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 №2.16**

**дисциплины**

**«Основы кроссплатформенного программирования»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Пинаева Диана Олеговна  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) | |
|  | | Проверил:  Воронкин Р. А., доцент кафедры инфокоммуникаций  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

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

Ставрополь, 2023 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:

Задание 1.Создала общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавила файл .gitignore с необходимыми правилами. Клонировала свой репозиторий на свой компьютер.

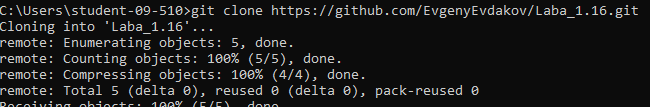


Рисунок 1. Клонирование репозитория

Задание 2.Организовала свой репозиторий в соответствие с моделью ветвления git-flow, появилась новая ветка develop в которой буду выполнять дальнейшие задачи.

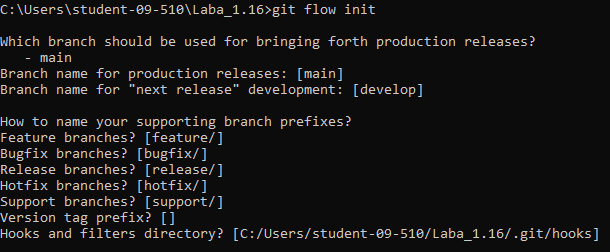


Рисунок 2. Модель ветвления git-flow

Задание 3.Создала виртуальное окружение conda и активировала его, также установила необходимые пакеты isort, black, flake8.

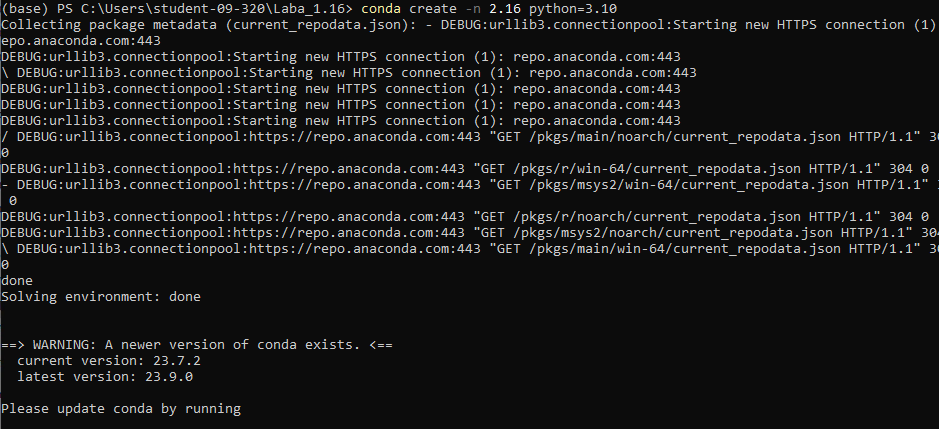
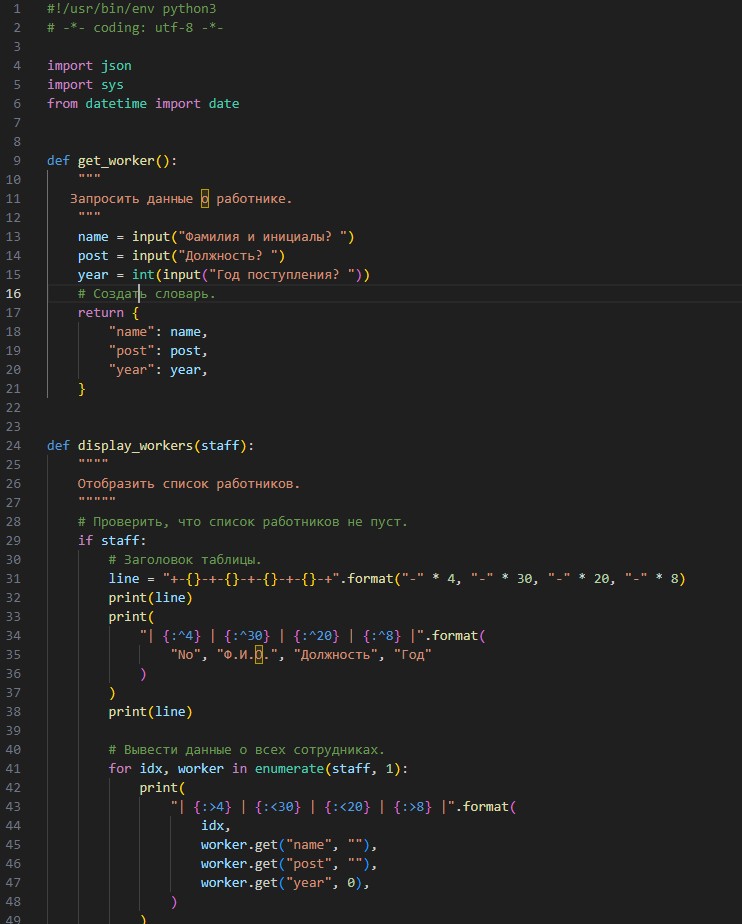
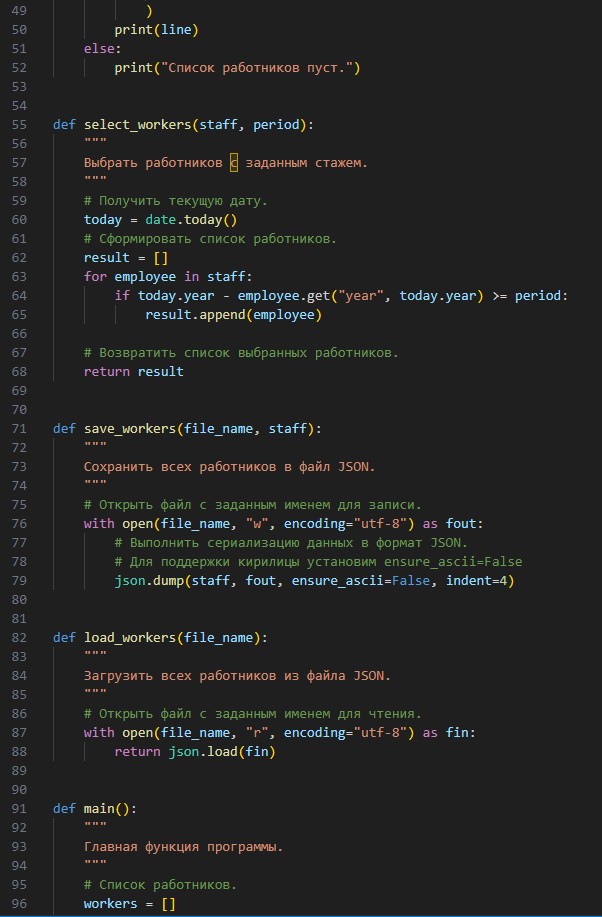


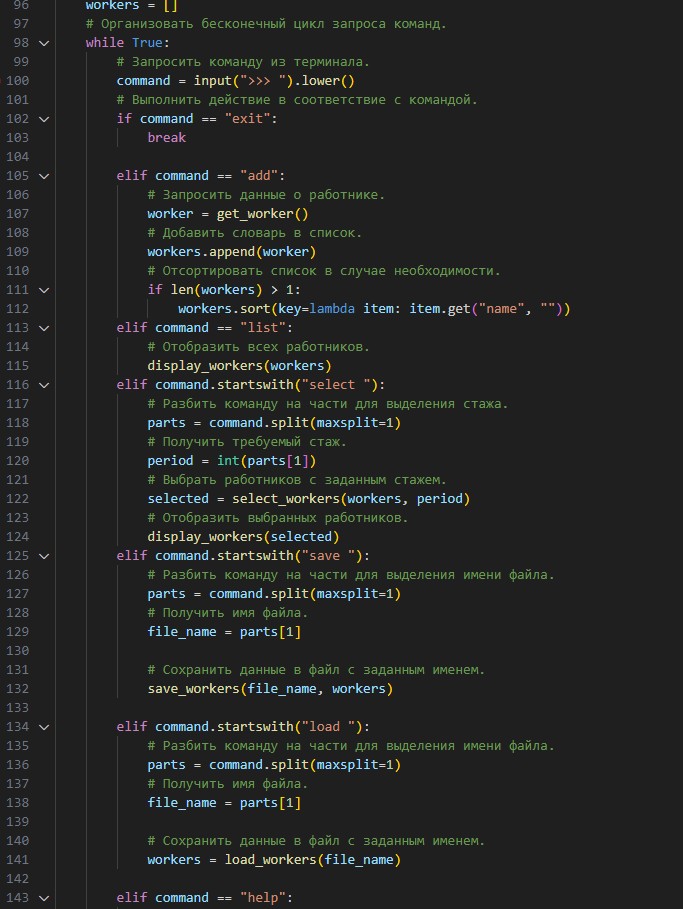
Рисунок 3. Создание виртуального окружения

Задание 4.Создала проект PyCharm в папке репозитория. Приступила к работе с примером. Добавила новый файл primer1.py.

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







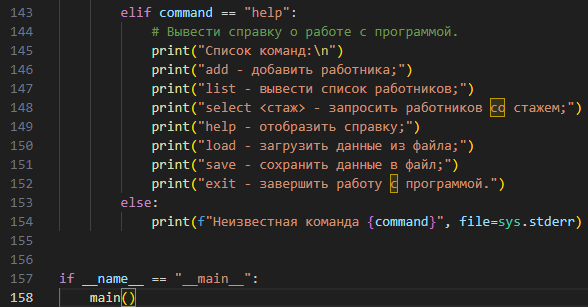


Рисунок 4-7. Пример 1

# Индивидуальное задание

# Вариант 10

Создала новый файл под названием idz.py.

Условие задания: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.

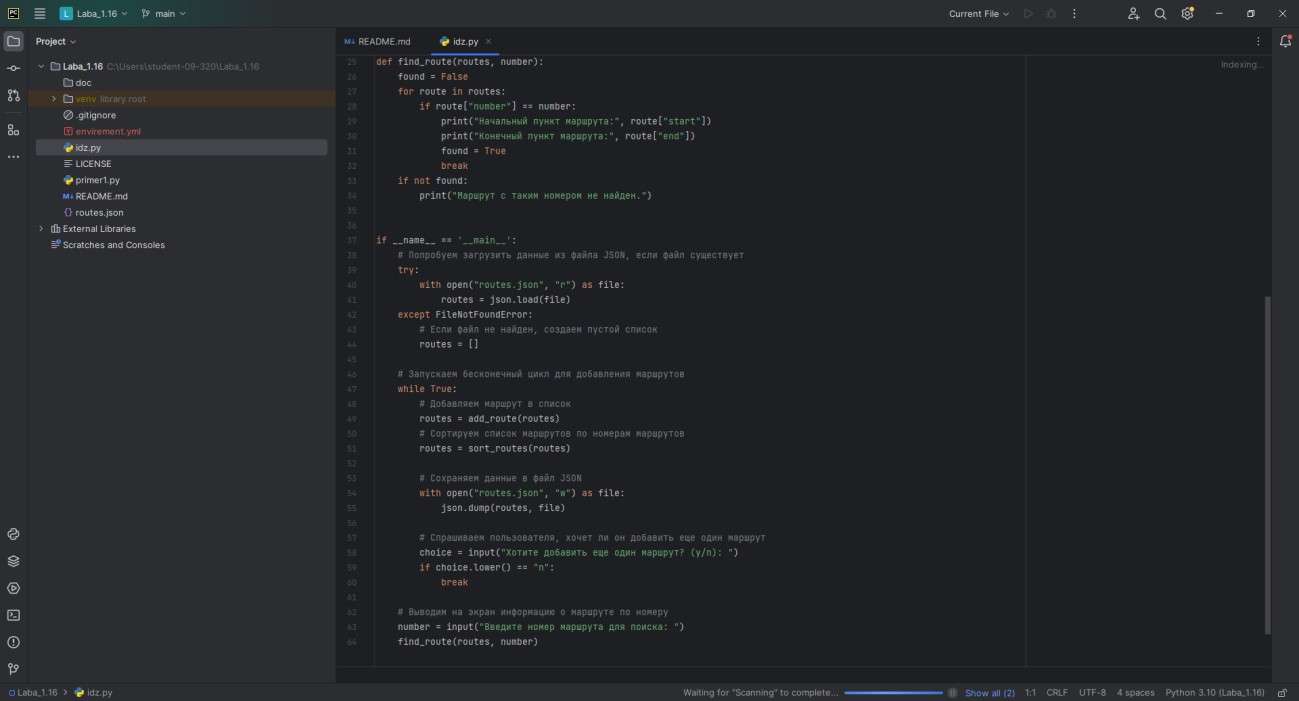


Рисунок 8. Программа индивидуального задания



Рисунок 9. Файл json

# Задание 5.

После выполнения работы на ветке develop, слила ее с веткой main и отправила изменения на удаленный сервер. Создала файл envirement.yml и деактивировала виртуальное окружение.



Ссылка: <https://github.com/Diana-372/Lab2.16.git>

# Ответы на контрольные вопросы:

1. Для чего используется JSON?

JSON — это стандарт обмена данными. Он позволяет легко сериализовать и десериализовать объекты. Стандарт часто применяют, когда разрабатывают API и веб-приложения.

# Какие типы значений используются в JSON?

В качестве значений в JSON могут быть использованы:

* запись — это неупорядоченное множество пар ключ:значение, заключённое в фигурные скобки «{ }». Ключ описывается строкой, между ним и значением стоит символ «:». Пары ключ-значение отделяются друг от друга запятыми.
* массив (одномерный) — это упорядоченное множество значений. Массив заключается в квадратные скобки «[ ]». Значения разделяются запятыми. Массив может быть пустым, то есть не содержать ни одного значения. Значения в пределах одного массива могут иметь разный тип.
* число (целое или вещественное).
* литералы true (логическое значение «истина»), false (логическое значение «ложь») и null.
* строка — это упорядоченное множество из нуля или более символов юникода, заключённое в двойные кавычки.

# Как организована работа со сложными данными в JSON?

* JSON позволяет организовать сложные структуры данных, такие как списки и вложенные словари (объекты).
* В JSON можно хранить разные типы данных, включая числа, строки, логические значения, массивы и объекты.
* Для организации сложных данных в JSON используются вложенные объекты и списки, позволяя создавать структуры данных любой сложности.

# Самостоятельно ознакомьтесь с форматом данных JSON5? В чем отличие этого формата от формата данных JSON?

* JSON5 - это расширение формата данных JSON, разработанное для улучшения читаемости и удобства записи JSON-данных.
* Отличие JSON5 от обычного JSON включает в себя дополнительные возможности, такие как использование комментариев, разделителей ключей и значений, а также возможность использования одиночных кавычек вместо двойных.
* JSON5 является более гибким и читаемым форматом для записи данных, но не является стандартом и не поддерживается всеми JSON- парсерами.

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

Для работы с данными в формате JSON5 на Python, вы можете использовать парсеры, поддерживающие JSON5, такие как demjson. Однако, JSON5 не является стандартом, поэтому поддержка может быть ограничена.

# Какие средства предоставляет язык Python для сериализации данных в формате JSON?

Для сериализации данных в формат JSON в Python можно использовать модуль json. Он предоставляет функции json.dump() и json.dumps(), а также класс json.JSONEncoder, который может быть настроен для сериализации данных в формат JSON.

# В чем отличие функций json.dump() и json.dumps()?

json.dump() записывает данные в файл. Вы используете его, когда хотите сохранить данные в файле.

json.dumps() превращает данные в строку. Вы используете его, когда хотите получить данные в виде строки для дальнейшей обработки, но не сохранять их в файле.

# Какие средства предоставляет язык Python для десериализации данных из формата JSON?

Для десериализации данных из формата JSON в Python используется модуль json, предоставляющий функции json.load() и json.loads().

# Какие средства необходимо использовать для работы с данными формата JSON, содержащими кирилицу?

Для работы с данными JSON, содержащими кириллицу, важно убедиться, что данные правильно кодируются и декодируются. Обычно это не вызывает проблем, поскольку JSON поддерживает Unicode, включая кириллические символы. Однако, при чтении и записи JSON-файлов, убедитесь, что правильно установлена кодировка (например, utf-8) для текстовых данных.

# Самостоятельно ознакомьтесь со спецификацией JSON Schema? Что такое схема данных?

JSON Schema - это спецификация, которая описывает формат данных JSON и правила их валидации. С помощью JSON Schema можно определить структуру, типы данных и ограничения для JSON-данных. JSON Schema используется для проверки соответствия данных определенным правилам. Это полезно, например, при валидации данных, получаемых из внешних источников. JSON Schema не является частью стандартной библиотеки Python, но существуют библиотеки и инструменты, поддерживающие JSON Schema, которые могут использоваться в Python.

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