МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И  
КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине

«ИНФОРМАТИКА»

Исследование протоколов,

форматов обмена информацией и языков разметки документов

Вариант №34

***Выполнил:***Студент группы P3107  
Софьин Вячеслав Евгеньевич

***Проверил:***Балакшин Павел Валерьевич

кандидат технических наук, доцент факультета ПИиКТ

**Содержание**

[Задание 3](#_Toc181641843)

[Основные этапы вычисления 5](#_Toc181641844)

[Расписание: 5](#_Toc181641845)

[Обязательное задание: 7](#_Toc181641846)

[Дополнительное задание №1: 7](#_Toc181641847)

[Дополнительное задание №2: 7](#_Toc181641848)

[Дополнительное задание №3: 7](#_Toc181641849)

[Дополнительное задание №4: 7](#_Toc181641850)

[Заключение 8](#_Toc181641851)

[Список литературы 9](#_Toc181641852)

# Задание

1. Определить номер варианта как остаток деления на 36 последних двух цифр своего идентификационного номера в ISU: например, 1255**98** / 36 = 26. В случае, если в оба указанных 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).
   1. a) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
   2. b) 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.
10. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
    1. Переписать исходный код, добавив в него использование регулярных выражений.
    2. 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).
    1. 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.
    2. Проверку осуществить как минимум для расписания с двумя учебными днями по два занятия в каждом.
    3. 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).
    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.
13. Дополнительное задание № 5 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).
    1. Переписать исходную программу, чтобы она осуществлял 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.
14. Проверить, что все пункты задания выполнены и выполнены верно.
15. Написать отчёт о проделанной работе.
16. Подготовиться к устным вопросам на защите.

# Основные этапы вычисления

## Расписание:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1 - Расписание в понедельник

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 2 - Расписание в субботу

На *рисунке 1* и *рисунке 2* отображено моё расписание. Ниже представлен вариант содержания yaml документа, с исходными данными.

days:  
 monday:  
 count\_lessons: 5  
 lesson:  
 - name: Программирование  
 type: Лабораторная  
 time: 8:20-9:50  
 lector: Данилов Павел Юрьевич  
 place: Ауд. 1338, Кронверкский пр., д.49, лит.А  
 - name: Программирование  
 type: Лабораторная  
 time: 10:00-11:30  
 lector: Данилов Павел Юрьевич  
 place: Ауд. 1338, Кронверкский пр., д.49, лит.А  
 - name: Линейная алгебра  
 type: Лекция  
 time: 15:20-16:50  
 lector: Карпов Дмитрий Валерьевич  
 place: Ауд. 1404, Кронверкский пр., д.49, лит.А  
 - name: Линейная алгебра  
 type: Практика  
 time: 17:00-18:30  
 lector: Карпов Дмитрий Валерьевич  
 place: Ауд. 2414, Кронверкский пр., д.49, лит.А  
 - name: Велоспорт  
 type: Спорт  
 time: 19:30-21:00  
 lector: Селищев Роман Александрович  
 place: Ауд. Локосфинкс, Северная дорога, 12А  
 saturday:  
 count\_lessons: 4  
 lesson:  
 - name: Математический анализ  
 type: Лекция  
 time: 8:20-9:50  
 lector: Холодова Светлана Евгеньевна  
 place: Ауд. 1405, Кронверкский пр., д.49, лит.А  
 - name: Математический анализ  
 type: Лекция  
 time: 10:00-11:30  
 lector: Холодова Светлана Евгеньевна  
 place: Ауд. 1405, Кронверкский пр., д.49, лит.А  
 - name: История русской культуры в контексте мировой культуры  
 type: Лекция  
 time: 17:00-18:30  
 lector: Жиркова Галина Петровна  
 - name: История русской культуры в контексте мировой культуры  
 type: Практика  
 time: 18:40-20:10  
 lector: Щеглова Иванна Павловна

## Обязательное задание:

Листинг: <https://github.com/safarislava/Informatics/blob/main/lab4/parser/stupid.py>

Результат: <https://github.com/safarislava/Informatics/blob/main/lab4/schedule_stupid.xml>

Особенности: yaml не содержит закрывающие теги в явном виде, поэтому в моей реализации есть стек поддерживающий табуляцию. Его поддержка позволяет правильно закрывать теги. В силу условия задания форматирование xml не идеально: нет поддержки одинарных тегов, тег не может открываться и закрываться в одной строке.

## Дополнительное задание №1:

Листинг: <https://github.com/safarislava/Informatics/blob/main/lab4/parser/ready.py>

Результат:

<https://github.com/safarislava/Informatics/blob/main/lab4/schedule_ready.xml>

Сравнение: Готовое решение имеет настройки по оформлению xml файла, поэтому выглядит и структурирован лучше, чем простая замена метасимволов. Однако файлы полностью совпадают по тегам и значениям. Ещё одно отличие – готовое решение воспринимает тег без значения как просто значение.

## Дополнительное задание №2:

Листинг: <https://github.com/safarislava/Informatics/blob/main/lab4/parser/regular.py>

Результат:

<https://github.com/safarislava/Informatics/blob/main/lab4/schedule_regular.xml>

Сравнение: Данная реализация по результату работы не отличается от первой, но регулярные выражения упрощают работу со строками, поэтому исходный код легче читается и воспринимается.

## Дополнительное задание №3:

Листинг: <https://github.com/safarislava/Informatics/blob/main/lab4/parser/formal.py>

Результат:

<https://github.com/safarislava/Informatics/blob/main/lab4/schedule_formal.xml>

Сравнение: Парсер, использующий формальные грамматики сильно лучше первой и третьей реализации. Этот подход позволяет форматировать xml, как угодно, поэтому его результат работы точно такой, как у готовой библиотеки.

## Дополнительное задание №4:

Тестирование скорости работы проходило при помощи тысячекратного повторения конвертации из yaml в xml без чтения/записи в файл.

Время выполнения реализации 1: 0.1077728271484375

Время выполнения реализации 2: 2.549070119857788

Время выполнения реализации 3: 0.07314395904541016

Время выполнения реализации 4: 0.20739984512329102

Первая реализация довольно быстрая, так как никаких сложных операций не выполняет, из дополнительных структур создаёт только стек, поддерживающий табуляцию строк, чтобы расставлять закрывающие теги.

Вторая реализация оказалась сильно медленнее. Это связано с тем, что происходит дополнительная конвертация в словарь.

Третья реализация - самая быстрая. Регулярные выражения реализуют быстрый поиск некого паттерна, что увеличивает скорость общего выполнения программы.

Четвёртая реализация выполняется в 2 раза дольше, чем первая. Это происходит, потому что использование формальных грамматик я реализовал через рекурсию, а она на языке программирования Python работает медленнее, чем стек. В ней сначала создаётся рекуррентное дерево тегов, а только после – строится xml.

# Заключение

Я познакомился с основными протоколами данных и их синтаксисом. Придумал способ перевода из yaml в xml без формальных грамматик, основанный на стеке и поддерживающий конвертацию массивов. Отточил навыки рекуррентных структур данных. Обеспечил себе работы в ROSTIC’S, так как научился перекладывать данные из одного JSON в другой.

# Список литературы

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>.
2. YAML Ain't Markup Language : сайт. – URL: https://yaml.org/spec/1.2.2/ (дата обращения: 06.11.2024)