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

«Исследование протоколов,

форматов обмена информацией и языков разметки

документов»

Факультет Программной инженерии и компьютерной техники

Дисциплина “Информатика”

Отчёт по лабораторной работе №4

Вариант №29

Выполнил:

Ануфриев Андрей Сергеевич, Р3119

Проверил:

Авксентьева Елена Юрьевна, к. п. н., доцент

г. Санкт-Петербург

2024 год

Оглавление

[Порядок выполнения работы 3](#_Toc183044731)

[Обязательное задание 5](#_Toc183044732)

[Исходный текст 5](#_Toc183044733)

[Код 6](#_Toc183044734)

[Вывод 8](#_Toc183044735)

[Дополнительное задание №1 9](#_Toc183044736)

[Код 9](#_Toc183044737)

[Вывод 9](#_Toc183044738)

[Дополнительное задание №2 11](#_Toc183044739)

[Код 11](#_Toc183044740)

[Вывод 11](#_Toc183044741)

[Дополнительное задание № 3 13](#_Toc183044742)

[Ввод 13](#_Toc183044743)

[Код 16](#_Toc183044744)

[Вывод 19](#_Toc183044745)

[Дополнительное задание № 4 22](#_Toc183044746)

[Стократное время выполнения 22](#_Toc183044747)

[Вывод 22](#_Toc183044748)

[Вывод по работе 23](#_Toc183044749)

[Источники 23](#_Toc183044750)

Порядок выполнения работы

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.

# Обязательное задание

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

## Исходный текст

*<?*xml version="1.0" encoding="UTF-8"*?>*<weekday> ср, 6 ноября  
<day\_schedule>  
 <lesson\_1>  
 <time>  
 <lesson\_time\_start> 08:20 </lesson\_time\_start>  
 <lesson\_time\_end> 09:50 </lesson\_time\_end>  
 </time>  
 <body>  
 <lesson\_type> Лекция </lesson\_type>  
 <lesson\_name> Информатика </lesson\_name>  
 <lesson\_teacher> Балакшин Павел Валерьевич </lesson\_teacher>  
 <lesson\_place>  
 <room> Ауд. Актовый зал (1216/0 (усл)) </room>  
 <address> ул.Ломоносова, д.9, лит.М </address>  
 </lesson\_place>  
 </body>  
 </lesson\_1>  
 <lesson\_2>  
 <time>  
 <lesson\_time\_start> 10:00 </lesson\_time\_start>  
 <lesson\_time\_end> 11:30 </lesson\_time\_end>  
 </time>  
 <body>  
 <lesson\_type> Лекция </lesson\_type>  
 <lesson\_name> Основы профессиональной деятельности </lesson\_name>  
 <lesson\_teacher> Клименков Сергей Викторович</lesson\_teacher>  
 <lesson\_place>  
 <room> Ауд. Актовый зал (1216/0 (усл)) </room>  
 <address> ул.Ломоносова, д.9, лит.М </address>  
 </lesson\_place>  
 </body>  
 </lesson\_2>  
 <lesson\_3>  
 <time>  
 <lesson\_time\_start> 19:00 </lesson\_time\_start>  
 <lesson\_time\_end> 21:00 </lesson\_time\_end>  
 </time>  
 <body>  
 <lesson\_type> Занятие спортом </lesson\_type>  
 <lesson\_name> Гребной спорт (мужская сборная команда) </lesson\_name>  
 <lesson\_teacher> Фомченко Андрей Александрович </lesson\_teacher>  
 <lesson\_place>  
 <room> Ауд. Гребной клуб "Стрела" </room>  
 <address> Северная дорога, 25 </address>  
 </lesson\_place>  
 </body>  
 </lesson\_3>  
</day\_schedule>  
</weekday>

## Код

import time  
def xml\_to\_yaml(xml\_input):  
 *# Функция для удаления лишних пробелов и переноса строк* def clean\_text(text):  
 return ' '.join(text.split()).strip()  
 yaml\_output = []  
 indent\_level = 0  
 *# Перебираем строки входного XML* for line in xml\_input.splitlines():  
 line = clean\_text(line)  
  
 if line.startswith('<?xml'):  
 continue *# Игнорируем строку с заголовком XML* if not line: *# Игнорируем пустые строки* continue  
  
 *# Если строка открывающий тег* if line.startswith('<') and not line.startswith('</'):  
 tag\_name = line[1:line.find('>')]  
  
 *# Добавляем отступы для YAML* yaml\_output.append(' ' \* indent\_level + f"{tag\_name}"+":"+line[line.find(">")+1:line.find("</")])  
 indent\_level += 1  
 if "</"in line: indent\_level -=1  
  
 *# Если строка закрывающий тег* elif line.startswith('</'):  
 indent\_level -= 1  
  
 *# Если это текстовый элемент* else:  
 *# Извлекаем текстовые данные* text\_value = clean\_text(line)  
 yaml\_output.append(' ' \* indent\_level + f"- {text\_value}n")  
  
 return yaml\_output  
  
start\_time = time.perf\_counter()  
*# открываем файл*with open('Среда.xml', 'r', encoding= "utf-8") as file:  
 xml\_file = file.read()  
*# записываем в другой файл*with open('output0.yaml', 'w', encoding= "utf-8") as file:  
 for i in xml\_to\_yaml(xml\_file):  
 file.write(i)  
*# проверка, чисто для себя*print("время =", 100 \* (time.perf\_counter() - start\_time), " Всё должно быть в файле, но вот пример:")  
for i in xml\_to\_yaml(xml\_file):  
 print(i)

## Вывод

weekday: ср, 6 ноябр day\_schedule: lesson\_1: time: lesson\_time\_start: 08:20 lesson\_time\_end: 09:50 body: lesson\_type: Лекция lesson\_name: Информатика lesson\_teacher: Балакшин Павел Валерьевич lesson\_place: room: Ауд. Актовый зал (1216/0 (усл)) address: ул.Ломоносова, д.9, лит.М lesson\_2: time: lesson\_time\_start: 10:00 lesson\_time\_end: 11:30 body: lesson\_type: Лекция lesson\_name: Основы профессиональной деятельности lesson\_teacher: Клименков Сергей Викторович lesson\_place: room: Ауд. Актовый зал (1216/0 (усл)) address: ул.Ломоносова, д.9, лит.М lesson\_3: time: lesson\_time\_start: 19:00 lesson\_time\_end: 21:00 body: lesson\_type: Занятие спортом lesson\_name: Гребной спорт (мужская сборная команда) lesson\_teacher: Фомченко Андрей Александрович lesson\_place: room: Ауд. Гребной клуб "Стрела" address: Северная дорога, 25

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.

## Код

import xmltodict  
import yaml  
import time  
*# Функция для преобразования XML в YAML*def convert\_xml\_to\_yaml(xml\_file, yaml\_file):  
 *# Чтение XML файла* with open(xml\_file, 'r', encoding='utf-8') as file:  
 xml\_content = file.read()  
  
 *# Преобразование XML в Python словарь* dict\_data = xmltodict.parse(xml\_content)  
  
 *# Преобразование словаря в YAML и сохранение в файл* with open(yaml\_file, 'w', encoding='utf-8') as file:  
 yaml.dump(dict\_data, file, allow\_unicode=True, default\_flow\_style=False)  
  
start\_time = time.perf\_counter()  
*# Пример использования функции*xml\_file = 'Среда.xml' *# Укажите здесь путь к вашему XML файлу*yaml\_file = 'output1.yaml' *# Укажите сюда желаемое имя YAML файла*convert\_xml\_to\_yaml(xml\_file, yaml\_file)  
print('Время выполнения:' , 100 \* (time.perf\_counter() - start\_time))  
print(f"Преобразование завершено. YAML файл сохранен как: {yaml\_file}")

## Вывод

weekday: ср, 6 ноября  
day\_schedule:  
 lesson\_1:  
 body:  
 lesson\_name: Информатика  
 lesson\_place:  
 address: ул.Ломоносова, д.9, лит.М  
 room: Ауд. Актовый зал (1216/0 (усл))  
 lesson\_teacher: Балакшин Павел Валерьевич  
 lesson\_type: Лекция  
 time:  
 lesson\_time\_end: 09:50  
 lesson\_time\_start: 08:20  
 lesson\_2:  
 body:  
 lesson\_name: Основы профессиональной деятельности  
 lesson\_place:  
 address: ул.Ломоносова, д.9, лит.М  
 room: Ауд. Актовый зал (1216/0 (усл))  
 lesson\_teacher: Клименков Сергей Викторович  
 lesson\_type: Лекция  
 time:  
 lesson\_time\_end: '11:30'  
 lesson\_time\_start: '10:00'  
 lesson\_3:  
 body:  
 lesson\_name: Гребной спорт (мужская сборная команда)  
 lesson\_place:  
 address: Северная дорога, 25  
 room: Ауд. Гребной клуб "Стрела"  
 lesson\_teacher: Фомченко Андрей Александрович  
 lesson\_type: Занятие спортом  
 time:  
 lesson\_time\_end: '21:00'  
 lesson\_time\_start: '19:00'

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.

## Код

import re  
import time  
def xml\_to\_yaml(xml):  
 s = xml.read()  
 s = re.sub(' ', '', s)  
  
 s = re.sub(r'(?:</.\*>)|(?:<\?.\*\?>)|<', '', s)  
 s = re.sub('>', ':', s)  
 *#s = re.sub('lesson:', 'lesson:', s)* s = re.sub(r'^s\*lesson:', ' lesson:', s, flags=re.MULTILINE)  
 s = re.sub(r'^s\*\n$', '', s, flags=re.MULTILINE)  
 s = re.sub(r' {2,}', ' ', s, flags=re.MULTILINE)  
 s = re.sub('^\n', '', s, flags=re.MULTILINE)  
 s = re.sub(r'^\s\*\n', '', s, flags=re.MULTILINE)  
 YAML.write(s)  
  
start\_time = time.perf\_counter()  
XML = open("Среда.xml", mode="r", encoding="utf-8")  
YAML = open("output2.yaml", mode="w", encoding="utf-8")  
xml\_to\_yaml(XML)  
XML.close()  
YAML.close()  
print("Всё готово! Проверяйте! Время выполнения:", 100 \* (time.perf\_counter() - start\_time))

## Вывод

weekday: ср, 6 ноября  
day\_schedule:  
 lesson\_1:  
 time:  
 lesson\_time\_start: 08:20   
 lesson\_time\_end: 09:50   
 body:  
 lesson\_type: Лекция   
 lesson\_name: Информатика   
 lesson\_teacher: Балакшин Павел Валерьевич   
 lesson\_place:  
 room: Ауд. Актовый зал (1216/0 (усл))   
 address: ул.Ломоносова, д.9, лит.М   
 lesson\_2:  
 time:  
 lesson\_time\_start: 10:00   
 lesson\_time\_end: 11:30   
 body:  
 lesson\_type: Лекция   
 lesson\_name: Основы профессиональной деятельности   
 lesson\_teacher: Клименков Сергей Викторович  
 lesson\_place:  
 room: Ауд. Актовый зал (1216/0 (усл))   
 address: ул.Ломоносова, д.9, лит.М   
 lesson\_3:  
 time:  
 lesson\_time\_start: 19:00   
 lesson\_time\_end: 21:00   
 body:  
 lesson\_type: Занятие спортом   
 lesson\_name: Гребной спорт (мужская сборная команда)   
 lesson\_teacher: Фомченко Андрей Александрович   
 lesson\_place:  
 room: Ауд. Гребной клуб "Стрела"   
 address: Северная дорога, 25

# 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.

## Ввод

*<?*xml version="1.0" encoding="UTF-8"*?>*<week\_schedule>  
<weekday1> ср, 6 ноября  
<day\_schedule>  
<lesson\_1>  
 <time>  
 <lesson\_time\_start> 08:20 </lesson\_time\_start>  
 <lesson\_time\_end> 09:50 </lesson\_time\_end>  
 </time>  
 <body>  
 <lesson\_type> Лекция </lesson\_type>  
 <lesson\_name> Информатика </lesson\_name>  
 <lesson\_teacher> Балакшин Павел Валерьевич </lesson\_teacher>  
 <lesson\_place>  
 <room> Ауд. Актовый зал (1216/0 (усл)) </room>  
 <address> ул.Ломоносова, д.9, лит.М </address>  
 </lesson\_place>  
 </body>  
</lesson\_1>  
<lesson\_2>  
 <time>  
 <lesson\_time\_start> 10:00 </lesson\_time\_start>  
 <lesson\_time\_end> 11:30 </lesson\_time\_end>  
 </time>  
 <body>  
 <lesson\_type> Лекция </lesson\_type>  
 <lesson\_name> Основы профессиональной деятельности </lesson\_name>  
 <lesson\_teacher> Клименков Сергей Викторович</lesson\_teacher>  
 <lesson\_place>  
 <room> Ауд. Актовый зал (1216/0 (усл)) </room>  
 <address> ул.Ломоносова, д.9, лит.М </address>  
 </lesson\_place>  
 </body>  
</lesson\_2>  
<lesson\_3>  
 <time>  
 <lesson\_time\_start> 19:00 </lesson\_time\_start>  
 <lesson\_time\_end> 21:00 </lesson\_time\_end>  
 </time>  
 <body>  
 <lesson\_type> Занятие спортом </lesson\_type>  
 <lesson\_name> Гребной спорт (мужская сборная команда) </lesson\_name>  
 <lesson\_teacher> Фомченко Андрей Александрович </lesson\_teacher>  
 <lesson\_place>  
 <room> Ауд. Гребной клуб "Стрела" </room>  
 <address> Северная дорога, 25 </address>  
 </lesson\_place>  
 </body>  
</lesson\_3>  
</day\_schedule>  
</weekday1>  
<weekday2> ср, 6 ноября  
<day\_schedule>  
<lesson\_1>  
 <time>  
 <lesson\_time\_start> 08:20 </lesson\_time\_start>  
 <lesson\_time\_end> 09:50 </lesson\_time\_end>  
 </time>  
 <body>  
 <lesson\_type> Лекция </lesson\_type>  
 <lesson\_name> Информатика </lesson\_name>  
 <lesson\_teacher> Балакшин Павел Валерьевич </lesson\_teacher>  
 <lesson\_place>  
 <room> Ауд. Актовый зал (1216/0 (усл)) </room>  
 <address> ул.Ломоносова, д.9, лит.М </address>  
 </lesson\_place>  
 </body>  
</lesson\_1>  
<lesson\_2>  
 <time>  
 <lesson\_time\_start> 10:00 </lesson\_time\_start>  
 <lesson\_time\_end> 11:30 </lesson\_time\_end>  
 </time>  
 <body>  
 <lesson\_type> Лекция </lesson\_type>  
 <lesson\_name> Основы профессиональной деятельности </lesson\_name>  
 <lesson\_teacher> Клименков Сергей Викторович</lesson\_teacher>  
 <lesson\_place>  
 <room> Ауд. Актовый зал (1216/0 (усл)) </room>  
 <address> ул.Ломоносова, д.9, лит.М </address>  
 </lesson\_place>  
 </body>  
</lesson\_2>  
<lesson\_3>  
 <time>  
 <lesson\_time\_start> 19:00 </lesson\_time\_start>  
 <lesson\_time\_end> 21:00 </lesson\_time\_end>  
 </time>  
 <body>  
 <lesson\_type> Занятие спортом </lesson\_type>  
 <lesson\_name> Гребной спорт (мужская сборная команда) </lesson\_name>  
 <lesson\_teacher> Фомченко Андрей Александрович </lesson\_teacher>  
 <lesson\_place>  
 <room> Ауд. Гребной клуб "Стрела" </room>  
 <address> Северная дорога, 25 </address>  
 </lesson\_place>  
 </body>  
</lesson\_3>  
</day\_schedule>  
</weekday2>  
</week\_schedule>

## Код

import re  
import time  
def XML\_to\_obj(XML\_file):  
 s = XML\_file.read()  
 tags\_iters = re.finditer(r'(?:<[^?/].+?>)|(?:</[^?].+?>)', s)  
  
 *#переведём из итераторов в многомерные массивы* tags = []  
 for i in tags\_iters:  
 if re.fullmatch(r'<[^?/].+?>', i.group()):  
 tags.append([i.group()[1:-1], "starting\_tag", i.end()])  
 if re.search(r'".\*?"', tags[-1][0]) is not None:  
 attributes = re.findall(r'\b\S+?=".\*?"', tags[-1][0])  
 tags[-1][0] = re.search(r'.+?(?= \b\S\*="\S+")', tags[-1][0]).group()  
 for j in attributes:  
 buf\_tag = re.search(r'\b\w+(?==)', j).group()  
 buf\_inf = re.search(r'(?<=").\*?(?=")', j).group()  
 tags.append([buf\_tag, "attribute", buf\_inf])  
 else:  
 tags.append([i.group()[2:-1], "ending\_tag", i.start()])  
  
 *#Соберём данные в словарь* stck = []  
 parent\_map = {}  
 child\_map = {}  
 for i in range(len(tags)):  
 if (tags[i][1] == "starting\_tag") or (tags[i][1] == "attribute"):  
 stck.append(tags[i])  
 else:  
 start\_num = -1  
 while stck[start\_num][1] == "attribute":  
 start\_num -= 1  
 text = s[stck[start\_num][2]:tags[i][2]]  
 *#Если элемент содержит дочерние элементы, то переложим его* if re.search(r'(?:<[^/].+?>)|(?:</.+?>)', text) is not None:  
 text = re.findall(r'(?<=<)(.+?)(?: \b\S+?=\".\*?\")\*(?=>)[\w\W]\*?(?<=</)\1(?=>)', text)  
 *#Удалим повторяющиеся теги* text\_set = set()  
 j = 0  
 while j < len(text):  
 if text[j] in text\_set:  
 text.remove(text[j])  
 else:  
 text\_set.add(text[j])  
 j += 1  
  
 *# Создадим словарь со всеми дочерними объектами* for child\_tag in text:  
 child\_map[child\_tag] = parent\_map[child\_tag]  
 parent\_map.pop(child\_tag)  
  
 if stck[start\_num][0] in parent\_map.keys():  
 parent\_map[stck[start\_num][0]].append(child\_map)  
 else:  
 parent\_map[stck[start\_num][0]] = [child\_map]  
 child\_map = {}  
 else:  
 parent\_map[stck[start\_num][0]] = text  
  
 while stck[-1][1] != "starting\_tag":  
 parent\_map[stck[start\_num][0]].insert(-1, ['attribute', stck[-1][0], stck[-1][2]])  
 start\_num += 1  
 stck.pop()  
 stck.pop()  
 return parent\_map  
  
new\_file\_str = ''  
def child\_obj(parent\_key, parent\_map, YAML\_file, tab):  
 global new\_file\_str  
 tab += 1  
 if isinstance(parent\_map, str):  
 new\_file\_str += tab \* ' ' + parent\_key + ': ' + parent\_map + '\n'  
 else:  
 tag\_is\_printed = False  
 for map\_object in parent\_map:  
 if not tag\_is\_printed:  
 if len(parent\_map) > 1:  
 new\_file\_str += (tab - 1) \* ' ' + '- ' + parent\_key + ':\n'  
 else:  
 new\_file\_str += tab \* ' ' + parent\_key + ':\n'  
  
 if isinstance(map\_object, dict):  
 for i in map\_object.keys():  
 child\_obj(i, map\_object[i], YAML\_file, tab)  
 tag\_is\_printed = False  
 else:  
 tag\_is\_printed = True  
 new\_file\_str += (tab + 1) \* ' ' + '\'@' + map\_object[1] + '\'' + ': ' + map\_object[2] + '\n'  
 YAML\_file.write(new\_file\_str)  
 new\_file\_str = ''  
 return parent\_map  
  
def obj\_to\_YAML(obj, YAML\_file):  
 for i in obj.keys():  
 child\_obj(i, obj[i], YAML\_file, -1)  
  
def convert():  
 timetableXML = open("2.xml", mode="r", encoding="utf-8")  
 timetableYAML = open("output3.yaml", mode="w", encoding="utf-8")  
  
 obj = XML\_to\_obj(timetableXML)  
 obj\_to\_YAML(obj, timetableYAML)  
start\_time = time.perf\_counter()  
if \_\_name\_\_ == "\_\_main\_\_":  
 convert()  
print("Всё готово! Проверяйте! Время выполнения:", 100 \* (time.perf\_counter() - start\_time))

## Вывод

week\_schedule:  
 weekday1:  
 day\_schedule:  
 lesson\_1:  
 time:  
 lesson\_time\_start: 08:20   
 lesson\_time\_end: 09:50   
 body:  
 lesson\_type: Лекция   
 lesson\_name: Информатика   
 lesson\_teacher: Балакшин Павел Валерьевич   
 lesson\_place:  
 room: Ауд. Актовый зал (1216/0 (усл))   
 address: ул.Ломоносова, д.9, лит.М   
 lesson\_2:  
 time:  
 lesson\_time\_start: 10:00   
 lesson\_time\_end: 11:30   
 body:  
 lesson\_type: Лекция   
 lesson\_name: Основы профессиональной деятельности   
 lesson\_teacher: Клименков Сергей Викторович  
 lesson\_place:  
 room: Ауд. Актовый зал (1216/0 (усл))   
 address: ул.Ломоносова, д.9, лит.М   
 lesson\_3:  
 time:  
 lesson\_time\_start: 19:00   
 lesson\_time\_end: 21:00   
 body:  
 lesson\_type: Занятие спортом   
 lesson\_name: Гребной спорт (мужская сборная команда)   
 lesson\_teacher: Фомченко Андрей Александрович   
 lesson\_place:  
 room: Ауд. Гребной клуб "Стрела"   
 address: Северная дорога, 25   
 weekday2:  
 day\_schedule:  
 lesson\_1:  
 time:  
 lesson\_time\_start: 08:20   
 lesson\_time\_end: 09:50   
 body:  
 lesson\_type: Лекция   
 lesson\_name: Информатика   
 lesson\_teacher: Балакшин Павел Валерьевич   
 lesson\_place:  
 room: Ауд. Актовый зал (1216/0 (усл))   
 address: ул.Ломоносова, д.9, лит.М   
 lesson\_2:  
 time:  
 lesson\_time\_start: 10:00   
 lesson\_time\_end: 11:30   
 body:  
 lesson\_type: Лекция   
 lesson\_name: Основы профессиональной деятельности   
 lesson\_teacher: Клименков Сергей Викторович  
 lesson\_place:  
 room: Ауд. Актовый зал (1216/0 (усл))   
 address: ул.Ломоносова, д.9, лит.М   
 lesson\_3:  
 time:  
 lesson\_time\_start: 19:00   
 lesson\_time\_end: 21:00   
 body:  
 lesson\_type: Занятие спортом   
 lesson\_name: Гребной спорт (мужская сборная команда)   
 lesson\_teacher: Фомченко Андрей Александрович   
 lesson\_place:  
 room: Ауд. Гребной клуб "Стрела"   
 address: Северная дорога, 25

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.

## Стократное время выполнения

Обязательное задание: Время выполнения: 0.11391998268663883

Доп1: Время выполнения: 0.7629899773746729

Доп2: Время выполнения: 0.15373998321592808

Доп3: Время выполнения: 0.352009991183877

Вывод

Самое быстрое время показало решение без использованием регулярных выражений и библиотек, так как оно было выполнена без многократных проходов по изначальному файлу. Самыми медленными оказались алгоритмы с использованием библиотек. Это связано с тем, что в таком случае требуется время на подгрузку библиотеки и она обладает значительно большей функциональностью.

# Вывод по работе

В ходе работы я узнал, как общаются сайты и те, кто их пишут, узнал про языки разметки, изучил JSON, XML, YAML, HTML. Узнал, что такое формальные грамматики.

# Источники

1. Балакшин П.В. Информатика Лекция 4, 2024 - <https://t.me/balakshin_students/267>
2. Convert XML to YAML in Python - <https://www.pythonforbeginners.com/basics/convert-xml-to-yaml-in-python>