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 «Основы кроссплатформенного программирования»

Выполнил:
Баканов Артем Вадимович
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.

Порядок выполнения работы:

Задание 1. 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.

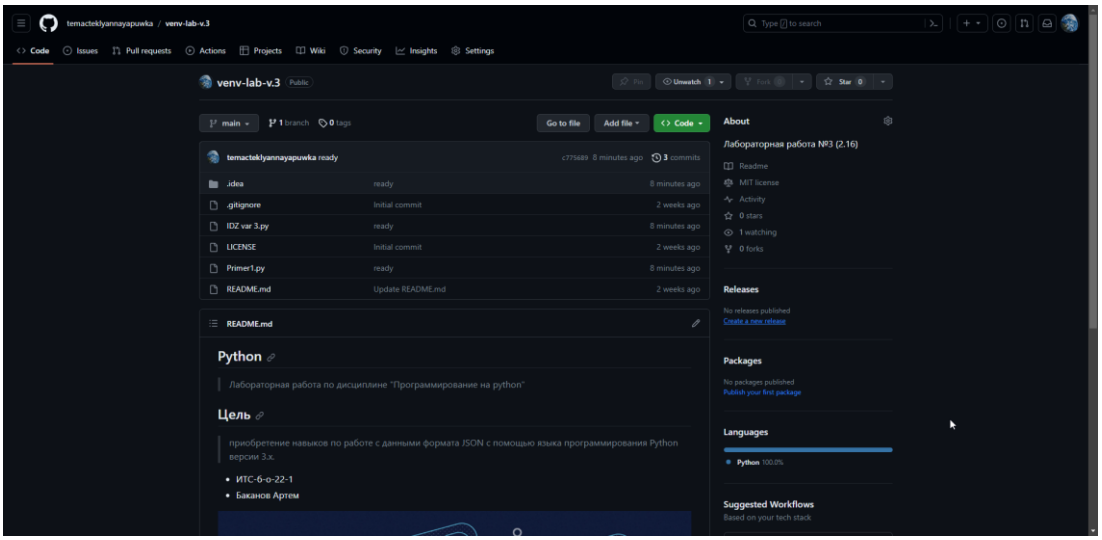

Рисунок 1. Создан новый репозиторий

Клонировал репозиторий на свой компьютер. В ходе данной лабораторной работы работала с моделью ветвления git-flow.

```
PS D:\study> git clone https://github.com/temacteklyannayapuwka/venv-lab-v.3.git
Cloning into 'venv-lab-v.3'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (1/1), done.
PS D:\study> cd venv-lab-v.3
PS D:\study\venv-lab-v.3> git init flow
Initialized empty Git repository in D:/study/venv-lab-v.3/flow/.git/
PS D:\study\venv-lab-v.3> 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? [D:/study/venv-lab-v.3/.git/hooks]
PS D:\study\venv-lab-v.3>
```

Рисунок 2. Клонирование и модель ветвления git-flow

Создал виртуальное окружение Anaconda

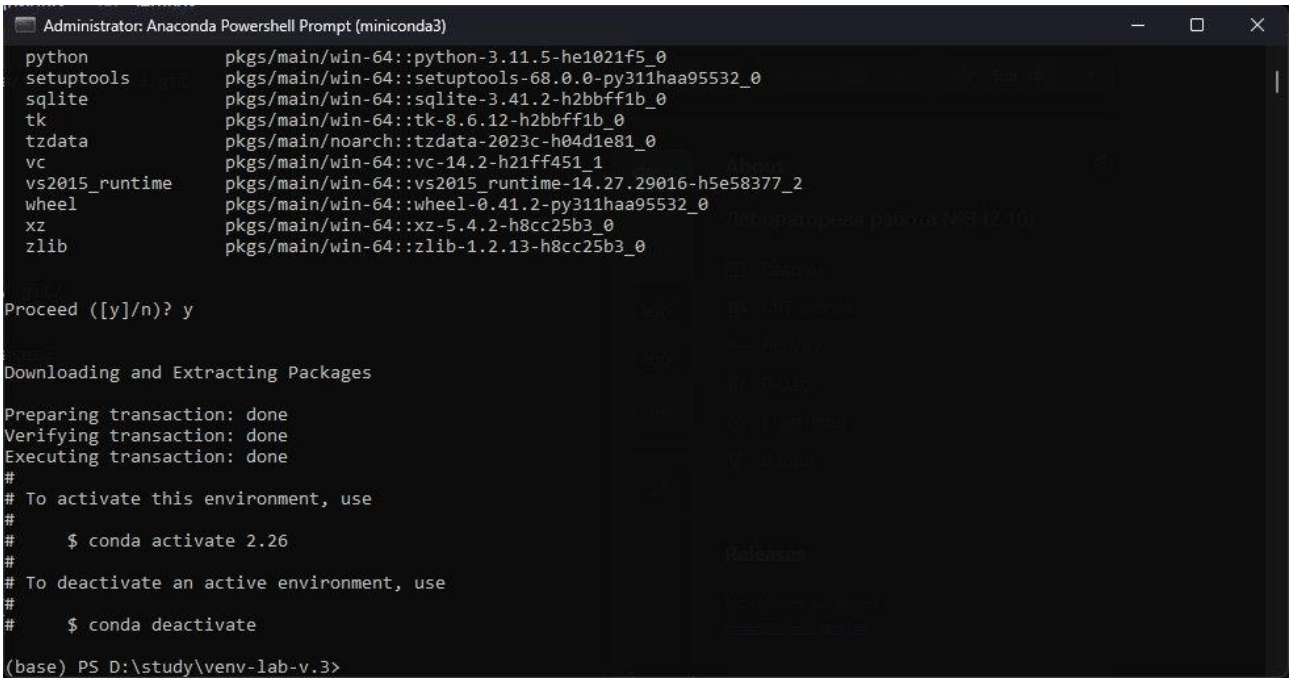

Рисунок 3. Создание виртуального окружения

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

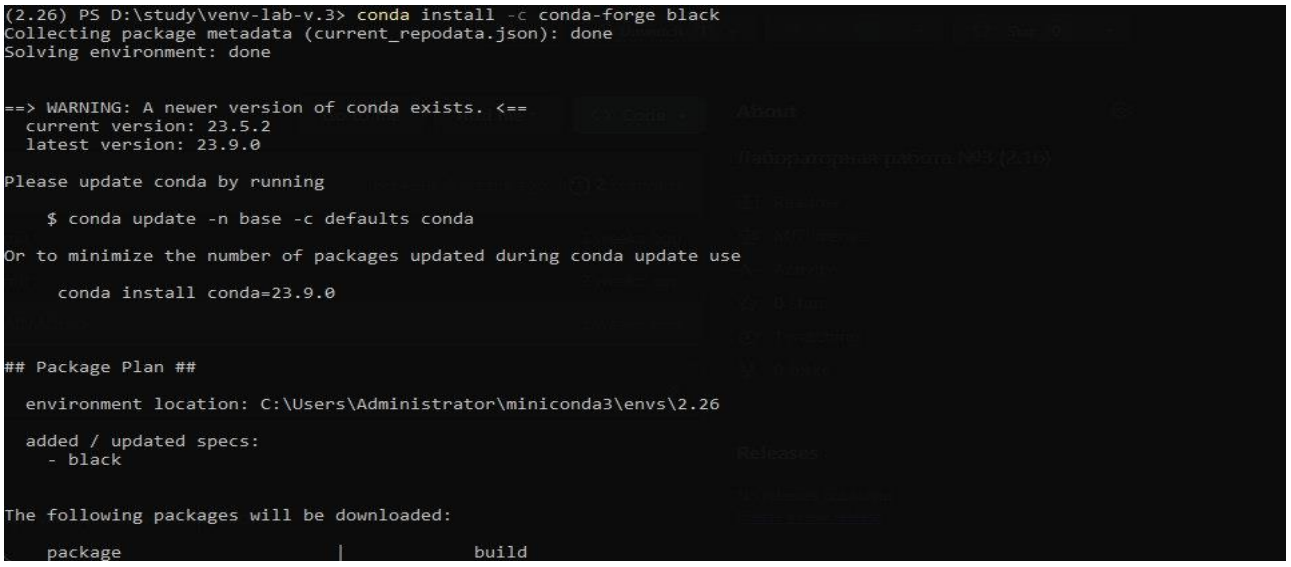

Рисунок 4. Установка пакета black

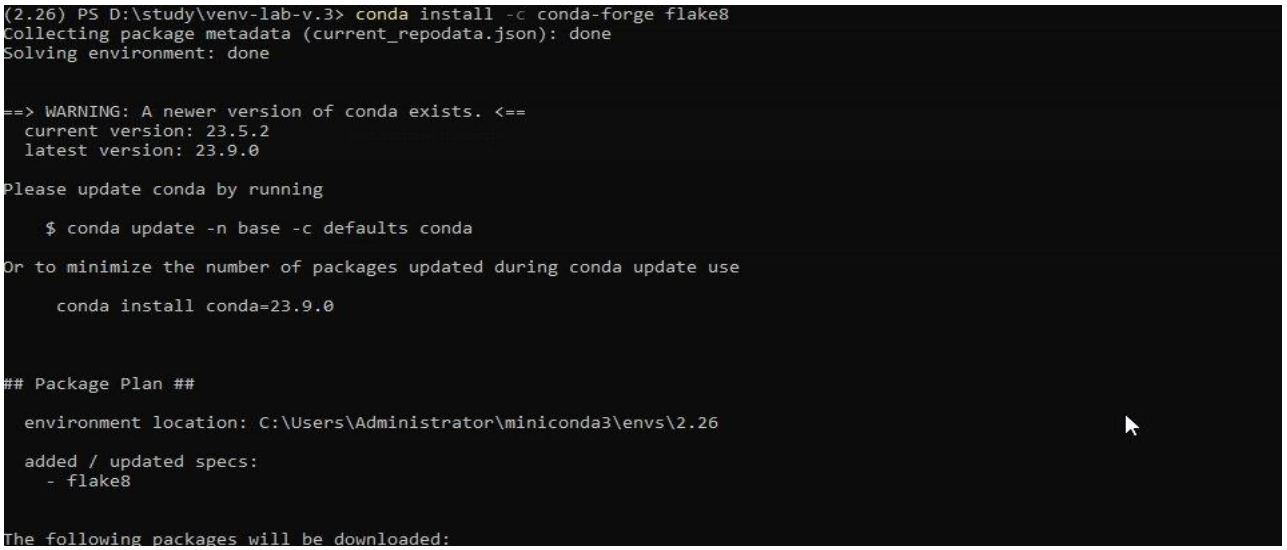

Рисунок 5. Установка пакета flake8

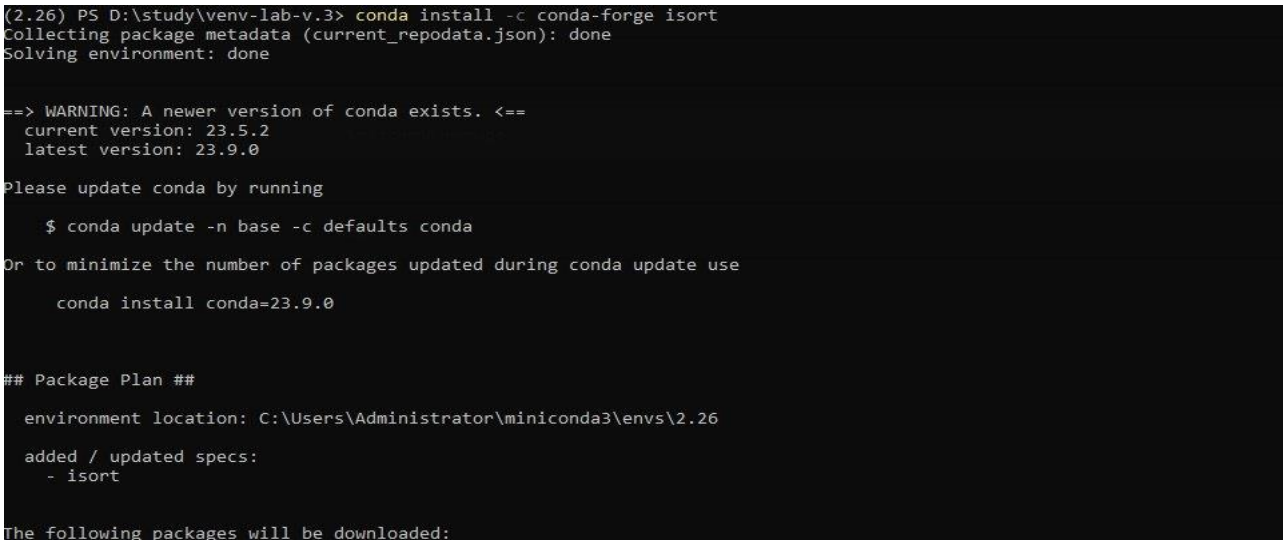

Рисунок 6. Установка пакета isort

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.

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

Работа с примером №1.

Условие примера: (что было в 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.

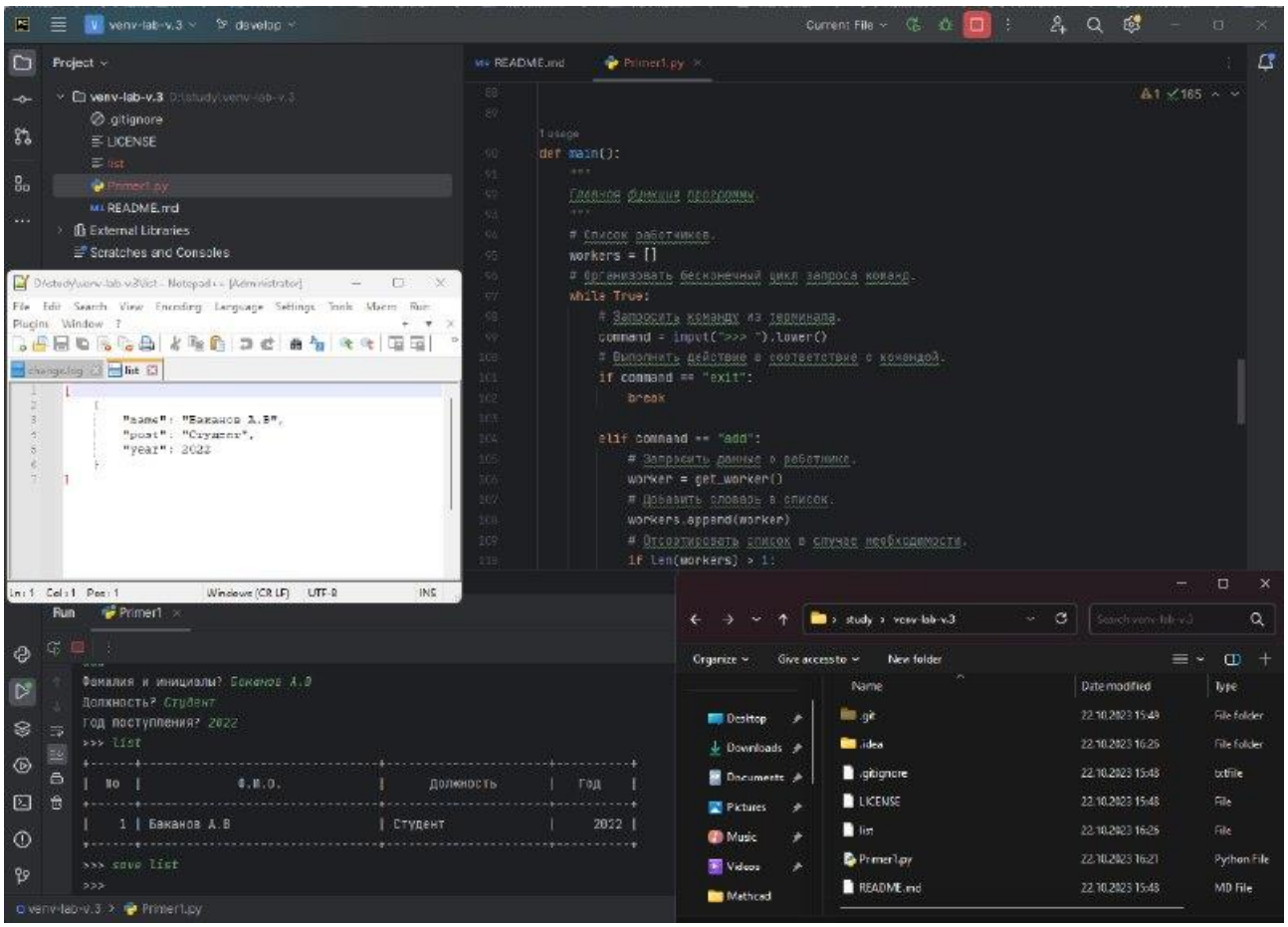

Рисунок 7. Работа с примером №1.

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.

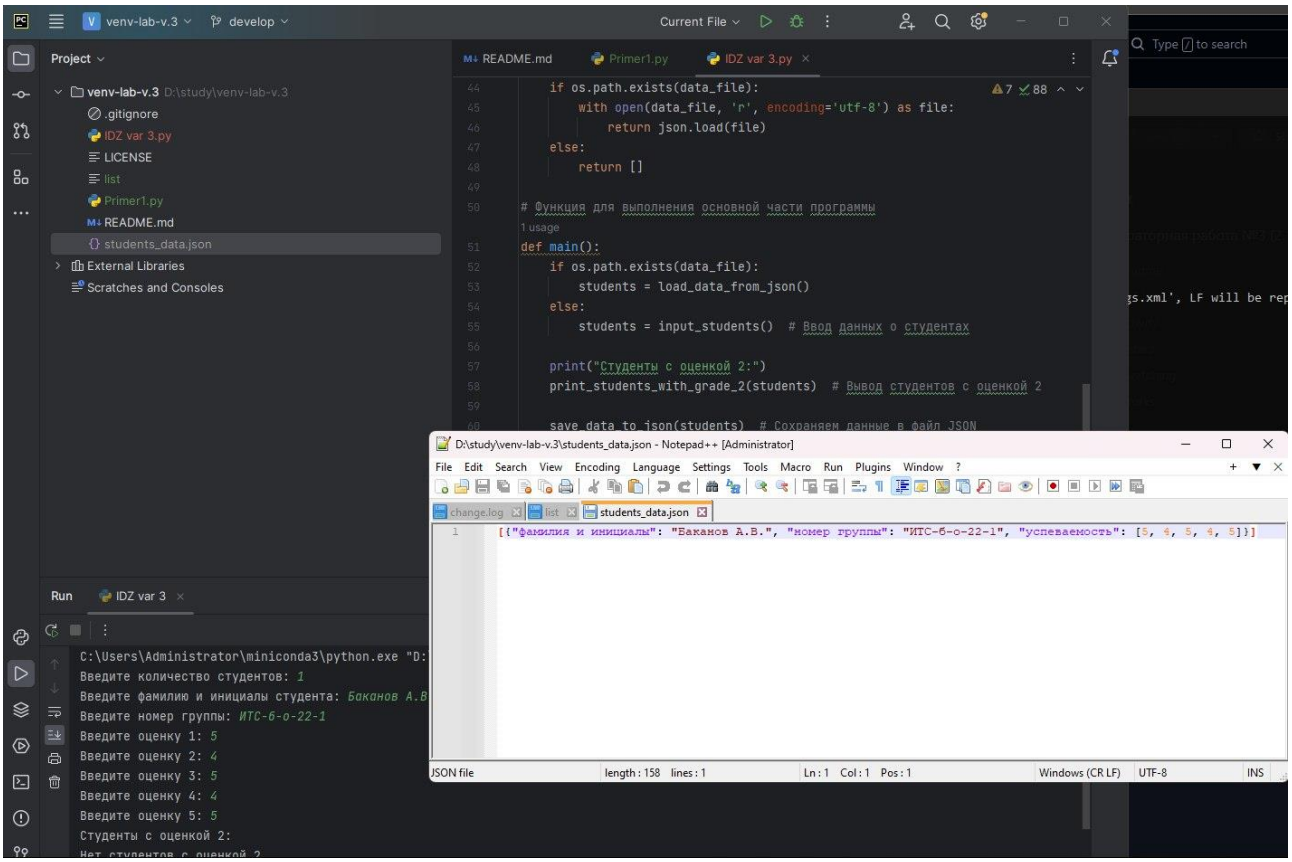

Рисунок 8. Выполнение индивидуального задания

Деактивировал виртуальное окружение.

```
(2.26) PS D:\study\venv-lab-v.3> conda deactivate
(base) PS D:\study\venv-lab-v.3>
```

Рисунок 9. Деактивация ВО

Слил ветку develop с веткой main и отправил на удаленный сервер

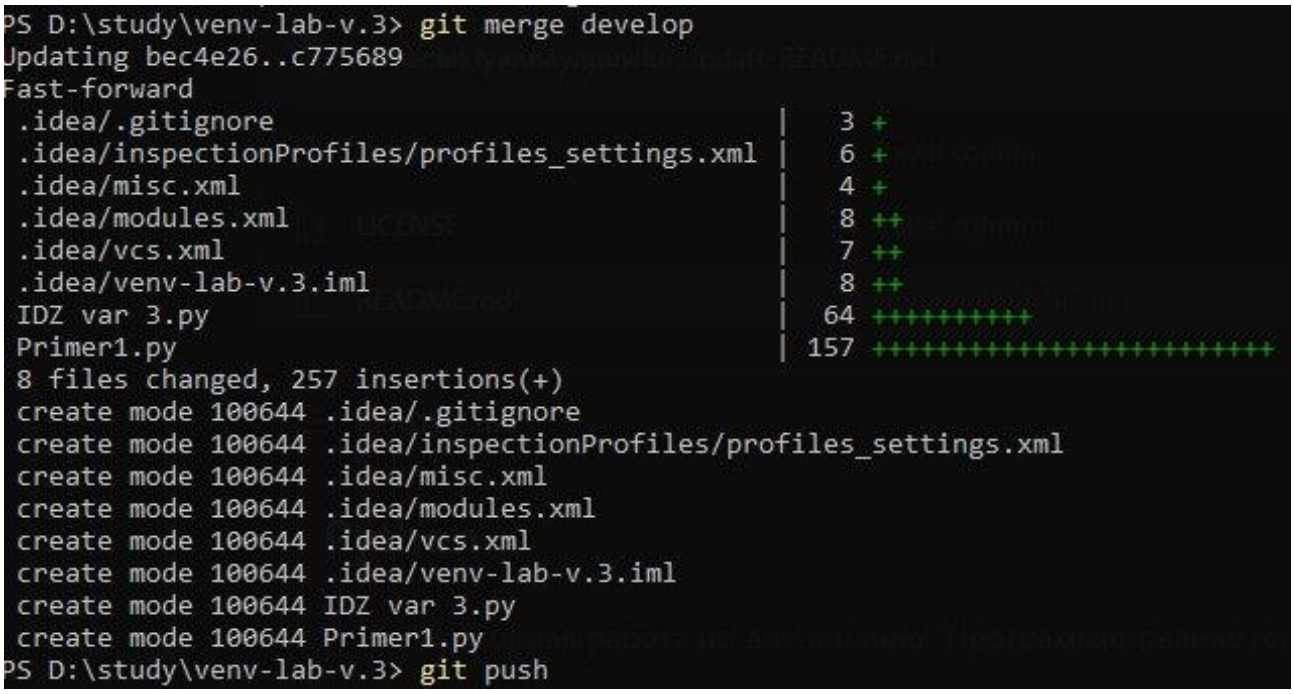

Рисунок 8. Слияние веток

Ссылка на репозиторий: https://github.com/temacteklyannayapuwka/venv-lab-v.3

Ответы на контрольные вопросы

1. Для чего используется JSON?

JSON (JavaScript Object Notation) используется для хранения и обмена данных 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 кирилицу?

Для работы с данными формата JSON, содержащими кирилицу, 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 кирилицу.

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: приобрел навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.