

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 ПРАКТИЧЕСКОЙ РАБОТЕ №2
дисциплины «Анализ данных»

Выполнил:
Степанов Леонид Викторович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение
средств вычислительной
техники и автоматизирование
систем», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент, доцент кафедры
инфокоммуникаций

(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.

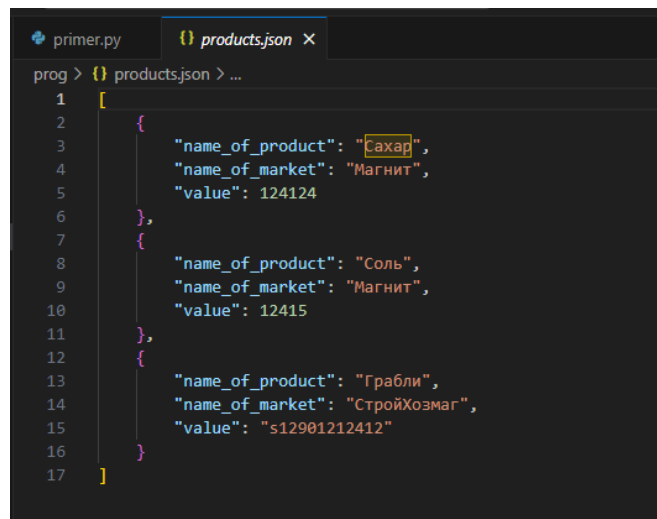
Ход работы:

1. Создал файл (primer.py), в котором проработал пример лабораторной работы на рис. 1 расположен ввод-вывод, а на рис. 2 содержимое созданного json файла:

```
(.venv) PS C:\Users\student-09-326\Desktop\Новая папка\lab2> python .\prog\primer.py
>>> add
Фамилия и инициалы? Никифоров А.М.
Должность? хирург
Год поступления? 2000
>>> add
Фамилия и инициалы? Литвинов С.С.
Должность? Хирург-Кардиолог
Год поступления? 2010
>>> add
Фамилия и инициалы? Сединцов С.С.
Должность? педиатр
Год поступления? 2015
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Литвинов С.С. | Хирург-Кардиолог | 2010 |
| 2 | Никифоров А.М. | хирург | 2000 |
| 3 | Сединцов С.С. | педиатр | 2015 |
+-----+-----+-----+-----+
>>> select 10
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Литвинов С.С. | Хирург-Кардиолог | 2010 |
| 2 | Никифоров А.М. | хирург | 2000 |
+-----+-----+-----+-----+
>>> save bolnica.json
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
load - загрузить данные из файла;
save - сохранить данные в файл;
exit - завершить работу с программой.
>>> 
```

Рисунок 1 – Результат работы primer.py

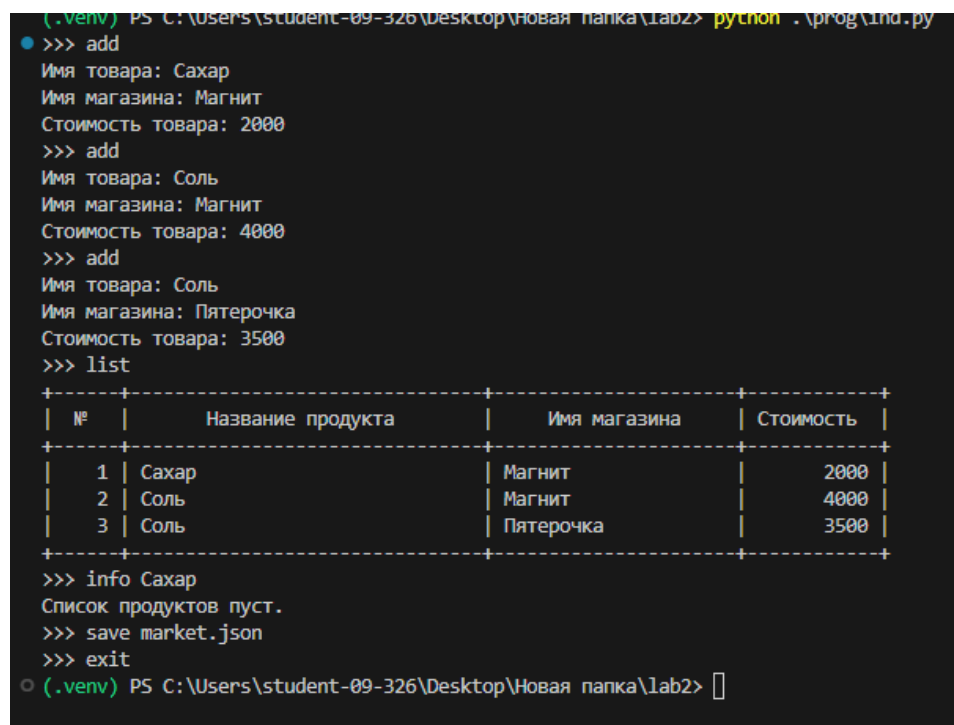


```
1 [
2   {
3     "name_of_product": "Сахар",
4     "name_of_market": "Магнит",
5     "value": 124124
6   },
7   {
8     "name_of_product": "Соль",
9     "name_of_market": "Магнит",
10    "value": 12415
11  },
12  {
13    "name_of_product": "Гроби",
14    "name_of_market": "СтройХозмаг",
15    "value": "s12901212412"
16  }
17 ]
```

Рисунок 2 – Json файл

2. Индивидуальное 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.

Создал файл (ind.py), в котором выполнил индивидуальное задание в соответствии с лабораторной 2.8, на рис. 3 указан результат выполнения команд, а на рис. 4 содержимое созданного json файла.

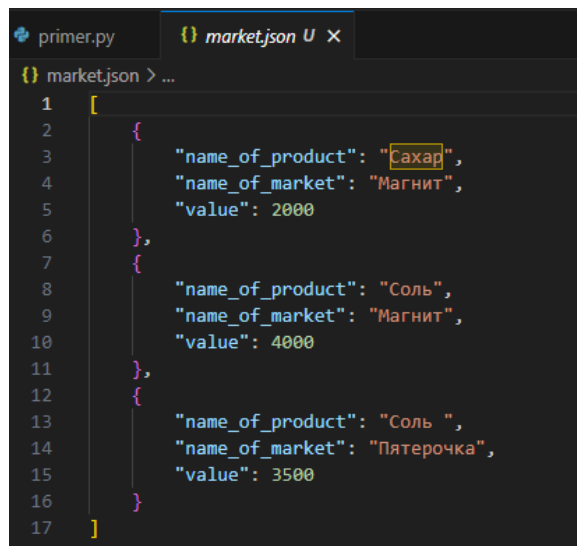


```
(.venv) PS C:\Users\student-09-326\Desktop\Новая папка\lab2> python .\prog\ind.py
>>> add
Имя товара: Сахар
Имя магазина: Магнит
Стоимость товара: 2000
>>> add
Имя товара: Соль
Имя магазина: Магнит
Стоимость товара: 4000
>>> add
Имя товара: Соль
Имя магазина: Пятерочка
Стоимость товара: 3500
>>> list
```

№	Название продукта	Имя магазина	Стоимость
1	Сахар	Магнит	2000
2	Соль	Магнит	4000
3	Соль	Пятерочка	3500

```
>>> info Сахар
Список продуктов пуст.
>>> save market.json
>>> exit
(.venv) PS C:\Users\student-09-326\Desktop\Новая папка\lab2>
```

Рисунок 3 – Результат работы ind.py



```

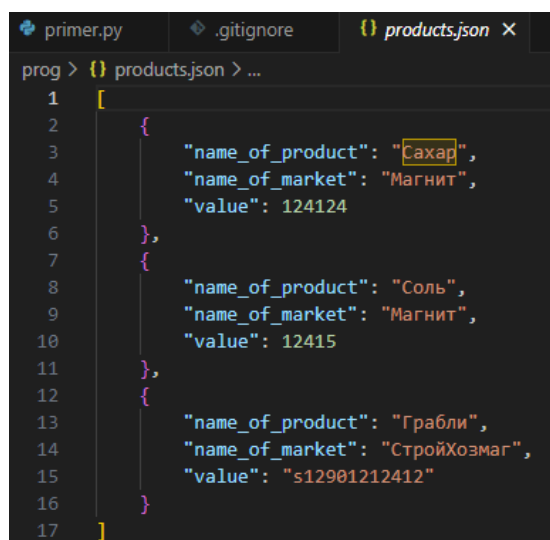
1  [
2    {
3      "name_of_product": "Сахар",
4      "name_of_market": "Магнит",
5      "value": 2000
6    },
7    {
8      "name_of_product": "Соль",
9      "name_of_market": "Магнит",
10     "value": 4000
11   },
12   {
13     "name_of_product": "Соль ",
14     "name_of_market": "Пятерочка",
15     "value": 3500
16   }
17 ]

```

Рисунок 4 – Json файл

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.

Входные данные (products.json):



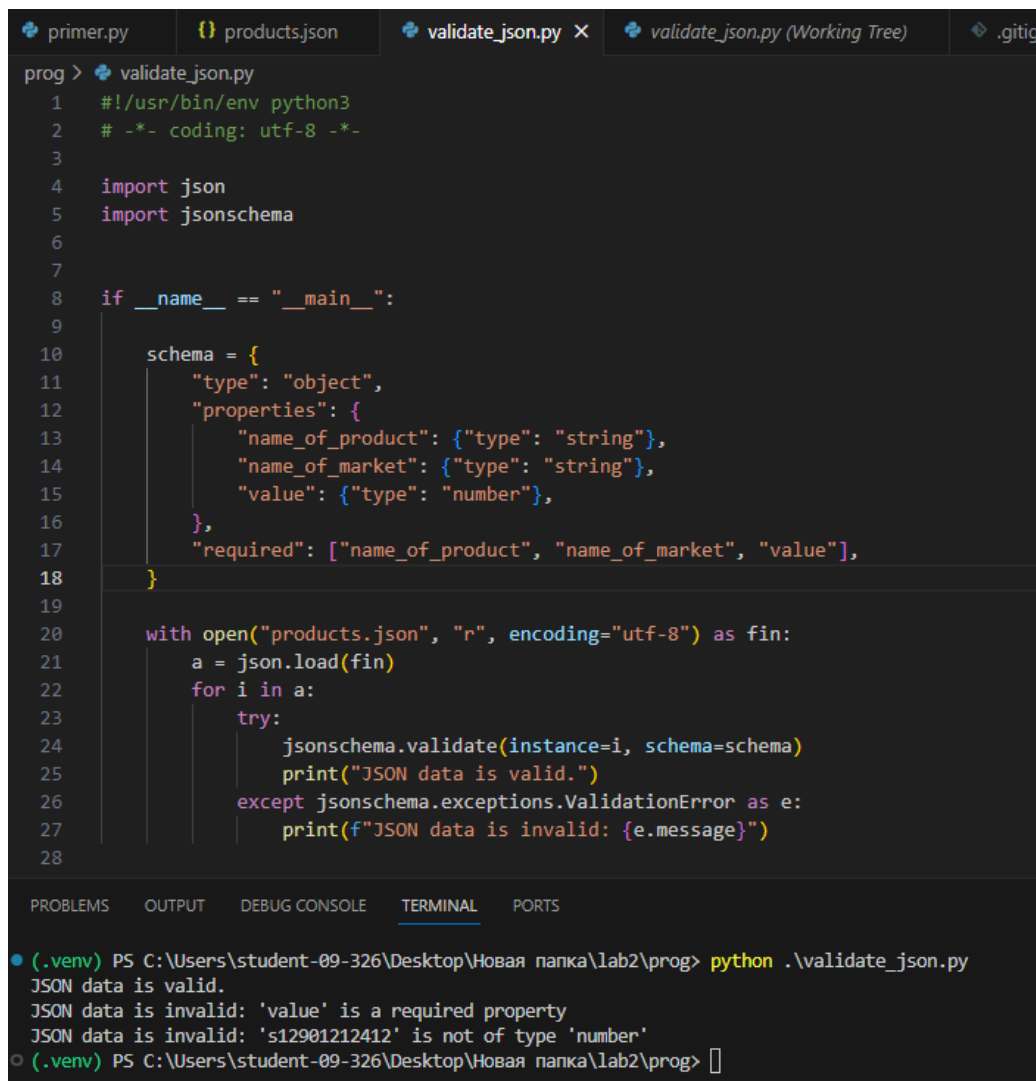
```

1  [
2    {
3      "name_of_product": "Сахар",
4      "name_of_market": "Магнит",
5      "value": 124124
6    },
7    {
8      "name_of_product": "Соль",
9      "name_of_market": "Магнит",
10     "value": 12415
11   },
12   {
13     "name_of_product": "Грабли",
14     "name_of_market": "СтройХозмаг",
15     "value": "s12901212412"
16   }
17 ]

```

Рисунок 5 – Входные данные

Результат валидации (рис.6):



The image shows a code editor with a file named `validate_json.py` open. The script defines a JSON schema and validates data from `products.json`. The terminal output shows the results of the validation.

```
primer.py  products.json  validate_json.py X  validate_json.py (Working Tree)  .gitig
prog > validate_json.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import json
5  import jsonschema
6
7
8  if __name__ == "__main__":
9
10     schema = {
11         "type": "object",
12         "properties": {
13             "name_of_product": {"type": "string"},
14             "name_of_market": {"type": "string"},
15             "value": {"type": "number"},
16         },
17         "required": ["name_of_product", "name_of_market", "value"],
18     }
19
20     with open("products.json", "r", encoding="utf-8") as fin:
21         a = json.load(fin)
22         for i in a:
23             try:
24                 jsonschema.validate(instance=i, schema=schema)
25                 print("JSON data is valid.")
26             except jsonschema.exceptions.ValidationError as e:
27                 print(f"JSON data is invalid: {e.message}")
28
```

Terminal Output:

```
(.venv) PS C:\Users\student-09-326\Desktop\Новая папка\lab2\prog> python .\validate_json.py
JSON data is valid.
JSON data is invalid: 'value' is a required property
JSON data is invalid: 's12901212412' is not of type 'number'
(.venv) PS C:\Users\student-09-326\Desktop\Новая папка\lab2\prog>
```

Рисунок 6 – Результат программы `validate_json.py`

Вывод: в ходе выполнения работы были приобретены навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.