

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 РАБОТЕ №2.16**  
**дисциплины «Программирование на языке Python»**

Выполнил:  
Хачатрян Владимир Владимирович  
2 курс, группа ИТС-б-о-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», очная форма обучения

---

(подпись)

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

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** работа с данными формата JSON в языке Python.

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

### Выполнение работы:

Изучил теоретический материал работы, создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами.

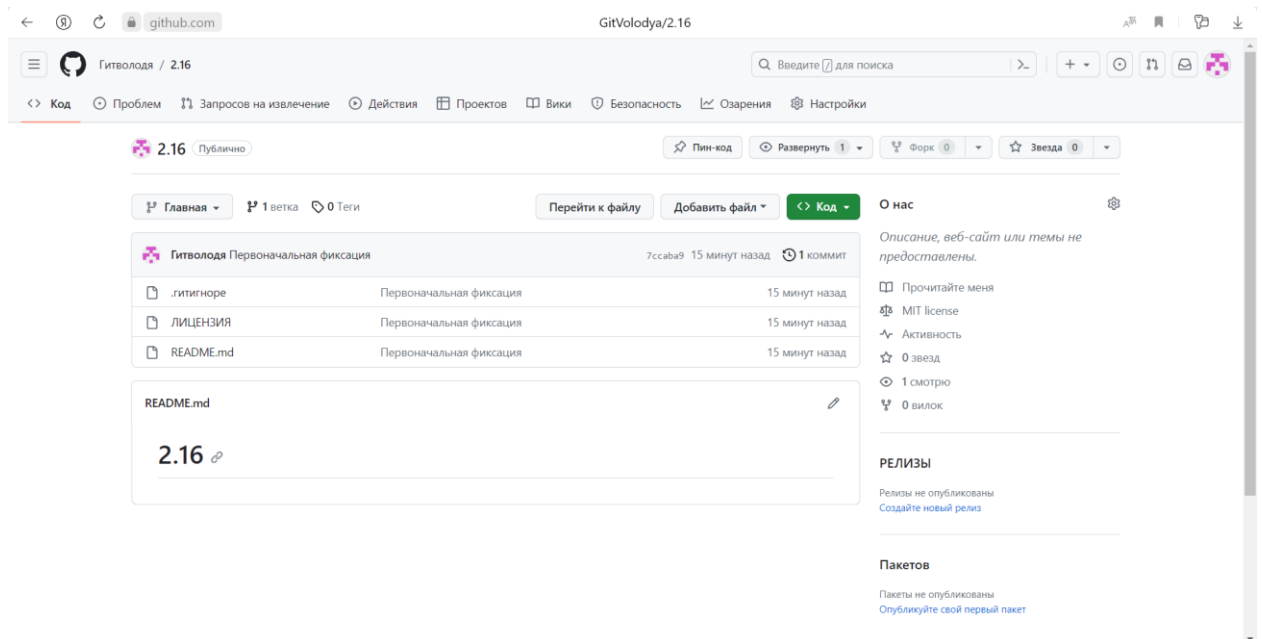


Рисунок 1. Репозиторий

Клонировал репозиторий на компьютер. Организовал свой репозиторий в соответствии с моделью ветвления git-flow, появилась новая ветка develop.

```
C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/Lab.Rab.2.16.git
Cloning into 'Lab.Rab.2.16'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\HUAWEI>cd C:\Users\HUAWEI\Lab.Rab.2.16

C:\Users\HUAWEI\Lab.Rab.2.16>git init flow
Initialized empty Git repository in C:/Users/HUAWEI/Lab.Rab.2.16/flow/.git/

C:\Users\HUAWEI\Lab.Rab.2.16>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/HUAWEI/Lab.Rab.2.16/.git/hooks]
```

Рисунок 2. Клонирование и модель ветвления git-flow

Создал виртуальное окружение (ВО) Miniconda и активировал его, а также установил необходимые пакеты isort, black, flake8.

```
(base) PS C:\Users\vovax> cd C:\Users\vovax\2.16  
(base) PS C:\Users\vovax\2.16> conda create -n 2.16 python=3.11
```

Рисунок 3. Создание ВО

```
(base) PS C:\Users\vovax\2.16> conda activate 2.16  
(2.16) PS C:\Users\vovax\2.16> conda install -c conda-forge black
```

Рисунок 4. Установка пакета black

```
(2.16) PS C:\Users\vovax\2.16> conda install -c conda-forge flake8
```

Рисунок 5. Установка пакета flake8

```
(2.16) PS C:\Users\vovax\2.16> conda install -c conda-forge isort
```

Рисунок 6. Установка пакета isort

Пакет isort является инструментом для автоматической сортировки импортов в Python-кодах. Он используется для удобства чтения и поддержания порядка в коде.

Пакет black представляет инструмент автоматического форматирования кода для языка Python. Он помогает обеспечить единообразие стиля кодирования в проекте и улучшает читаемость кода.

Пакет flake8 отвечает за статический анализ и проверку Python-кода. Он проводит проверку на соответствие стилю кодирования PEP 8, а также наличие потенциальных ошибок и проблемных паттернов в коде.

## Пример №1.

### Условие примера: (2.8)

**Пример 1.** *Использовать словарь, содержащий следующие ключи: фамилия и инициалы работника; название занимаемой должности; год поступления на работу. Написать программу, выполняющую следующие действия:*

- ввод с клавиатуры данных в список, состоящий из заданных словарей;
- записи должны быть размещены по алфавиту;
- вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;
- если таких работников нет, вывести на дисплей соответствующее сообщение.

*Решение:* Определим следующие ключи для словарей:

- *name* - фамилия и инициалы работника;
- *post* - название занимаемой должности;
- *year* - год поступления.

Введем следующие команды для работы со списком словарей в интерактивном режиме:

- *add* - запросить информацию о сотруднике с клавиатуры и добавить в список, поддерживая список в отсортированном состоянии;
- *list* - вывести на экран содержимое списка словарей;
- *select* - вывести на дисплей фамилий работников, чей стаж работы в организации превышает заданное значение, при этом это значение должно быть аргументом команды *select* и отделено от нее пробелом;
- *help* - вывести на дисплей список команд с описанием;
- *exit* - завершить работу программы.

Для примера 1 лабораторной работы 2.8 добавьте возможность сохранения списка в файл формата JSON и чтения данных из файла JSON.

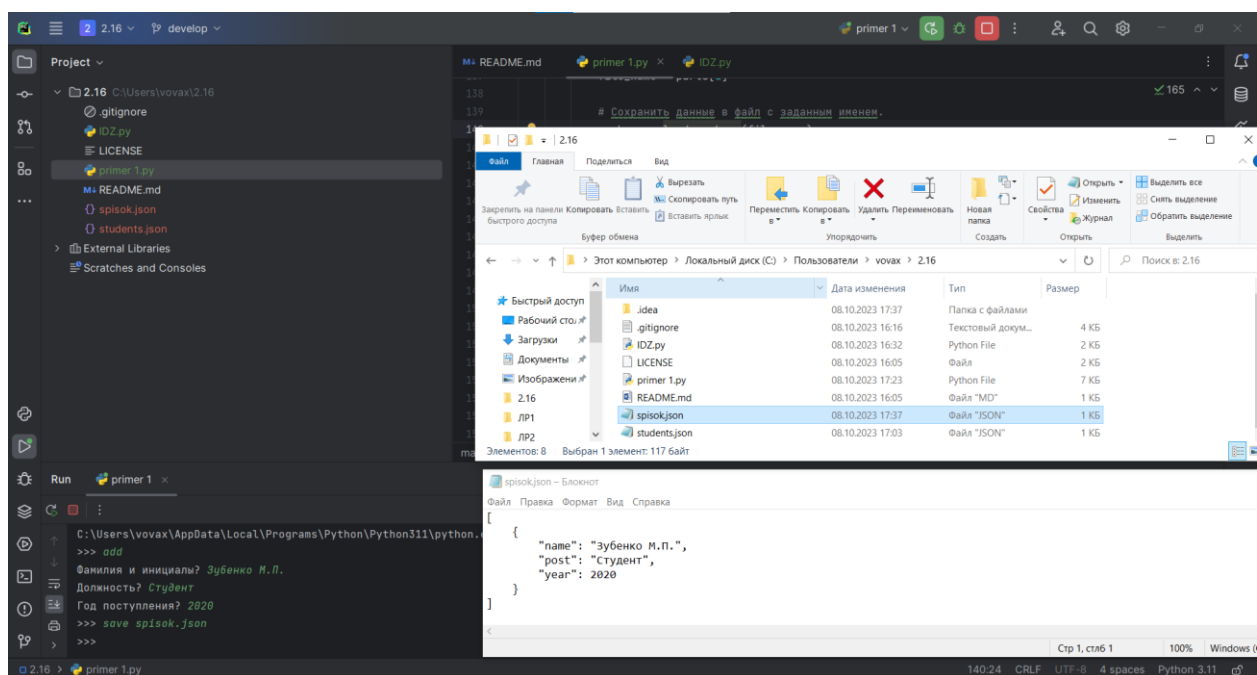


Рисунок 7. Работа с примером №1.

### Выполнение индивидуального задания.

Для своего варианта лабораторной работы 2.8 необходимо дополнительно реализовать сохранение и чтение данных из файла формата JSON.

**Условие задания:** Использовать словарь, содержащий следующие ключи: фамилия и инициалы; номер группы; успеваемость (список из пяти элементов). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по алфавиту; вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2; если таких студентов нет, вывести соответствующее сообщение.

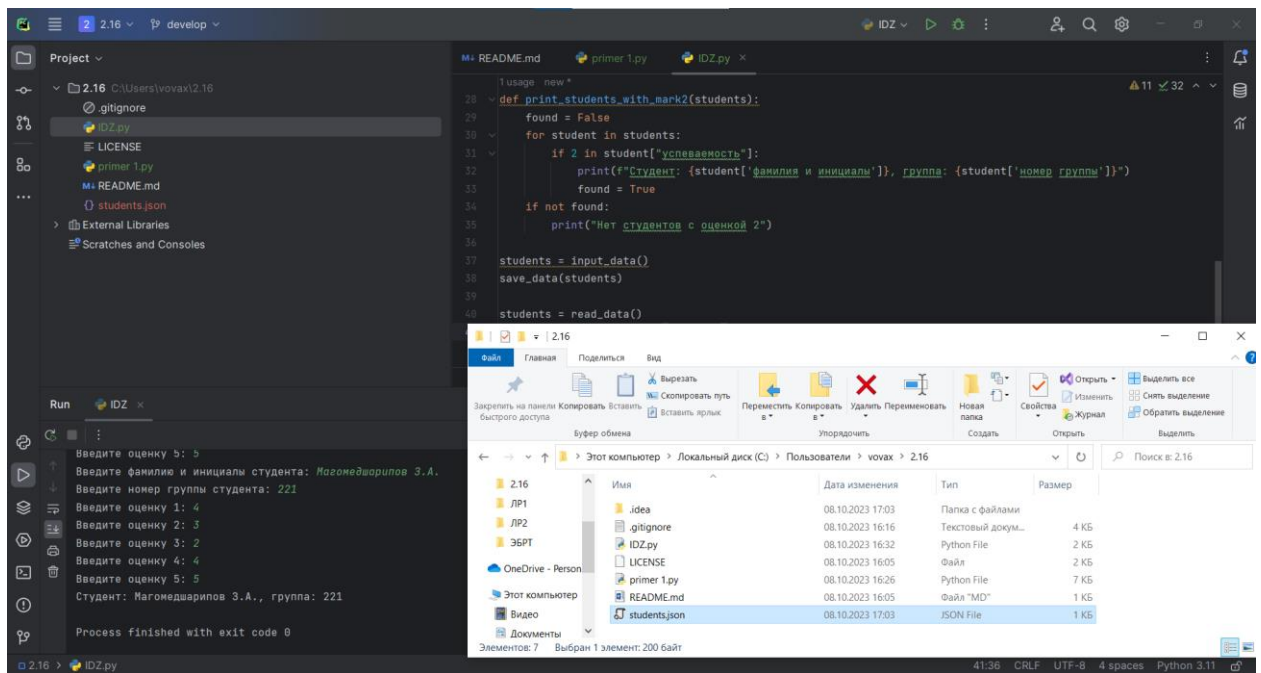


Рисунок 8. Выполнение индивидуального задания

Деактивировал виртуальное окружение.

```

(2.16) PS C:\Users\vovax\2.16> conda deactivate
(base) PS C:\Users\vovax\2.16>

```

Рисунок 9. Деактивация ВО

Слил ветку develop с веткой main и отправил на удаленный сервер.

```
C:\Users\vovax\2.16>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 343 bytes | 343.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/GitVolodya/2.16.git
  575c0ff..341c35b  main -> main
C:\Users\vovax\2.16>_
```

Рисунок 8. Слияние веток

Ссылка на репозиторий: <https://github.com/GitVolodya/2.16.git>

### Контрольные вопросы:

1. Для чего используется JSON?

JSON (JavaScript Object Notation) используется для хранения и обмена данными между клиентом и сервером в удобном для чтения и записи формате. Он является текстовым форматом, основанным на синтаксисе JavaScript, и может быть использован в различных языках программирования.

2. Какие типы значений используются в JSON?

В JSON используются следующие типы значений:

- Строки (в двойных кавычках)
- Числа (целые числа или числа с плавающей точкой)
- Логические значения (true или false)
- Массивы (упорядоченные списки значений)
- Объекты (наборы пар ключ-значение)

3. Как организована работа со сложными данными в JSON?

Для работы со сложными данными в JSON используются массивы и объекты. Массивы используются для хранения упорядоченного списка значений, а объекты представляют набор пар ключ-значение, где ключи являются строками, а значения могут быть любого типа.



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

JSON5 - это расширение формата данных JSON, которое добавляет некоторые удобные функции и возможности, такие как поддержка комментариев и необязательные запятые в конце списка. Основное отличие JSON5 от JSON заключается в дополнительном синтаксисе и расширенных возможностях для удобства разработки.

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

Для работы с данными в формате JSON5 в языке Python можно использовать библиотеку `json5`.

*6. Какие средства предоставляет язык Python для сериализации данных в формате JSON?*

Язык Python предоставляет модуль `json` для сериализации (преобразования объектов Python в формат JSON) и десериализации (преобразования данных JSON в объекты Python) данных.

*7. В чем отличие функций `json.dump()` и `json.dumps()`?*

Функция `json.dump()` используется для напрямую записи данных JSON в файл, в то время как функция `json.dumps()` возвращает строковое представление данных JSON, которое можно сохранить в переменной или передать дальше для обработки.

*8. Какие средства предоставляет язык Python для десериализации данных из формата JSON?*

В языке Python для десериализации данных из формата JSON используется встроенный модуль `json`. Этот модуль предоставляет функции для преобразования строк JSON в объекты Python и наоборот. Некоторые из основных функций модуля `json` это:

- `json.loads()`: преобразует строку JSON в объект Python.
- `json.load()`: преобразует файл с данными в формате JSON в объект Python.

- `json.dumps()`: преобразует объект Python в строку JSON.
- `json.dump()`: преобразует объект Python в формат JSON и записывает его в файл.

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

Для работы с данными формата JSON, содержащими кириллицу, необходимо учитывать кодировку. В Python, по умолчанию, при работе с JSON используется кодировка UTF-8, которая поддерживает символы кириллицы. Поэтому нет необходимости использовать дополнительные средства для работы с данными JSON, содержащими кириллицу.

*10. Самостоятельно ознакомьтесь со спецификацией JSON Schema? Что такое схема данных?*

JSON Schema - это спецификация для описания структуры и формата данных в формате JSON. Она позволяет определить ограничения и правила для данных, хранящихся в формате JSON. С помощью JSON Schema можно проверять и валидировать данные в формате JSON на соответствие заранее заданной структуре или формату. Схема данных описывает типы данных, значения по умолчанию, форматы и другие атрибуты, которые помогают определить правильность данных в формате JSON. Implementations of JSON Schema provide valuable tools for generating documentation, automating tests, data validation, and data generation based on JSON data structures.

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