Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития Кафедра инфокоммуникаций

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.16**

**дисциплины «Программирование на языке Python»**

|  |
| --- |
| Выполнила:  Мурашко Анастасия Юрьевна  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., 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:**

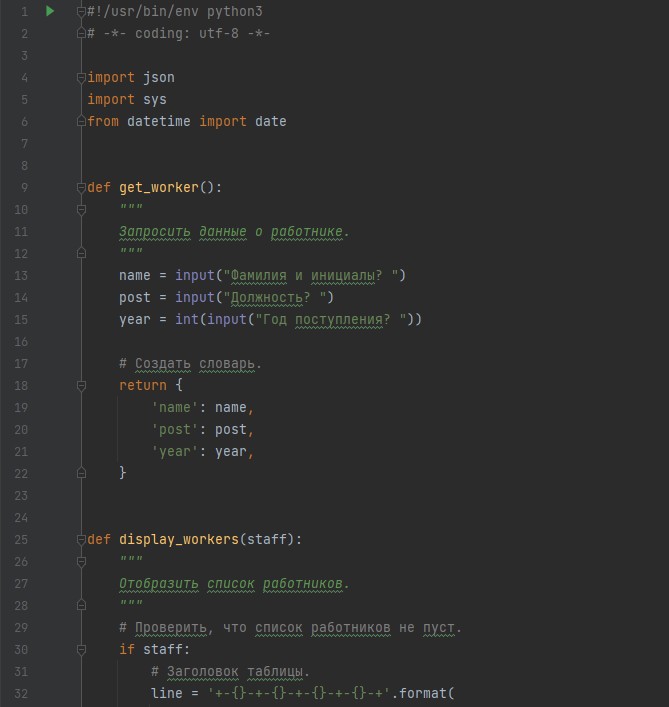
**Пример 1**

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

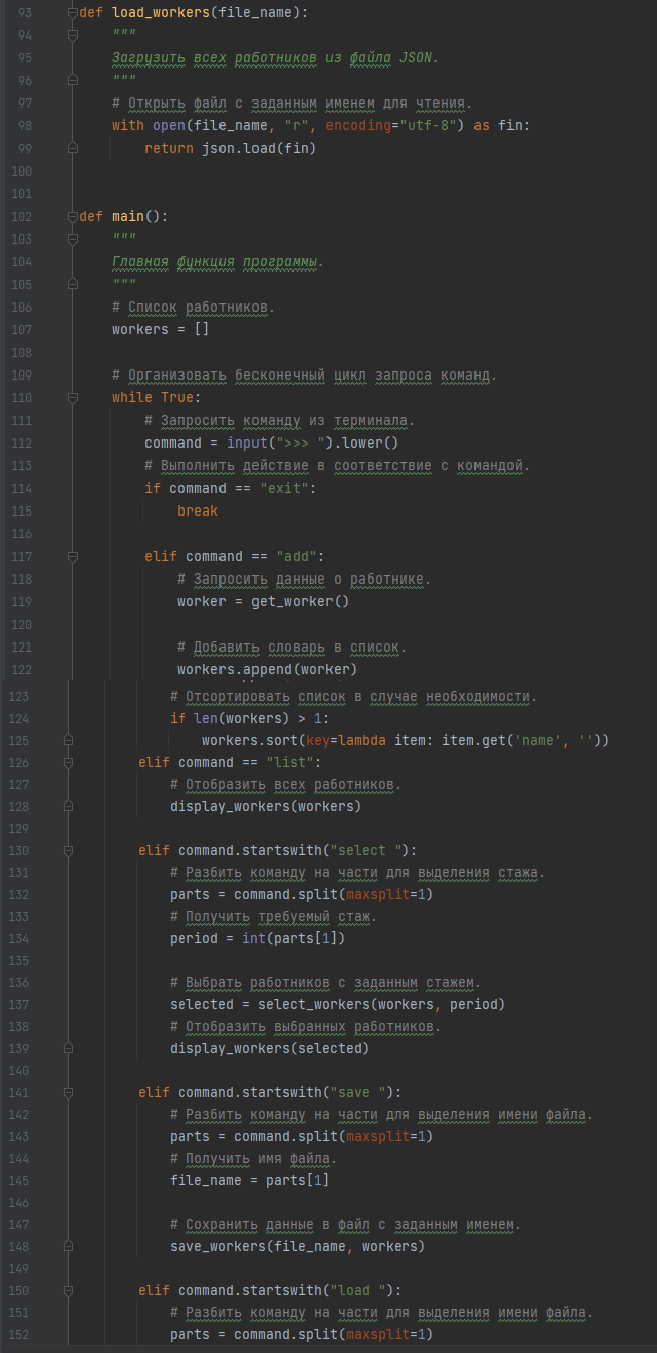
Решение: введем следующие команды для работы с файлом формата JSON в интерактивном режиме:

1. load - загрузить данные из файла, имя файла должно отделяться от команды load пробелом. Например: load data.json.
2. save - сохранить сделанные изменения в файл, имя файла должно отделяться от команды save пробелом. Например: save data.json.

Напишем программу для решения поставленной задачи.







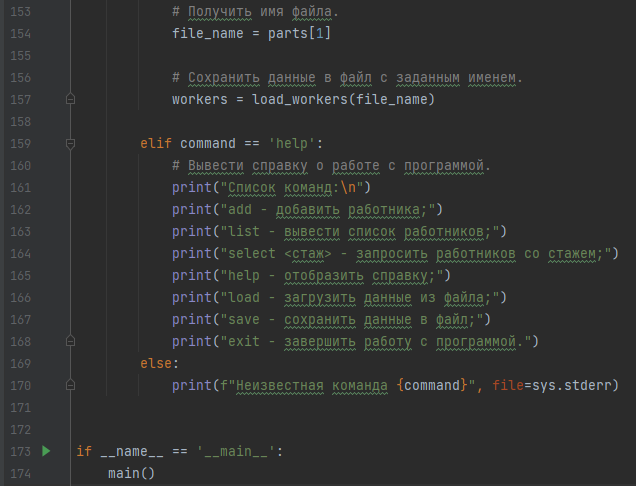


Рисунок 1. Код

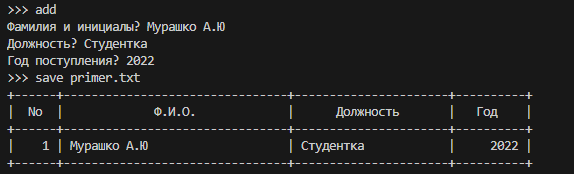
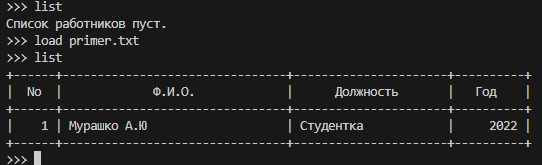
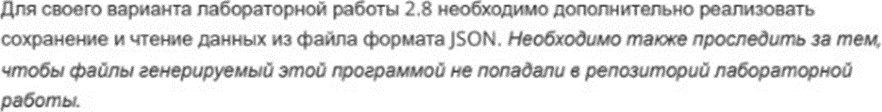


Рисунок 2. Результат выполнения

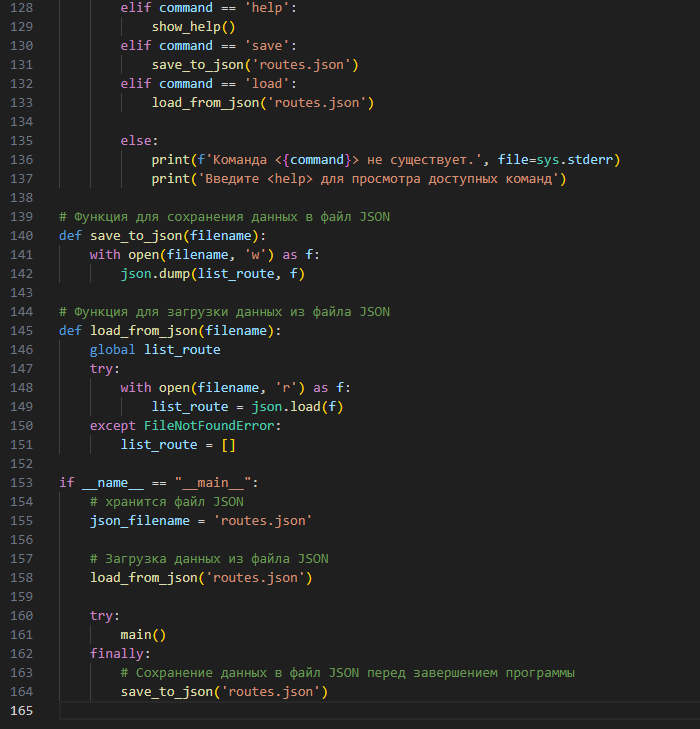
**Индивидуальное задание 1**









Рисунок 3. Код

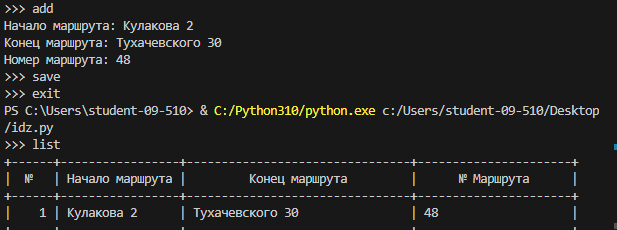


Рисунок 4. Результат выполнения индивидуального задания

**Ответы на контрольные вопросы:**

1. Для чего используется JSON?

JSON представляет собой хорошую альтернативу XML и требует куда меньше форматирования контента. Это информативное руководство поможет вам быстрее разобраться с данными, которые вы можете использовать с JSON и основной структурой с синтаксисом этого же формата.

1. Какие типы значений используются в JSON? Запись, массив, число, литералы, строка.
2. 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.

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

JSON5 - предложенное расширение формата json в соответствии с синтаксисом ECMAScript 5, вызванное тем, что json используется не только для общения между программами, но и создаётся/редактируется вручную. Файл JSON5 всегда является корректным кодом ECMAScript 5. 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.

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

JSON5 расширяет формат обмена данными JSON, чтобы сделать его немного более удобным в качестве языка конфигурации:

1. Комментарии в стиле JavaScript (как однострочные, так и многострочные) являются законными.
2. Ключи объектов могут быть без кавычек, если они являются законными идентификаторами ECMAScript.
3. Объекты и массивы могут заканчиваться запятыми.
4. Строки могут заканчиваться в одинарные кавычки, и допускаются многострочные строковые литералы.
5. Какие средства предоставляет язык Python для сериализации данных в формате JSON?

Модуль json предоставляет удобный метод dump() для записи данных в файл. Существует также метод dumps() для записи данных в обычную строку. Типы данных Python кодируются в формат JSON в соответствии с интуитивно понятными правилами преобразования.

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

Функции json.dump() и json.dumps() - это две разные функции из модуля json в Python, и они выполняют разные задачи:

1) json.dump(): Эта функция используется для записи данных в формате JSON в файл. Она принимает два аргумента: данные, которые вы хотите записать, и файловый объект (или объект, поддерживающий запись, такой как io.TextIOBase). Эта функция записывает данные в указанный файл в формате JSON.

2) json.dumps(): Эта функция используется для преобразования данных в формате JSON в строку (текстовое представление). Она принимает один аргумент - данные, которые вы хотите преобразовать. Эта функция возвращает строку, содержащую данные в формате JSON.

Итак, основное отличие между ними заключается в том, что json.dump() записывает данные в файл, а json.dumps() возвращает JSON-строку. Выбор между ними зависит от того, какие действия вы хотите выполнить: сохранить данные в файл или получить JSON-строку для дальнейшей обработки.

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

В модуле json определены методы load() и loads(), предназначенные для преобразования кодированных в формате JSON данных в объекты Python.

Подобно операции сериализации, также существует таблица преобразования типов, определяющая правила для обратного декодирования данных.

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

Параметр ensure\_ascii.

1. 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:** в ходе лабораторной работы приобретены навыки по работе с данными формата JSON с помощью языка программирования Python версии

3.x.