Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа №4**

**Исследование протоколов, форматов обмена информацией и языков разметки документов**

Вариант №466650 mod 36 = 18

Выполнил:

Марьин Григорий Алексеевич

Группа P3112

Проверил:

Доцент ПИиКТ, кандидат технических наук

Малышева Татьяна Алексеевна

Оглавление

[Задание 2](#_Toc182391663)

[Решение заданий: 4](#_Toc182391664)

[Обязательное задание 4](#_Toc182391665)

[Дополнительное задание №1. 4](#_Toc182391666)

[Дополнительное задание №2. 4](#_Toc182391667)

[Дополнительное задание №3. 4](#_Toc182391668)

[Дополнительное задание №4. 5](#_Toc182391669)

[Дополнительное задание №5. 6](#_Toc182391670)

[Вывод: 6](#_Toc182391671)

[Источники: 6](#_Toc182391672)

# 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. 1 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.

11.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.

12.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.

13.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. 2 b) Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте.

14.Проверить, что все пункты задания выполнены и выполнены верно.

15.Написать отчёт о проделанной работе.

16.Подготовиться к устным вопросам на защите.

# Решение заданий:

## Обязательное задание

Решение: <https://github.com/wwwyssa/wwwyssa_in_itmo/blob/3e184438a0944dda594caf35363938db205916fc/informatic/laba4/solves/mainTask.py>

## Дополнительное задание №1.

А) Для python есть библиотеки json и xml, которые содержат реализацию перевода из одного формата в иной.

Б) решение: <https://github.com/wwwyssa/wwwyssa_in_itmo/blob/3e184438a0944dda594caf35363938db205916fc/informatic/laba4/solves/dop1.py>

В) Единственное различие: <even\_week>True</even\_week> в файле, полученном с библиотекой. <even\_week>true</even\_week> в файле созданном без библиотек.

## Дополнительное задание №2.

А) Код основного задания был переписан с использованием регулярных выражений.

Б) Решение:

https://github.com/wwwyssa/wwwyssa\_in\_itmo/blob/3e184438a0944dda594caf35363938db205916fc/informatic/laba4/solves/dop2.py

В) Различий в файлах, полученных исходным алгоритмом и алгоритмом с регулярными выражениями нет.

## Дополнительное задание №3.

А) Для реализации парсера был написан класс ParserJSON. В данном классе реализованы несколько методов:

1. parse() – запускает парсинг файла. Результатом является класс dict() в python.
2. parse\_value() – запускает парсинг значения.
3. parse\_object() – запускает парсинг json объекта.
4. parse\_array() – запускает парсинг массива в json.
5. parse\_string() – запускает парсинг строки.
6. parse\_number() – запускает парсинг чисел.
7. peek() – возвращает текущую позицию «курсора»
8. skip\_whitespace() – пропускает все пробельные символы.

После получения словаря (dict()) запускается его перевод в xml формат. Его выполняет функция dict\_to\_xml(). Которая рекурсивно создает xml файл из полученного словаря.

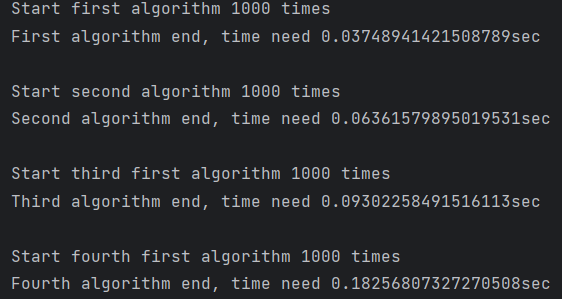
Б) решение: <https://github.com/wwwyssa/wwwyssa_in_itmo/blob/3e184438a0944dda594caf35363938db205916fc/informatic/laba4/solves/ParserJSON.py>

https://github.com/wwwyssa/wwwyssa\_in\_itmo/blob/3e184438a0944dda594caf35363938db205916fc/informatic/laba4/solves/dop3.py

В) Различий в файлах, полученных с помощью библиотек и данной реализацией нет.

## Дополнительное задание №4.

А) Решение: <https://github.com/wwwyssa/wwwyssa_in_itmo/blob/3e184438a0944dda594caf35363938db205916fc/informatic/laba4/solves/dop4.py>. Результат работы: Рисунок 1



Рисунок

Б) Самой быстрой оказалась первая реализация, так как в ней происходит простая замена метасимволов одного языка на метасимволы другого. Этот способ крайне не универсален.

Вторым по скорости оказалась реализация с использованием библиотек. Библиотеки написаны оптимизированно и выполняют задачу с максимальной эффективностью, что делает их быстрее третьего и четвертого алгоритма, но они все еще медленнее первого из-за того, что являются универсальными и проверяют больше различных параметров и условий.

Третьим по скорости оказались регулярные выражения. Сами по себе регулярные выражения не сильно быстрые, из-за этого и алгоритмы с их использованием не сильно быстрые.

Самой медленной оказалась собственная реализация. Это из-за того, что там почти нет никаких оптимизаций, происходит глубокая рекурсия, из-за чего алгоритм работает небыстро.

## Дополнительное задание №5.

А) Используя библиотеку bson, произведен перевод json файла в bson. Решение:

<https://github.com/wwwyssa/wwwyssa_in_itmo/blob/09f80d6c2968bda1a44c8a0087e472a870cfcc4c/informatic/laba4/solves/dop5.py>

Б) BSON (Binary JSON) — это бинарный формат сериализации данных. Более компактный и быстрый, чем JSON. Так же BSON поддерживает больше различных типов данных, например: даты, регулярные выражения.

# Вывод:

В ходе данной лабораторной работы я познакомился с различными типами файлов, научился с ними работать, понял их различие, научился использовать библиотеки python для работы с файлами различных типов данных.

# Источники:

1. <https://divancoder.ru/2017/06/backus-naur-form/>
2. <https://habr.com/ru/articles/177109/>
3. [bson – BSON (Binary JSON) Encoding and Decoding - PyMongo 4.10.1 documentation](https://pymongo.readthedocs.io/en/stable/api/bson/index.html)
4. <https://www.youtube.com/watch?v=XvDdCQxCYJ0>
5. <https://www.youtube.com/watch?v=xtjvAsrpKf4&t=642s>