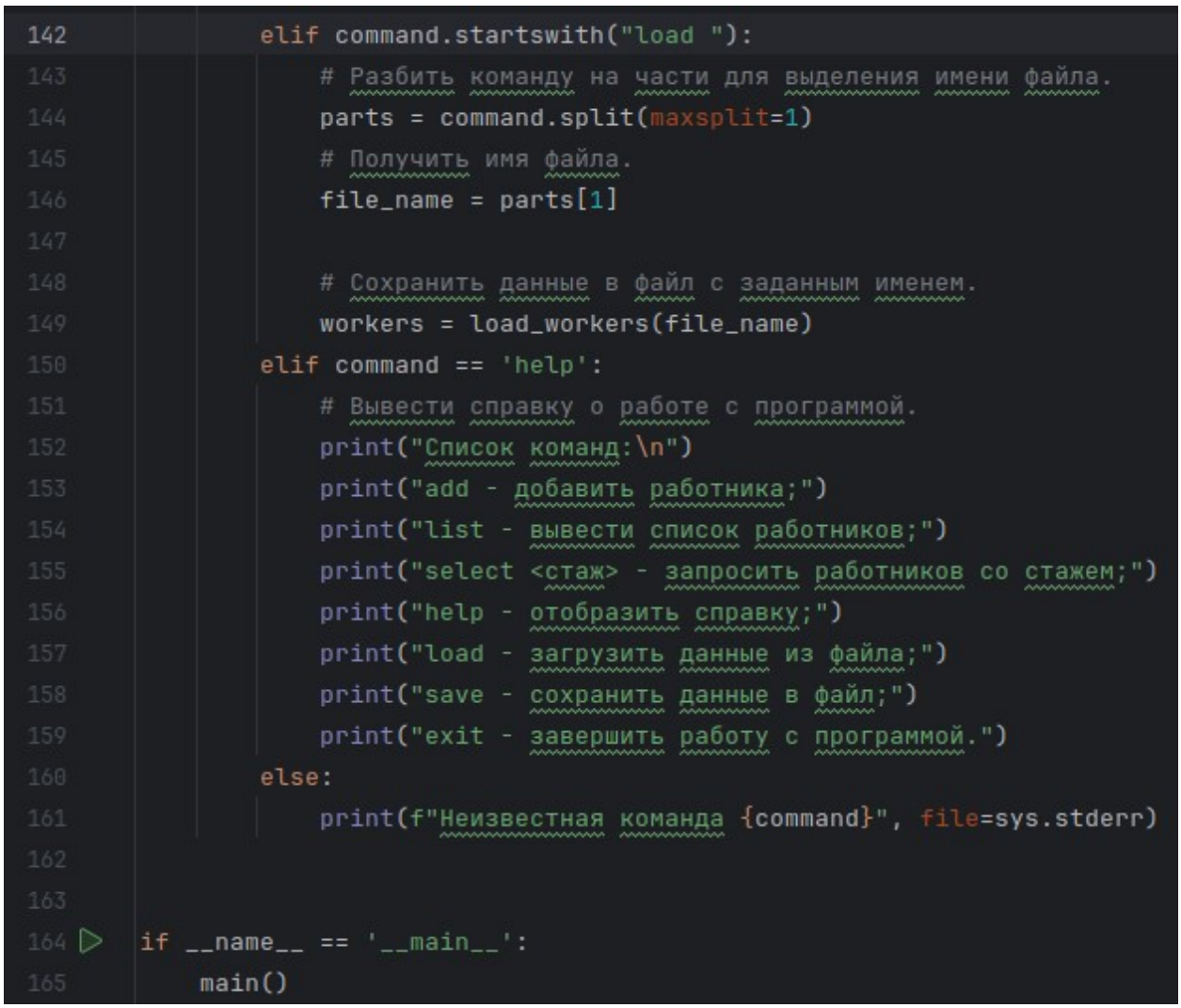
Рисунок 2. Пример (часть 2)

```
74                result.append(employee)
75
76            # Возвратить список выбранных работников.
77            return result
78
79
    1 usage
80    def save_workers(file_name, staff):
81        """
82        Сохранить всех работников в файл JSON.
83        """
84        # Открыть файл с заданным именем для записи.
85        with open(file_name, "w", encoding="utf-8") as fout:
86            # Выполнить сериализацию данных в формат JSON.
87            # Для поддержки кирилицы установим ensure_ascii=False
88            json.dump(staff, fout, ensure_ascii=False, indent=4)
89
90
    1 usage
91    def load_workers(file_name):
92        """
93        Загрузить всех работников из файла JSON.
94        """
95        # Открыть файл с заданным именем для чтения.
96        with open(file_name, "r", encoding="utf-8") as fin:
97            return json.load(fin)
98
99
100   def main():
101       """
102       Главная функция программы.
103       """
104       # Список работников.
105       workers = []
106       # Организовать бесконечный цикл запроса команд.
```

Рисунок 3. Пример (часть 3)

```
107     while True:
108         # Запросить команду из терминала.
109         command = input(">>> ").lower()
110         # Выполнить действие в соответствие с командой.
111         if command == "exit":
112             break
113         elif command == "add":
114             # Запросить данные о работнике.
115             worker = get_worker()
116             # Добавить словарь в список.
117             workers.append(worker)
118             # Отсортировать список в случае необходимости.
119             if len(workers) > 1:
120                 workers.sort(key=lambda item: item.get('name', ''))
121         elif command == "list":
122             # Отобразить всех работников.
123             display_workers(workers)
124         elif command.startswith("select "):
125             # Разбить команду на части для выделения стажа.
126             parts = command.split(maxsplit=1)
127             # Получить требуемый стаж.
128             period = int(parts[1])
129             # Выбрать работников с заданным стажем.
130             selected = select_workers(workers, period)
131             # Отобразить выбранных работников.
132             display_workers(selected)
133         elif command.startswith("save "):
134             # Разбить команду на части для выделения имени файла.
135             parts = command.split(maxsplit=1)
136             # Получить имя файла.
137             file_name = parts[1]
138 
139             # Сохранить данные в файл с заданным именем.
140             save_workers(file_name, workers)
141 
```

Рисунок 4. Пример (часть 4)



```
142             elif command.startswith("load "):
143                 # Разбить команду на части для выделения имени файла.
144                 parts = command.split(maxsplit=1)
145                 # Получить имя файла.
146                 file_name = parts[1]
147 
148                 # Сохранить данные в файл с заданным именем.
149                 workers = load_workers(file_name)
150             elif command == 'help':
151                 # Вывести справку о работе с программой.
152                 print("Список команд:\n")
153                 print("add - добавить работника;")
154                 print("list - вывести список работников;")
155                 print("select <стаж> - запросить работников со стажем;")
156                 print("help - отобразить справку;")
157                 print("load - загрузить данные из файла;")
158                 print("save - сохранить данные в файл;")
159                 print("exit - завершить работу с программой.")
160             else:
161                 print(f"Неизвестная команда {command}", file=sys.stderr)
162 
163 
164 if __name__ == '__main__':
165     main()
```

Рисунок 5. Пример (часть 5)

Выполнил индивидуальное задание: для лабораторной работы 2.8 сделал дополнительно возможность записывать все данные в отдельный JSON-файл:

```
16 
   1 usage  new *
17 def export_to_json(file, routes_list):
18     with open(file, 'w', encoding='utf-8') as fileout:
19         json.dump(routes_list, fileout, ensure_ascii=False, indent=4)
20 
21 
   1 usage  new *
22 def import_json(file):
23     with open(file, 'r', encoding='utf-8') as filein:
24         return json.load(filein)
25 
```

Рисунок 6. Добавленный код (часть 1)

```python
elif command == 'export':
    export_to_json( file: 'individual.json', routes)

elif command.startswith('import '):
    parts = command.split(maxsplit=1)
    file = parts[1]
    routes = import_json(file)
```

Рисунок 7. Добавленный код (часть 2)

```
/usr/bin/python3.11 /home/code_ra1der/git/DataAnalysis_LW2/Progs/Individual.py
>>> add
Первая точка маршрута: Ставрополь
Вторая точка маршрута: Ессентуки
>>> add
Первая точка маршрута: Петербург
Вторая точка маршрута: Москва
>>> list
+----------------+----------------------+----------------------+
| Номер маршрута |    Место отправки    |    Место прибытия    |
+----------------+----------------------+----------------------+
| 1              | Ставрополь           | Ессентуки            |
+----------------+----------------------+----------------------+
| 2              | Петербург            | Москва               |
+----------------+----------------------+----------------------+
>>> export
>>> exit

Process finished with exit code 0
```

Рисунок 8. Проверка экспортирования

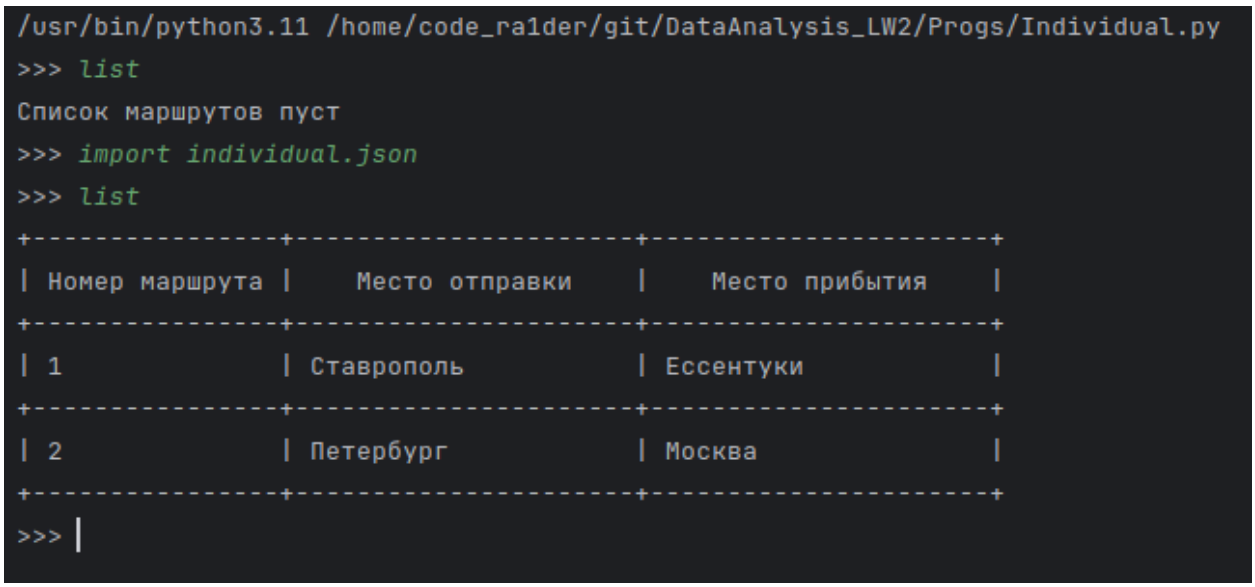


Рисунок 9. Проверка импортирования

Ответы на вопросы

1. JSON (JavaScript Object Notation) используется для обмена данными между программами. Обычно в них хранятся необходимые данные и конфигурации для работы программ

2. В JSON используются следующие типы значений:

- Строковые
- Численные
- Логические
- Массивы
- Объекты
- Null

3. Для этого в JSON обычно используется индексация посредством ключей и категорий

4. JSON5 — расширение формата JSON, улучшающий его функционал посредством добавления множества дополнительных объектов, облегчающих форматирование и чтение данных с файла

5. Для работы с данными в формате JSON5 на языке Python можно использовать сторонние библиотеки, как например json5

6. В языке Python есть встроенная библиотека json, позволяющая работать с данным типом файлов

7. Отличие этих двух команд заключается в том, что json.dump() записывает данные JSON в файл, а json.dumps() преобразует их в строку

8. Для десериализации данных из JSON можно использовать команды json.load() и json.loads(), чей функционал схож с json.dump() и json.dumps()

9. Для этого необходимо убедиться в том, что файл JSON имеет кодировку UTF-8

10. JSON Schema — спецификация, описывающая структуру и ограничения данных в формате JSON. Схема определяет типы данных и допустимые их ограничения и значения

Вывод: в ходе данной работе были приобретены навыки по работе с JSON-файлами, а также по включении их в язык Python