

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:

Арзютов Иван Владиславович
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., 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.

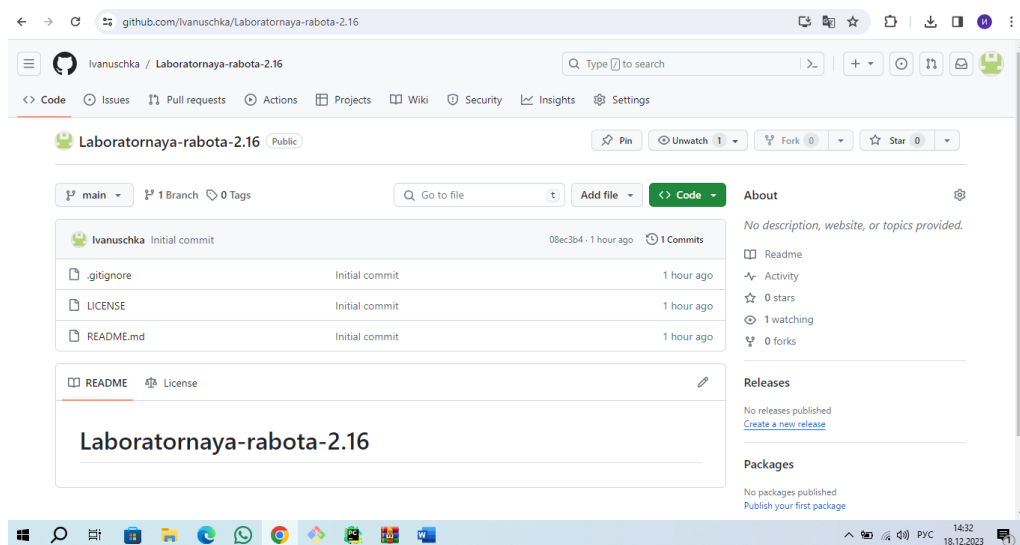


Рисунок 1. Создан новый репозиторий

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

```
C:\Program Files\Git>git-cmd.exe
C:\Program Files\Git>cd C:\Users\ivana\OneDrive\Рабочий стол\Иван
C:\Users\ivana\OneDrive\Рабочий стол\Иван>git clone https://github.com/Ivanuschka/Laboratornaya-rabota-2.16.git
Cloning into 'Laboratornaya-rabota-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\ivana\OneDrive\Рабочий стол\Иван>cd C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornaya-rabota-2.16
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornaya-rabota-2.16>git checkout -b test_1
Switched to a new branch 'test_1'
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornaya-rabota-2.16>git add .
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornaya-rabota-2.16>git commit -m 'primer'
[test_1 3a291f6] 'primer'
1 file changed, 128 insertions(+)
create mode 100644 primer.py
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornaya-rabota-2.16>
```

Рисунок 2. Клонирование и модель ветвления git-flow

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

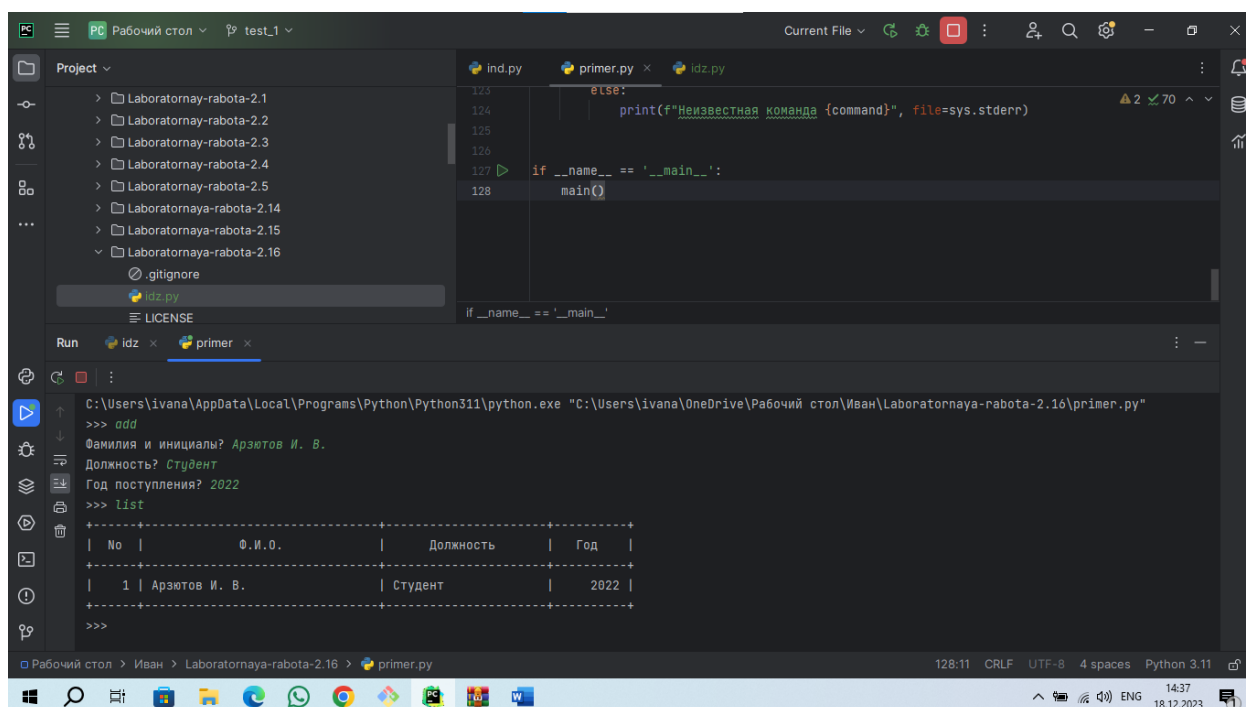


Рисунок 7. Работа с примером №1.

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 возрастанию номера группы; вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0; если таких студентов нет, вывести соответствующее сообщение.

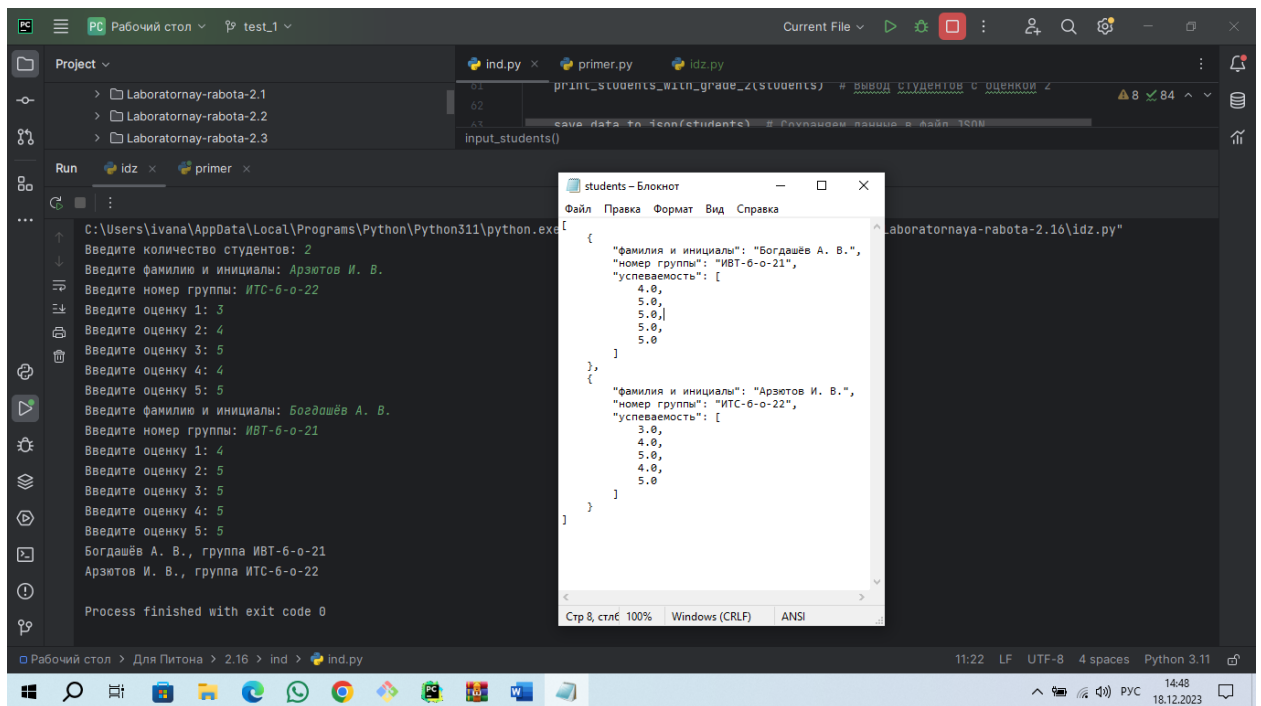


Рисунок 8. Выполнение индивидуального задания

Слил ветку test_1 с веткой main и отправил на удаленный сервер

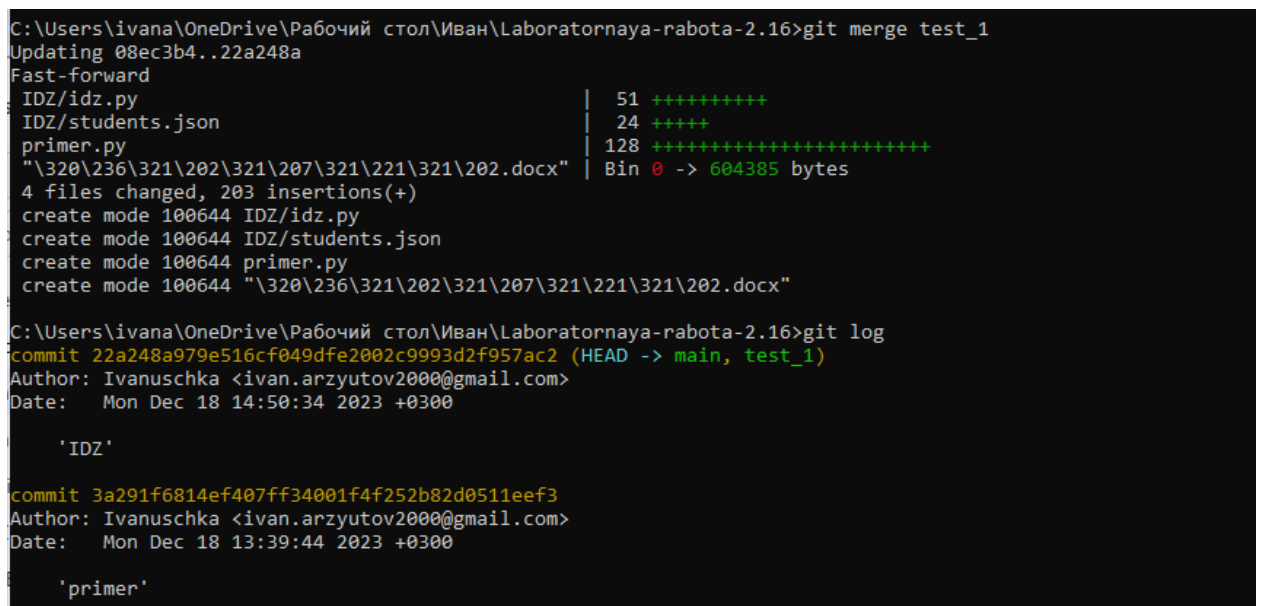


Рисунок 9. Слияние веток

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