

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
дисциплины «Анализ данных»

Выполнил:
Гайчук Дарья Дмитриевна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика
и вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А.-доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 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. Изучила теоретический материал работы, создала общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python, также добавила файл .gitignore с необходимыми правилами.

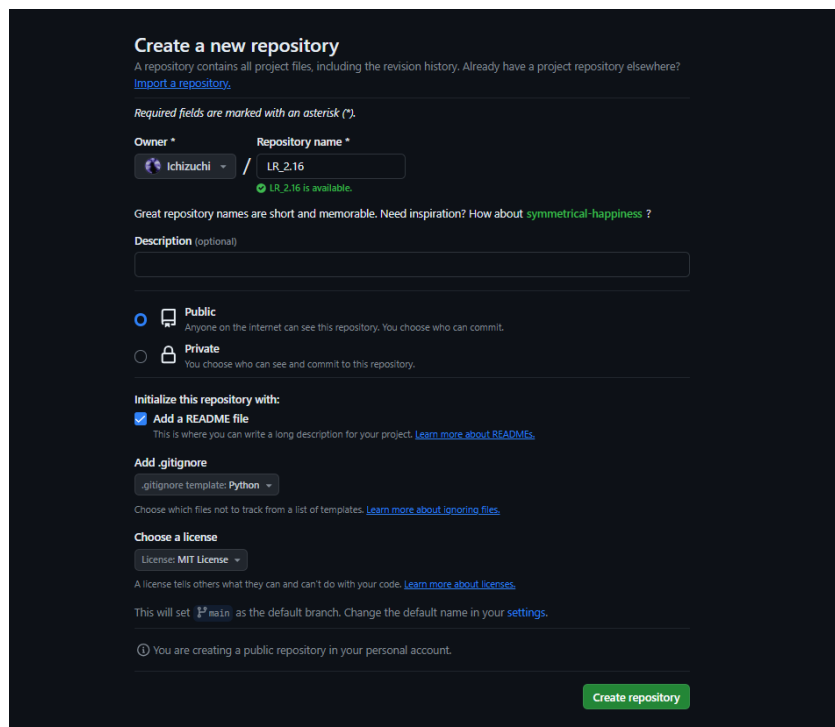


Рисунок 1. Создан новый репозиторий

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

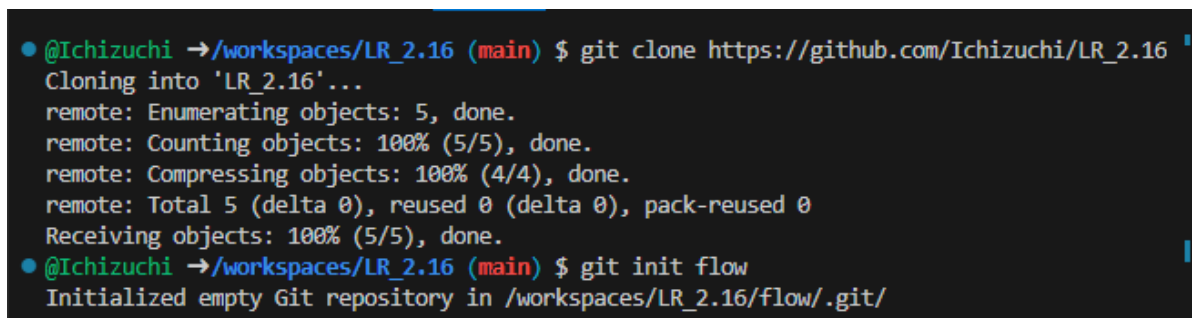


Рисунок 2. Клонирование и модель ветвления git-flow

Создала виртуальное окружение Anaconda

```
● @Ichizuchi → /workspaces/LR 2.16 (main) $ conda create -n myenv python=3.10
Retrieving notices: ...working... done
Channels:
- defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /opt/conda/envs/myenv

added / updated specs:
- python=3.10

The following packages will be downloaded:

package | build
-----|-----
_libgcc_mutex-0.1 | main 3 KB
_openmp_mutex-5.1 | 1_gnu 21 KB
bzip2-1.0.8 | h7b6447c_0 78 KB
ca-certificates-2023.12.12 | h06a4308_0 126 KB
ld_impl_linux-64-2.38 | h1181459_1 654 KB
libffi-3.4.4 | h6a678d5_0 142 KB
libgcc-ng-11.2.0 | h1234567_1 5.3 MB
libgomp-11.2.0 | h1234567_1 474 KB
libstdcxx-ng-11.2.0 | h1234567_1 4.7 MB
libuuid-1.41.5 | h5eee18b_0 27 KB
ncurses-6.4 | h6a678d5_0 914 KB
pip-23.3.1 | py310h06a4308_0 2.7 MB
python-3.10.13 | h955ad1f_0 26.8 MB
readline-8.2 | h5eee18b_0 357 KB
setuptools-68.2.2 | py310h06a4308_0 957 KB
sqlite-3.41.2 | h5eee18b_0 1.2 MB
tk-8.6.12 | h1ccaba5_0 3.0 MB
tzdata-2023d | h04d1e81_0 117 KB
wheel-0.41.2 | py310h06a4308_0 109 KB
xz-5.4.5 | h5eee18b_0 646 KB
zlib-1.2.13 | h5eee18b_0 103 KB
-----|-----
Total: 48.3 MB

The following NEW packages will be INSTALLED:

_libgcc_mutex pkgs/main/linux-64::_libgcc_mutex-0.1-main
_openmp_mutex pkgs/main/linux-64::_openmp_mutex-5.1-1_gnu
```

Рисунок 3. Создание виртуального окружения

Создала виртуальное окружение (ВО) Miniconda и активировала его, также установила необходимые пакеты isort, black, flake8.

```
● @Ichizuchi → /workspaces/LR_2.16 (main) $ conda install -c conda-forge black
Channels:
- conda-forge
- defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /opt/conda

added / updated specs:
- black
```

Рисунок 4. Установка пакета black

```
● @Ichizuchi →/workspaces/LR_2.16 (main) $ conda install -c conda-forge flake8
Channels:
  - conda-forge
  - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /opt/conda

added / updated specs:
  - flake8
```

Рисунок 5. Установка пакета flake8

```
● @Ichizuchi →/workspaces/LR_2.16 (main) $ conda install -c conda-forge isort
Channels:
  - conda-forge
  - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /opt/conda

added / updated specs:
  - isort
```

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.

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

```
list.json
1
2
3 {
4   "name": "Ufqxer",
5   "post": "студент",
6   "year": 2022
7 },
8 {
9   "name": "Гайчук Д.Д.",
10  "post": "Студент",
11  "year": 2022
12 }
```

```
primer_1.py
90 def main():
141
142     elif command == "help":
143         # Вывести справку о работе с программой.
144         print("Список команд:\n")
145         print("add - добавить работника;")
146         print("list - вывести список работников;")
147         print("select <стаж> - запросить работников со стажем;")
148         print("help - отобразить справку;")
149         print("load - загрузить данные из файла;")
150         print("save - сохранить данные в файл;")
151         print("exit - завершить работу с программой.")
152     else:
153         print(f"Неизвестная команда {command}", file=sys.stderr)
154
155
156 if __name__ == "__main__":
157     main()
```

```
Год поступления? 2022
>>> lust
Неизвестная команда lust
>>> list
+-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Ufqxer | студент | 2022 |
+-----+-----+-----+-----+
>>> save filw.json
>>> save list.json
>>> add
Фамилия и инициалы? Гайчук Д.Д.
Должность? Студент
Год поступления? 2022
>>> save list.json
>>> load list.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: название пункта назначения рейса; номер рейса; тип самолета. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по возрастанию номера рейса; вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры; если таких рейсов нет, выдать на дисплей соответствующее сообщение.

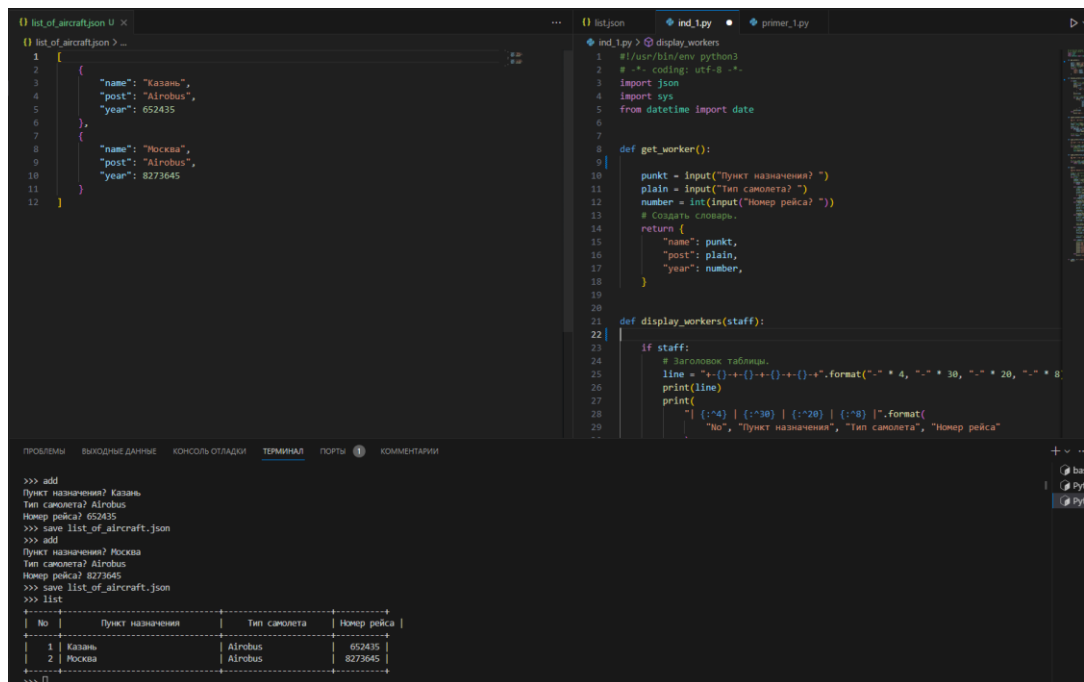


Рисунок 8. Результат индивидуального задания

Слила ветку develop 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 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 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: приобрела навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x