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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант 9 (4663[09] % 36 = 9)

Выполнил: Косов Артём Андреевич

Группа: P3130

Проверил: преподаватель практики факультета ПИиКТ Рыбаков Степан Дмитриевич

Санкт-Петербург 2024

# Содержание

[Задание 3](#_Toc181822249)

[Основные этапы вычисления 6](#_Toc181822250)

[Обязательное задание 6](#_Toc181822251)

[Дополнительное задание №1 6](#_Toc181822252)

[Дополнительное задание №2 7](#_Toc181822253)

[Дополнительное задание №3 7](#_Toc181822254)

[Дополнительное задание №4 8](#_Toc181822255)

[Дополнительное задание №5 9](#_Toc181822256)

[Заключение 10](#_Toc181822257)

[Список использованных источников 11](#_Toc181822258)

# Задание

1. Определить номер варианта как остаток деления на 36 последних двух цифр своего идентификационного номера в ISU: например, 125598 / 36 = 26. В случае, если в оба указнных дня недели нет занятий, то увеличить номер варианта на восемь. В случае, если занятий нет и в новом наборе дней, то продолжать увеличивать на восемь.

2. Изучить форму Бэкуса-Наура.

3. Изучить основные принципы организации формальных грамматик.

4. Изучить особенности языков разметки/форматов JSON, YAML, XML.

5. Понять устройство страницы с расписанием на примере расписания

лектора: https://itmo.ru/ru/schedule/3/125598/raspisanie\_zanyatiy.htm

6. Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного. При этом необходимо, чтобы хотя бы в одной из выбранных дней было не менее двух занятий (можно использовать своё персональное). В случае, если в данный день недели нет таких занятий, то увеличить номер варианта ещё на восемь.

7. **Обязательное задание** (позволяет набрать до 45 процентов от максимального числа баллов БаРС за данную лабораторную): написать программу на языке Python 3.x или любом другом, которая бы осуществляла парсинг и конвертацию исходного файла в новый путём простой замены метасимволов исходного формата на метасимволы результирующего формата.

8. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.

9. **Дополнительное задание №1** (позволяет набрать +10 процентов от

максимального числа баллов БаРС за данную лабораторную).

a) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.

b) Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.

c) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

10. **Дополнительное задание №2** (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

a) Переписать исходный код, добавив в него использование регулярных выражений.

b) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

1. **Дополнительное задание №3** (позволяет набрать +25 процентов от максимального числа баллов БаРС за данную лабораторную).

а) Переписать исходный код таким образом, чтобы для решения задачи использовались формальные грамматики. То есть ваш код должен уметь осуществлять парсинг и конвертацию любых данных, представленных в исходном формате, в данные, представленные в результирующем формате: как с готовыми библиотеками из дополнительного задания №1.

b) Проверку осуществить как минимум для расписания с двумя учебными днями по два занятия в каждом.

с) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

1. **Дополнительное задание №4** (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).

a) Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле.

b) Проанализировать полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в

отчёте.

1. **Дополнительное задание №5** (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).

a) 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.

b) Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте

# Основные этапы вычисления

Условие варианта №9: YAML -> JSON Вторник, пятница

## Обязательное задание

Программа для обязательного задания построчно обрабатывает исходный файл путем замены метасимволов исходного формата, а также добавлением нужных строк при необходимости.

Исходный код программы представлен по данному URL-адресу:

<https://github.com/atemcozz/inf-lab4/blob/master/parser/t0_dumb_parser.py>

## Дополнительное задание №1

В качестве библиотеки для парсинга файла был выбран модуль pyyaml, позволяющий осуществить преобразование исходной строки в объект-словарь, который позже переводится в строку посредством встроенного модуля json. Исходный код сильно сократился из-за инкапсуляции логики библиотеки, а результат выполнения представлен в виде единой строки и может быть видоизменен, т.к. библиотека представила результат в виде объекта, а не абстрактной строки. Исходный код программы представлен на рисунке 2.

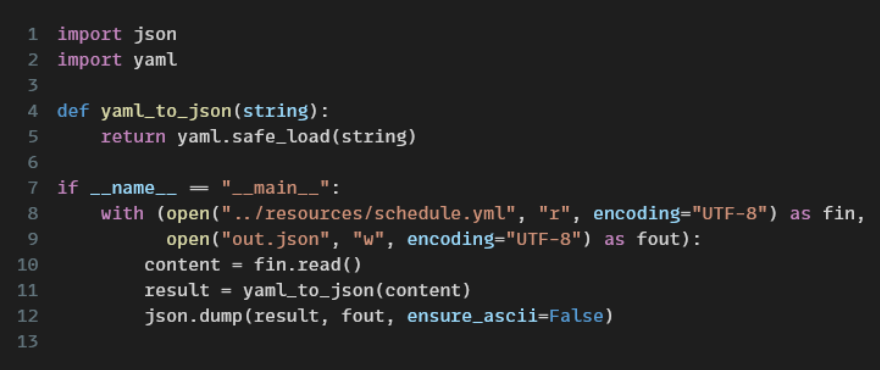


Рисунок 1 — программа с использованием библиотек

## Дополнительное задание №2

Программа из обязательного задания была переписано мною с использованием регулярных выражений, в частности, модуля re. Исходный код стал заметно короче за счёт компактности регулярных выражений. Результат для файла с расписанием аналогичен результату программы для обязательного задания, однако поведение программы стало менее предсказуемым из-за сложности и вариативности выражений.

Исходный код программы представлен по данному URL-адресу: <https://github.com/atemcozz/inf-lab4/blob/master/parser/t2_regex_parser.py>

## Дополнительное задание №3

Исходный код был переписан с использованием формальных грамматикой, т.е. во время исполнения исходная строка разбирается на подстроки по формам Бэкуса-Наура данного формата. Результат выполнения программы не имеет заметных отличий от результата программы из дополнительного задания 1 из-за схожестей в работе, однако при работе со некоторыми документами результат может отличаться из-за специфичности и сложности интерпретации формата YAML.

Исходный код программы представлен по данному URL-адресу: <https://github.com/atemcozz/inf-lab4/blob/master/parser/t3_fgrammar_parser.py>

## Дополнительное задание №4

Для сравнения производительности полученных программ было использовано стократной исполнение каждой программы с замером времени исполнения. Результаты замеров представлены на рисунке 2. Самым быстрым оказался способ с использованием формальных грамматик из-за линейной интерпретации документа, вторым является способ с простой заменой строк из-за отсутствия комплексных входных данных при сравнении и простоты операций. Далее за ним следует способ с использованием регулярных выражений, имеющий выраженное отставание от предыдущего способа из-за сложности компиляции и интерпретации регулярных выражений. Способы с заменой метасимволом не показывают сильных различий в производительности по сравнению с остальными в силу того, что они сразу преобразуют исходный формат в необходимый, минуя представление в объектном виде. Последнюю позицию в списке занимает способ с использованием готовых библиотек. Данная программа заметно отстает даже от самых простых программ ввиду попытке полного соответствия спецификации YAML и его множества конструкций.

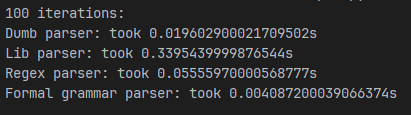


Рисунок 2 — результаты замеров производительности программ

Исходный код программы представлен по данному URL-адресу:

## Дополнительное задание №5

В качестве основы для программы была использована программа из доп. задания №3, а в качестве целевого формата - формат csv. Данный формат используется для представления табличных значений через символы “;” и переноса строки, а значит не поддерживает вложенные документы. Для того, чтобы конвертация была возможна, необходима ручная выборка данных из преобразованного документа. В моем случае вручную осуществляется выбор ключей и значений из объекта расписания, которые затем переводятся в плоские списки, которые затем объединяются в строку и записываются в файл:

Исходный код программы представлен по данному URL-адресу: <https://github.com/atemcozz/inf-lab4/blob/master/parser/t5_yaml_to_csv.py>

# Заключение

В ходе выполнения работы я ознакомился с принципом работы и устройством различных протоколов, форматов обмена информацией и языков разметки документов, таких как JSON, YAML, XML, CSV и т.п. Также я научился составлять, анализировать документы в различных форматах, осуществлять конвертацию между ними с использованием различных методов, выявлять разницу в производительности между различными способами.

# Список использованных источников

1. Лямин А.В., Череповская Е.Н. Объектно-ориентированное программирование. Компьютерный практикум. – СПб: Университет ИТМО, 2017. – 143 с. – Режим доступа: https://books.ifmo.ru/file/pdf/2256.pdf. (Дата обращения: 05.11.2024).
2. Википедия. Форма Бэкуса — Наура. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0\_%D0%91%D1%8D%D0%BA%D1%83%D1%81%D0%B0\_%E2%80%94\_%D0%9D%D0%B0%D1%83%D1%80%D0%B0. (Дата обращения: 05.11.2024).
3. Википедия. Формальная грамматика. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F\_%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0. (Дата обращения: 05.11.2024).