Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №4

по информатике

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант 15. YAML to JSON

Выполнил:

Жеребцов М. А. Р3117

Проверил:

Марухленко Д.С.

Содержание

[Обязательное задание 3](#_Toc183019065)

[Дополнительное задание 1 3](#_Toc183019066)

[Дополнительное задание 2 3](#_Toc183019067)

[Дополнительное задание 3 3](#_Toc183019068)

[Дополнительное задание 4 4](#_Toc183019069)

[Заключение 5](#_Toc183019070)

# Обязательное задание

С помощью replace заменяю метасимволы в определенном порядке, полученным эмпирическим путем.

<https://github.com/mihozhereb/infa-lab-4/blob/main/main_task/main.py>

# Дополнительное задание 1

С помощью библиотек yaml и json конвертирую файл.

Отличие заключается в том, что с помощью обычного replace мне не удалось спарсить список чисел. В результате получился только список строк. Плюс при конвертации через библиотеку в точности соблюдаются все отступы в json файле. В остальном результат обоих программ схож.

<https://github.com/mihozhereb/infa-lab-4/blob/main/dop_task_1/dop1.py>

# Дополнительное задание 2

Заменил replace на re.sub, получился примерно такой же по объему код.

Отличие в том, что получилось с помощью регулярных выражений исправить проблему со списком чисел. Теперь он парсится нормально. В остальном получился такой же результат.

<https://github.com/mihozhereb/infa-lab-4/blob/main/dop_task_2/dop2.py>

# Дополнительное задание 3

Программа работает по принципу вложенности, которая в yaml обозначается пробелами. Скрипт переходит от уровня к уровню рекурсивно и обрабатывает каждый уровень изолированно от внешних. Используется несколько ключевых точек в разметке, исходя из которых поведение скрипта меняется. Далее полученный dict преобразуется в json, также рекурсивно. Результат записывается в файл.

Работа скрипта максимально схожа с встроенной python библиотекой, но, возможно, не обрабатывает все особенности синтаксиса yaml (тк их огромное количество). Но для базовых задач парсинга (словари, массивы, стандартные типы данных) скрипта достаточно.

Отличия от исходного скрипта аналогичны дополнительному заданию 1.

<https://github.com/mihozhereb/infa-lab-4/blob/main/dop_task_3/dop3.py>

# Дополнительное задание 4

В новый скрипт импортировал все предыдущие и замерил быстродействие (Рисунок 1. Результат проверки быстродействия).

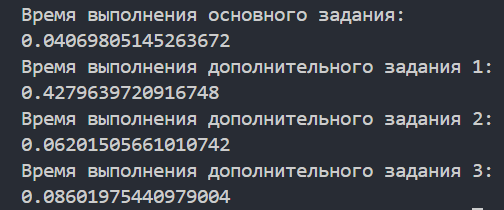


Рисунок 1. Результат проверки быстродействия

По полученным результатам можно судить, что самый быстрый скрипт – самый тупой. Самый медленный – самый умный. Регулярные выражения медленнее простого replace, при учете, что они выполняют одинаковую задачу. Скрипт, использующий формальные грамматики не сильно проигрывает «тупым» скриптам, хотя позволяет корректно парсить большинство yaml файлов. Также он намного быстрее библиотеки, поэтому его имеет смысл применять, когда yaml несложен, а быстродействие важно. Но, конечно, в случаях, когда наполнение yaml файла неизвестно, лучше использовать готовую библиотеку.

<https://github.com/mihozhereb/infa-lab-4/blob/main/dop4.py>

# Заключение

Я изучил различные языки разметки документов. Научился преобразовывать один язык в другой различными способами. Проанализировал плюсы и минусы каждого из способов.