

Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**Отчёт по лабораторной работе 4**

Тема: Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант: 8

Выполнил**:** студент группы Р3115 Храбров Артём Алексеевич

Проверил**:** Белокон Юлия Алексеевна

Дата сдачи: 18.11.2024

2024

Содержание

[Исходный файл в формате JSON 3](#_Toc182740629)

[Задание 1 3](#_Toc182740630)

[Задание 2 3](#_Toc182740631)

[Задание 3 4](#_Toc182740632)

[Задание 4 4](#_Toc182740633)

[Задание 5 4](#_Toc182740634)

[Задание 6 4](#_Toc182740635)

[Заключение 5](#_Toc182740636)

# Исходный файл в формате JSON

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/timetable.json>

# Задание 1

***Задание:***

Написать программу на Python, которая будет конвертировать json файл в yaml. Нельзя использовать сторонние библиотеки.

***Код программы:***

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/task1/main_1.py>

***Вывод программы:***

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/output/timetable_1.yaml>

# Задание 2

***Задание:***

Переписать код с использованием библиотек.

***Подготовка:***

Сначала установлю необходимую библиотеку:

pip install PyYaml

Дальше всё просто.

***Код программы:***

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/task2/main_2.py>

***Вывод программы:***

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/output/timetable_2.yaml>

***Сравнение с заданием 1:***

В готовых библиотеках в выходном файле строки записываются без кавычек, а числа с кавычками. В задании 1 в выходном файле все строки заключены в кавычки, а также после каждого элемента массива идет пустая строка.

# Задание 3

***Задание:***

Переписать код с использованием регулярный выражений.

***Код программы:***

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/task3/main_3.py>

***Вывод программы:***

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/output/timetable_3.yaml>

***Сравнение с заданием 1:***

Регулярные выражения используются для определения типа того, что в строке(массив, число и тд.). В остальном алгоритм как и в 1 задании.

# Задание 4

***Задание:***

Переписать код с использованием формальных грамматик.

***Код программы:***

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/task4/main_4.py>

***Вывод программы:***

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/output/timetable_4.yaml>

***Сравнение с заданием 1:***

В этом задании алгоритм работы совсем другой. Парсинг json происходит не за счёт обработки вложенных структур, а за счёт формальных грамматик. Время выполнения программы из-за этого сильно увеличилось.

# Задание 5

***Задание:***

Сравнить стократное время парсинга и конвертации в заданиях 1-4

***Решение:***

В каждый файл добавил эти строки в начало программы:

**from** time **import** time

start\_time **=** time**()**

И в конец:

**print(**f"Стократное время выполнения: {**(**time**()** **-** start\_time**)\***100}"**)**

Для каждого задания получилось найти стократное время его выполнения.

|  |  |
| --- | --- |
| Задание | Стократное время выполнения |
| Основное, без библиотек | 0.2591133117675781 |
| С использованием библиотек | 0.3000497817993164 |
| Регулярные выражения | 0.40047168731689453 |
| Формальные грамматики | 3.902149200439453 |

Время выполнения основной программы без библиотек, в среднем, показало самый быстрый результат, около 0.2-0.3. Библиотеки выдавали всегда около 0.3, а регулярные около 0.3-04. Это означает, что разница между ними незначительна. А вот формальные грамматики оказались в 10 раз медленнее других способов. Это значит, что в первых трёх пунктах схожие алгоритмы и выполняются одинаково быстро, в отличие от формальных грамматик.

# Задание 6

***Задание:***

Переписать программу, чтобы она осуществляла парсинг и конвертацию в другой формат. Сделаю преобразование в формат ProtoBuf.

***Подготовка:***

Сначала установлю необходимую библиотеку:

pip install py-json-to-proto

***Код программы:***

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/task6/main_6.py>

***Вывод программы:***

<https://github.com/artem961/INFORMATICS/blob/main/лр_4/output/timetable_6.proto>

***Сравнение с заданием 1:***

В этом задании алгоритм работы совсем другой. Парсинг json происходит не за счёт обработки вложенных структур, а за счёт формальных грамматик. Время выполнения программы из-за этого сильно увеличилось.

# Заключение

В ходе лабораторной работы я научился конвертировать json в yaml без всего, с регулярными выражениями, библиотеками и формальными грамматиками. Ещё с научился преобразованию в формат ProtoBuf, а также заёжился делать кучу доп. заданий с описанием.