Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерных технологий

Лабораторная работа №4

Исследование протоколов,

форматов обмена информацией и языков разметки документов

Вариант 20

Выполнила:

Павличенко Софья Алексеевна, Р3115

Проверила:

Авксентьева Елена Юрьевна,

к.п.н., доцент факультета ПИиКТ

Санкт-Петербург 2023г.

Оглавление

[Обязательное задание 3](#_Toc152078344)

[Исходный файл 3](#_Toc152078345)

[Решение 4](#_Toc152078346)

[Исходный код 4](#_Toc152078347)

[Результат 5](#_Toc152078348)

[Дополнительное задание №1 6](#_Toc152078349)

[Решение 6](#_Toc152078350)

[Исходный код 6](#_Toc152078351)

[Результат 7](#_Toc152078352)

[Сравнение результатов 7](#_Toc152078353)

[Дополнительное задание №2 8](#_Toc152078354)

[Решение 8](#_Toc152078355)

[Исходный код 8](#_Toc152078356)

[Результат 9](#_Toc152078357)

[Сравнение результатов 9](#_Toc152078358)

[Дополнительное задание №4 10](#_Toc152078359)

[Результат и вывод 10](#_Toc152078360)

[Дополнительное задание №5 11](#_Toc152078361)

[Решение 11](#_Toc152078362)

[Исходный код 11](#_Toc152078363)

[Результат 11](#_Toc152078364)

[Вывод 12](#_Toc152078365)

[Заключение 13](#_Toc152078366)

[Список источников 14](#_Toc152078367)

# Обязательное задание

Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый путём простой замены метасимволов исходного формата (JSON) на метасимволы результирующего формата (YAML).

## Исходный файл

Исходный файл JSON представлен на Рис. 1.

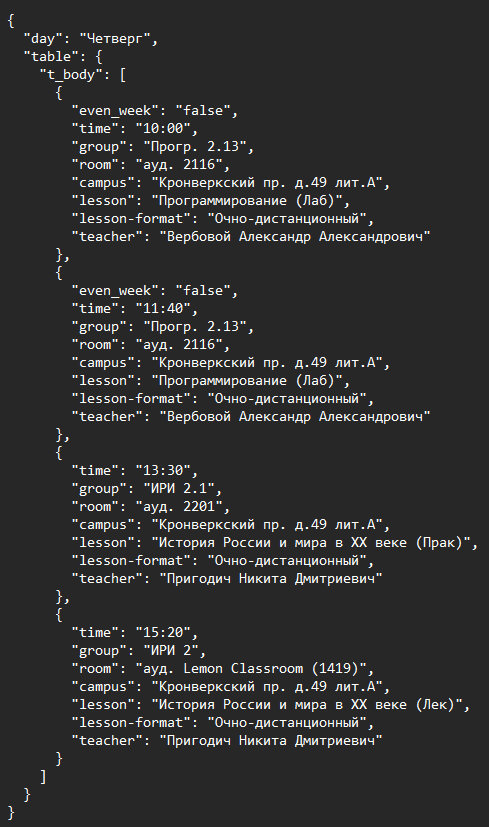


Рис.1

## Решение

### Исходный код

Исходный код обязательного задания представлен на Рис. 2.

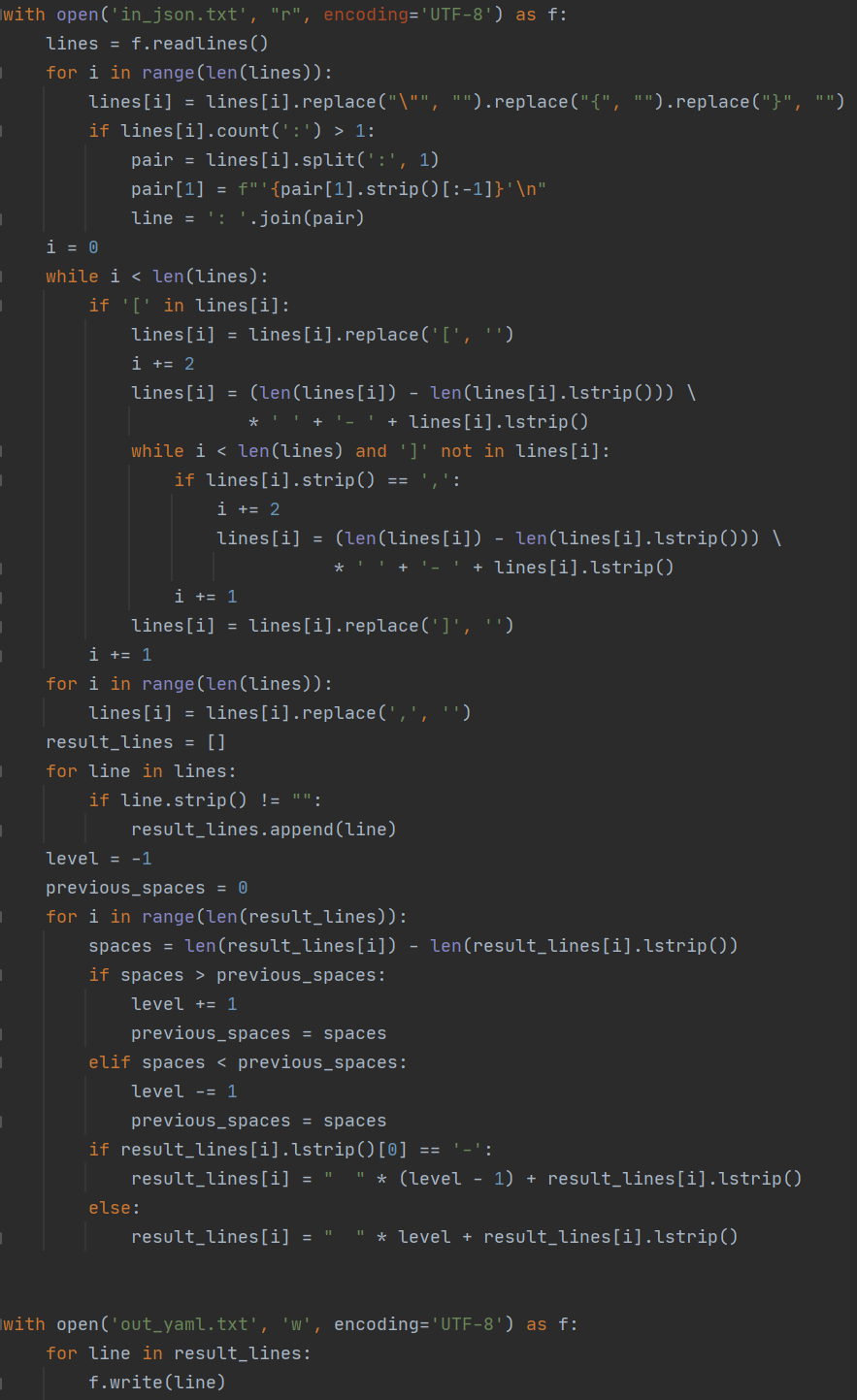


Рис. 2

### Результат

Результат выполнения кода обязательного задания представлен на Рис. 3.

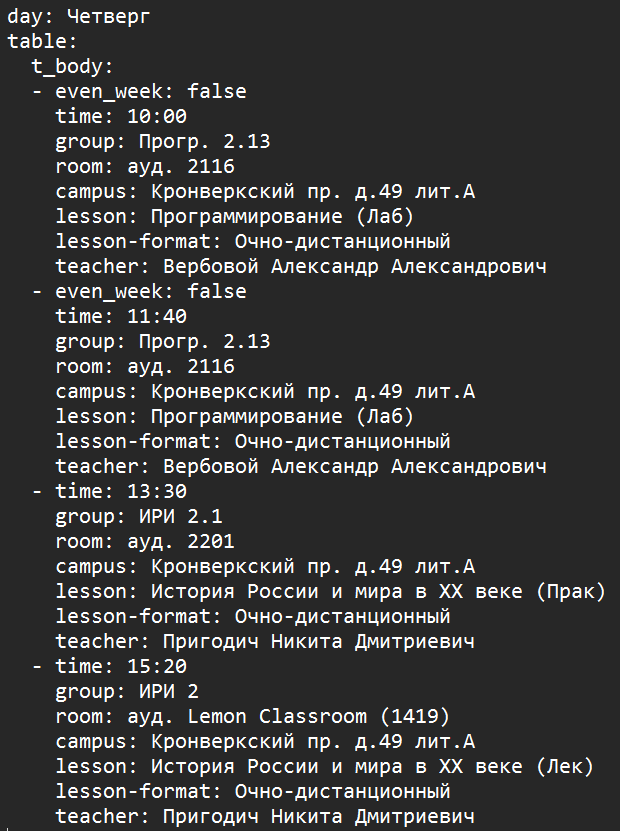


Рис. 3

# Дополнительное задание №1

1. Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
2. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.
3. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

## Решение

### Исходный код

Решение дополнительного задания №1 представлено на Рис. 4.

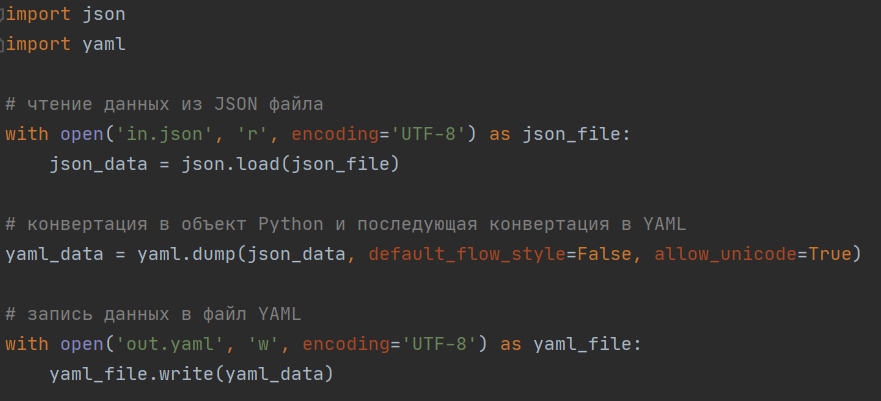


Рис. 4

### Результат

Результат выполнения кода дополнительного задания №1 представлен на Рис. 5.

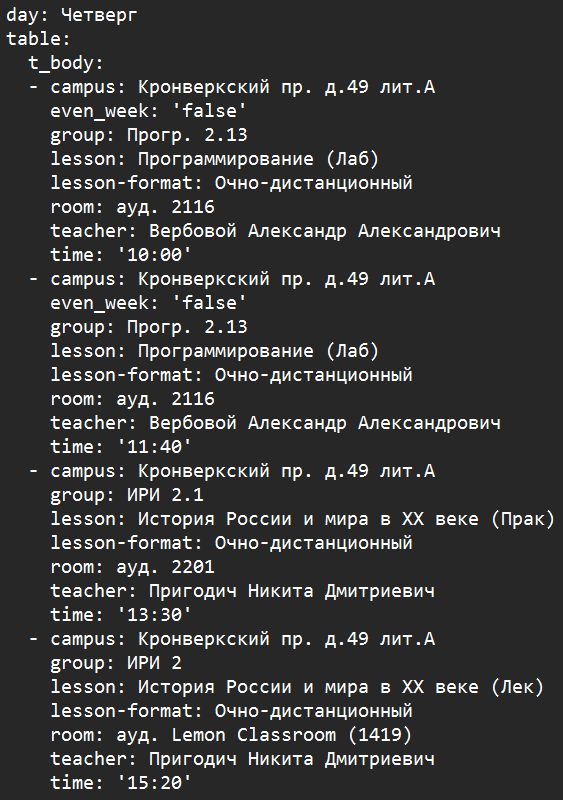


Рис. 5

## Сравнение результатов

При использовании специальных библиотек json и pyyaml в Python элементы массива в YAML файле строятся в алфавитном порядке, а не в исходном. Это единственное отличие от результата выполнения обязательного задания, но оно не является важным, так как формат YAML не требует обязательной сортировки ключей в алфавитном порядке. Можно сделать вывод, что оба варианта парсинга дают верные результаты.

# Дополнительное задание №2

1. Переписать исходный код, добавив в него использование регулярных выражений.
2. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

## Решение

### Исходный код

Решение дополнительного задания №2 представлено на Рис. 6.



Рис. 6

### Результат

Результат выполнения кода дополнительного задания №2 представлен на Рис. 7.

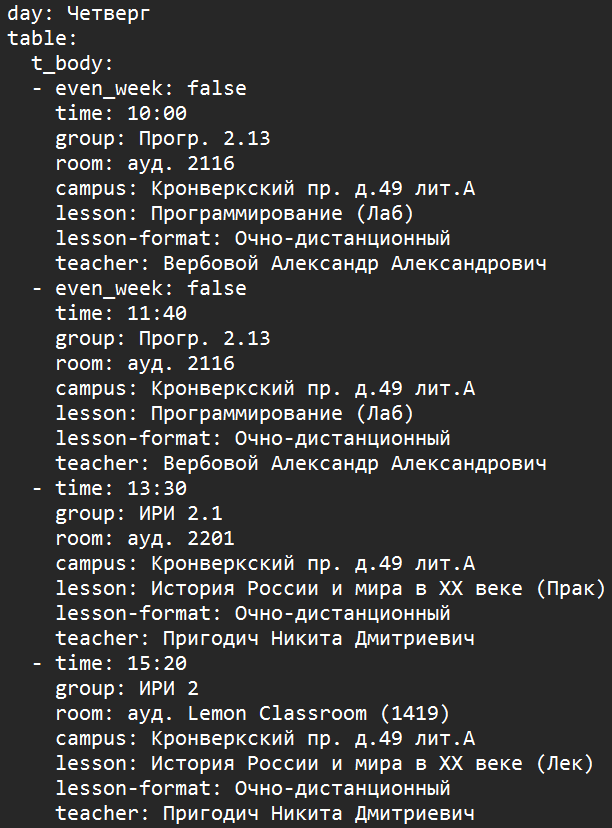


Рис. 7

## Сравнение результатов

Парсинг с использованием регулярных выражений и без дали идентичные результаты.

# Дополнительное задание №4

1. Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле.
2. Проанализировать полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

## Результат и вывод

Используя библиотеку time в Python я измерила стократное время выполнения заданий. Результаты представлены на Рис. 8.

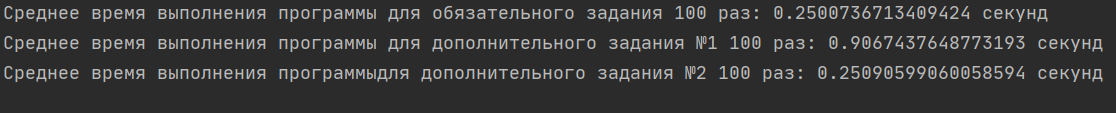


Рис.8

Сравнивая результаты, можно заметить, что программа, не использующая готовые библиотеки и регулярные выражения, справляется быстрее всего. Это может быть связано с тем, что исходный файл содержит сложную структуру (массив ключей), которую сложно обрабатывать библиотеке pyyaml. Регулярные выражения так же не оптимизируют программу за счёт неиспользования сложных методов для строк в исходной программе. Можно сделать вывод о том, что использование специальных библиотек и регулярных выражений может замедлить программу парсинга.

# Дополнительное задание №5

1. Переписать исходную программу, чтобы она осуществляла парсинг и конвертацию исходного файла в любой другой формат (кроме JSON, YAML, XML, HTML): PROTOBUF, **TSV**, CSV, WML и т.п.
2. Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте.

## Решение

### Исходный код

Решение дополнительного задания №5 представлено на Рис. 9.

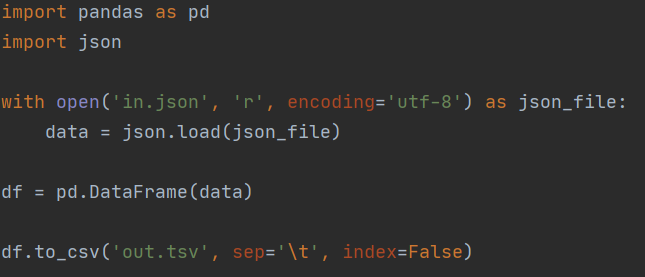


Рис. 9

### Результат

Результат выполнения кода дополнительного задания №5 представлен на Рис. 10.

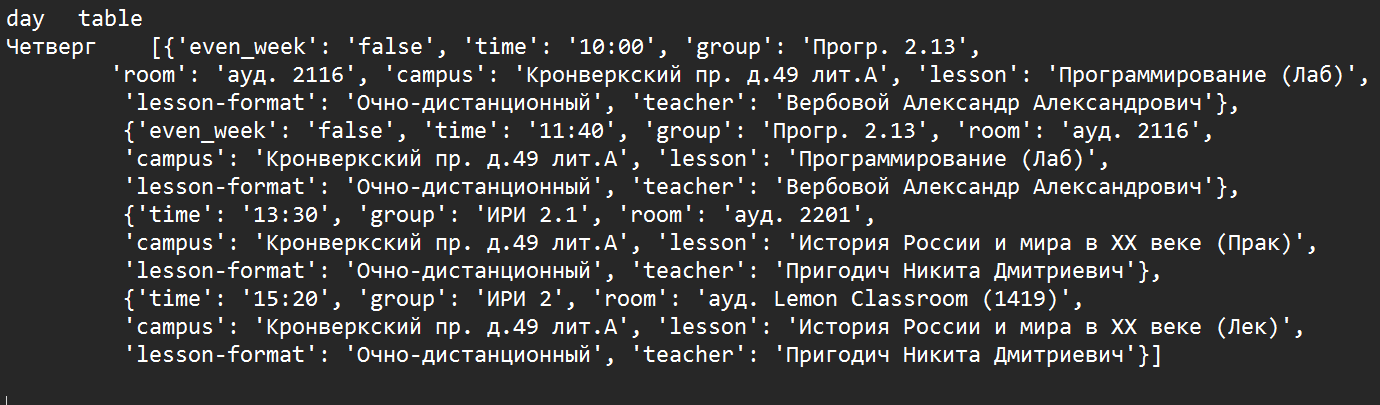


Рис. 10

## Вывод

Формат TSV, подобно CSV, предполагает, что данные упорядочены в таблице, где каждая строка содержит значения, разделенные табуляциями, и каждое значение соответствует столбцу таблицы. При конвертации JSON в TSV, можно убедиться, что структура моего JSON файла соответствует табличной структуре. Получилось два столбца, соответствующие дню недели и массиву ключей, то есть описанию расписания этого дня.

# Заключение

В результате выполнения лабораторной работы я ознакомилась с такими форматами, как JSON, YAML, TSV, научилась парсить файлы с помощью специальных библиотек и регулярных выражений в языке Python, закрепила и дополнила свои знания о библиотеке RegEx.

# Список источников

1. Регулярные выражения в Python от простого к сложному. Подробности, примеры, картинки, упражнения – URL: <https://habr.com/ru/articles/349860/>
2. Сравнение Json и YAML – URL: <https://www.pvsm.ru/java/70568>