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 22

Выполнила:

Хахулина Светлана Алексеевна

Группа:

Р3106

Проверил:

Доцент ПИиКТ, кандидат технических наук

Балакшин Павел Валерьевич

Санкт-Петербург, 2024

Оглавление

[Задание 3](#_Toc182593966)

[Решение 5](#_Toc182593967)

[Обязательное задание 5](#_Toc182593968)

[Дополнительное задание 1 8](#_Toc182593969)

[Дополнительное задание 2 10](#_Toc182593970)

[Дополнительное задание 3 13](#_Toc182593971)

[Дополнительное задание 4 14](#_Toc182593972)

[Дополнительное задание 5 15](#_Toc182593973)

[Вывод 17](#_Toc182593974)

[Источники 18](#_Toc182593975)

# Задание

1. 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.
2. 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.
3. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.
4. 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

1. 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.
2. 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).
2. 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.
3. Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте.

# Решение

## Обязательное задание

def yaml\_to\_xml(text):  
 xml = '<?xml version="1.0" encoding="UTF-8"?>\n'  
 tag0 = 0 #schedule  
 tag1 = 0 #день недели  
 last\_tag1 = 0  
 tag2 = 0 #class  
 tag = 0 #когда num\_tag принимает значения от 3 до 10  
 describe = 0 #когда num\_tag принимает значения от 3 до 10  
 cnt\_teg = 0  
 for i in text:  
 cnt\_space = len(i) - len(i.lstrip(' ')) #количество пробелов перед текстом(тегом)  
 num\_tag = cnt\_space // 2 #номер тега  
 i = i.split(':', maxsplit=1) #сплитим по : 1 раз потому что : также есть во времени  
 if num\_tag == 0:  
 tag0 = i[0]  
 xml += '<' + tag0 + '>' + "\n"  
 elif num\_tag == 1:  
 tag1 = i[0].lstrip(' ')  
 if last\_tag1 == 0:  
 last\_tag1 = tag1  
 else:  
 if tag1 != last\_tag1:  
 xml += ' ' \* cnt\_space + '</' + last\_tag1 + '>' + "\n"  
 xml += ' ' \* cnt\_space + '<' + tag1 + '>' + "\n"  
 elif num\_tag == 2:  
 tag2 = i[0].lstrip(' ')  
 xml += ' ' \* cnt\_space + '<' + tag2 + '>' + "\n"  
 elif num\_tag == 3:  
 tag = i[0].lstrip(' ')  
 describe = i[1]  
 xml += ' ' \* cnt\_space + '<' + tag + '>' + describe.lstrip(' ') + '</' + tag + '>' + '\n'  
 cnt\_teg += 1  
 if cnt\_teg == 8: #проверяем пора ли нам закрывать класс  
 xml += ' ' \* (cnt\_space - 2) + '</' + tag2 + '>' + '\n'  
 cnt\_teg = 0  
 xml += ' ' \* 2 + '</' + tag1 + '>' + '\n'  
 xml += '</' + tag0 + '>'  
 return xml  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 in\_yaml = ''  
 with open('schedule.yml', 'r', encoding='utf-8') as f:  
 in\_yaml = f.read().split('\n')  
 out\_xml = yaml\_to\_xml(in\_yaml)  
 print(out\_xml)  
 with open('1.xml', 'w', encoding='utf-8') as f:  
 f.write(out\_xml)

Результат

<?xml version="1.0" encoding="UTF-8"?>  
<schedule>  
 <tuesday>  
 <class1>  
 <title>Мат. анализ</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>10:00</begin\_time>  
 <end\_time>11:30</end\_time>  
 <teacher>Юрова Татьяна Сергеевна</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>2414</classroom>  
 </class1>  
 <class2>  
 <title>Лин. алгебра</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>11:40</begin\_time>  
 <end\_time>13:10</end\_time>  
 <teacher>Ефремов Илья Дмитриевич</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>1316</classroom>  
 </class2>  
 <class3>  
 <title>Англ. язык А2</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>15:20</begin\_time>  
 <end\_time>16:50</end\_time>  
 <teacher>Яковлева Ольга Александровна</teacher>  
 <building>ул. Ломоносова, д. 9, лит. Е</building>  
 <classroom>3211</classroom>  
 </class3>  
 <class4>  
 <title>Англ. язык А2</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>17:00</begin\_time>  
 <end\_time>18:30</end\_time>  
 <teacher>Яковлева Ольга Александровна</teacher>  
 <building>ул. Ломоносова, д. 9, лит. Е</building>  
 <classroom>3211</classroom>  
 </class4>  
 </tuesday>  
 <thursday>  
 <class1>  
 <title>История</title>  
 <type>Лекция</type>  
 <weeks>Нечётная</weeks>  
 <begin\_time>13:30</begin\_time>  
 <end\_time>15:00</end\_time>  
 <teacher>Белоусов Александр Сергеевич</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>1404</classroom>  
 </class1>  
 <class2>  
 <title>История</title>  
 <type>Практика</type>  
 <weeks>Нечётная</weeks>  
 <begin\_time>17:00</begin\_time>  
 <end\_time>18:30</end\_time>  
 <teacher>Белоусов Александр Сергеевич</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>2414</classroom>  
 </class2>  
 </thursday>  
</schedule>

## Дополнительное задание 1

import yaml  
import xmltodict  
  
  
def yaml\_to\_xml\_1(yaml\_file):# Преобразование словаря в XML  
 yaml\_dict = yaml.safe\_load(yaml\_file)  
 return xmltodict.unparse(yaml\_dict, pretty=True)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':# Чтение YAML-файла  
 with open('schedule.yml', 'r', encoding='utf-8') as yaml\_file:  
 yaml\_data = yaml\_to\_xml\_1(yaml\_file.read())  
 print(yaml\_data)  
   
 with open('dop1.xml', 'w', encoding='utf-8') as xml\_file:  
 xml\_file.write(yaml\_data)# Запись в файл

Результат

<?xml version="1.0" encoding="utf-8"?>  
<schedule>  
 <tuesday>  
 <class1>  
 <title>Мат. анализ</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>600</begin\_time>  
 <end\_time>690</end\_time>  
 <teacher>Юрова Татьяна Сергеевна</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>2414</classroom>  
 </class1>  
 <class2>  
 <title>Лин. алгебра</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>700</begin\_time>  
 <end\_time>790</end\_time>  
 <teacher>Ефремов Илья Дмитриевич</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>1316</classroom>  
 </class2>  
 <class3>  
 <title>Англ. язык А2</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>920</begin\_time>  
 <end\_time>1010</end\_time>  
 <teacher>Яковлева Ольга Александровна</teacher>  
 <building>ул. Ломоносова, д. 9, лит. Е</building>  
 <classroom>3211</classroom>  
 </class3>  
 <class4>  
 <title>Англ. язык А2</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>1020</begin\_time>  
 <end\_time>1110</end\_time>  
 <teacher>Яковлева Ольга Александровна</teacher>  
 <building>ул. Ломоносова, д. 9, лит. Е</building>  
 <classroom>3211</classroom>  
 </class4>  
 </tuesday>  
 <thursday>  
 <class1>  
 <title>История</title>  
 <type>Лекция</type>  
 <weeks>Нечётная</weeks>  
 <begin\_time>810</begin\_time>  
 <end\_time>900</end\_time>  
 <teacher>Белоусов Александр Сергеевич</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>1404</classroom>  
 </class1>  
 <class2>  
 <title>История</title>  
 <type>Практика</type>  
 <weeks>Нечётная</weeks>  
 <begin\_time>1020</begin\_time>  
 <end\_time>1110</end\_time>  
 <teacher>Белоусов Александр Сергеевич</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>2414</classroom>  
 </class2>  
 </thursday>  
</schedule>

Сравнение результатов обязательного задания и 1 дополнительного

Различия: в 1 допе используется функция safe\_load(), которая загружает данные из файла YAML в переменную в виде словаря, чего нет в обязательном задании, там мы храним теги и описания в переменных

Сходства: кроме результата сходств особых нет

## Дополнительное задание 2

import re  
  
  
def find\_teg(line):#так можно найти теги и запомнить их  
 match = re.search(r'\b\w+\b', line)  
 if match:  
 return match.group(0)  
  
  
def cnt\_spase(text): #функция возвращает кол-во 0 в начале строки  
 match = re.match(r'^\s\*', text)  
 return (len(match.group(0))) if match else 0  
  
  
def replace\_line(line): # -> <title>Мат. анализ</title>  
 line = re.sub(r'^(\s\*)(\w+):\s\*(.\*)', r'\1<\2>\3</\2>', line) + '\n'  
 return line  
  
  
def replace\_teg(line): # tuesday: -> <tuesday>  
 line = re.sub(r'(\s\*)(\w+):', r'\1<\2>', line) + '\n'  
 return line  
  
  
def close\_teg(cnt\_space, teg): # -> </tuesday>  
 return ' ' \* cnt\_space + '</' + teg + '>' + '\n'  
  
  
def yaml\_to\_xml\_2(yaml\_text):  
 xml\_text = '<?xml version="1.0" encoding="UTF-8"?>\n'  
 tegs = [0, 0, 0, 0]  
 for i in range(len(yaml\_text) - 1):  
 line = yaml\_text[i]  
 next\_line = yaml\_text[i + 1]  
 teg = find\_teg(line)  
 first\_space = cnt\_spase(line) #кол-во нулей в начале строки  
 num\_teg = first\_space // 2  
 if num\_teg == 0:  
 tegs[0] = teg  
 xml\_text += replace\_teg(line)  
 if num\_teg == 1: # день недели  
 xml\_text += replace\_teg(line)  
 tegs[1] = find\_teg(line)  
 if num\_teg == 2:  
 xml\_text += replace\_teg(line)  
 tegs[2] = find\_teg(line)  
 if num\_teg == 3: #информация  
 xml\_text += replace\_line(line)  
 if cnt\_spase(next\_line) == 4:  
 xml\_text += close\_teg(4, tegs[2]) #закрываем старый класс  
 tegs[2] = find\_teg(next\_line) #добавляем новый класс  
 elif cnt\_spase(next\_line) == 2:  
 xml\_text += close\_teg(4, tegs[2]) #закрываем класс  
 xml\_text += close\_teg(2, tegs[1]) #закрываем день недели  
 tegs[1] = find\_teg(next\_line) #обновляем день недели  
 xml\_text += close\_teg(4, tegs[2]) #закрываем класс  
 xml\_text += close\_teg(2, tegs[1]) #закрываем день  
 xml\_text += close\_teg(0, tegs[0])  
 return xml\_text

if \_\_name\_\_ == '\_\_main\_\_':  
 xml = ''  
 with open('schedule.yml', 'r', encoding='utf-8') as f:  
 in\_yaml = f.read().split('\n')  
 out\_xml = yaml\_to\_xml\_2(in\_yaml)  
 print(out\_xml)  
 with open('dop2.xml', 'w', encoding='utf-8') as f:  
 f.write(out\_xml)

Результат

<?xml version="1.0" encoding="UTF-8"?>  
<schedule>  
 <tuesday>  
 <class1>  
 <title>Мат. анализ</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>10:00</begin\_time>  
 <end\_time>11:30</end\_time>  
 <teacher>Юрова Татьяна Сергеевна</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>2414</classroom>  
 </class1>  
 <class2>  
 <title>Лин. алгебра</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>11:40</begin\_time>  
 <end\_time>13:10</end\_time>  
 <teacher>Ефремов Илья Дмитриевич</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>1316</classroom>  
 </class2>  
 <class3>  
 <title>Англ. язык А2</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>15:20</begin\_time>  
 <end\_time>16:50</end\_time>  
 <teacher>Яковлева Ольга Александровна</teacher>  
 <building>ул. Ломоносова, д. 9, лит. Е</building>  
 <classroom>3211</classroom>  
 </class3>  
 <class4>  
 <title>Англ. язык А2</title>  
 <type>Практика</type>  
 <weeks>Чётная, Нечётная</weeks>  
 <begin\_time>17:00</begin\_time>  
 <end\_time>18:30</end\_time>  
 <teacher>Яковлева Ольга Александровна</teacher>  
 <building>ул. Ломоносова, д. 9, лит. Е</building>  
 <classroom>3211</classroom>  
 </class4>  
 </tuesday>  
 <thursday>  
 <class1>  
 <title>История</title>  
 <type>Лекция</type>  
 <weeks>Нечётная</weeks>  
 <begin\_time>13:30</begin\_time>  
 <end\_time>15:00</end\_time>  
 <teacher>Белоусов Александр Сергеевич</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 <classroom>1404</classroom>  
 </class1>  
 <class2>  
 <title>История</title>  
 <type>Практика</type>  
 <weeks>Нечётная</weeks>  
 <begin\_time>17:00</begin\_time>  
 <end\_time>18:30</end\_time>  
 <teacher>Белоусов Александр Сергеевич</teacher>  
 <building>Кронверкский пр., д. 49, лит. А</building>  
 </class2>  
 </thursday>  
</schedule>

Сравнение результатов обязательного задания и 2 дополнительного

Сходства: результат и идея

Различия: во 2 дополнительном задании конвертация и парсинг осуществлялись через регулярные выражения

## Дополнительное задание 3

import re  
  
  
def dict\_to\_xml(data, indent=0):  
 xml\_str = ""  
 indentation = " " \* indent # 2 пробела \* количество  
  
 if isinstance(data, dict):  
 for key, value in data.items():  
 xml\_str += f"{indentation}<{key}>\n"  
 xml\_str += dict\_to\_xml(value, indent + 1)  
 xml\_str += f"{indentation}</{key}>\n"  
 else:  
 xml\_str += f"{indentation}{data}\n"  
 return xml\_str  
  
  
def parse\_yaml\_to\_dict(text):  
 result = {}  
 stack = [] #список тегов нужен чтобы понимать на каком уровне находишься  
 current\_dict = result #промежуточный словарь  
 levels = ['schedule']  
  
 for line in text.splitlines():  
 match = re.match(r'^(\s\*)([a-zA-Z0-9\_]+):\s\*(.\*)$', line)  
 if match:  
 spaces, key, value = match.groups()  
 level = len(spaces) // 2 + 1  
  
 while len(stack) >= level:  
 current\_dict = stack.pop()  
 levels.pop()  
  
 if value:  
 current\_dict[key] = value  
 else:  
 current\_dict[key] = {}  
 stack.append(current\_dict)  
 current\_dict = current\_dict[key]  
 levels.append(key)  
 return result  
  
  
def yaml\_to\_xml\_3(text):  
 dict\_data = parse\_yaml\_to\_dict(text)  
 first\_line = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n"  
 return first\_line + dict\_to\_xml(dict\_data)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 xml = ''  
 with open('schedule.yml', 'r', encoding='utf-8') as f:  
 in\_yaml = f.read()  
 out\_xml = yaml\_to\_xml\_3(in\_yaml)  
 print(out\_xml)  
 with open('dop3.xml', 'w', encoding='utf-8') as f:  
 f.write(out\_xml)

Результат

<?xml version="1.0" encoding="UTF-8"?>  
<schedule>  
 <tuesday>  
 <class1>  
 <title>  
 Мат. анализ  
 </title>  
 <type>  
 Практика  
 </type>  
 <weeks>  
 Чётная, Нечётная  
 </weeks>  
 <begin\_time>  
 10:00  
 </begin\_time>  
 <end\_time>  
 11:30  
 </end\_time>  
 <teacher>  
 Юрова Татьяна Сергеевна  
 </teacher>  
 <building>  
 Кронверкский пр., д. 49, лит. А  
 </building>  
 <classroom>  
 2414  
 </classroom>  
 </class1>  
 <class2>  
 <title>  
 Лин. алгебра  
 </title>  
 <type>  
 Практика  
 </type>  
 <weeks>  
 Чётная, Нечётная  
 </weeks>  
 <begin\_time>  
 11:40  
 </begin\_time>  
 <end\_time>  
 13:10  
 </end\_time>  
 <teacher>  
 Ефремов Илья Дмитриевич  
 </teacher>  
 <building>  
 Кронверкский пр., д. 49, лит. А  
 </building>  
 <classroom>  
 1316  
 </classroom>  
 </class2>  
 <class3>  
 <title>  
 Англ. язык А2  
 </title>  
 <type>  
 Практика  
 </type>  
 <weeks>  
 Чётная, Нечётная  
 </weeks>  
 <begin\_time>  
 15:20  
 </begin\_time>  
 <end\_time>  
 16:50  
 </end\_time>  
 <teacher>  
 Яковлева Ольга Александровна  
 </teacher>  
 <building>  
 ул. Ломоносова, д. 9, лит. Е  
 </building>  
 <classroom>  
 3211  
 </classroom>  
 </class3>  
 <class4>  
 <title>  
 Англ. язык А2  
 </title>  
 <type>  
 Практика  
 </type>  
 <weeks>  
 Чётная, Нечётная  
 </weeks>  
 <begin\_time>  
 17:00  
 </begin\_time>  
 <end\_time>  
 18:30  
 </end\_time>  
 <teacher>  
 Яковлева Ольга Александровна  
 </teacher>  
 <building>  
 ул. Ломоносова, д. 9, лит. Е  
 </building>  
 <classroom>  
 3211  
 </classroom>  
 </class4>  
 </tuesday>  
 <thursday>  
 <class1>  
 <title>  
 История  
 </title>  
 <type>  
 Лекция  
 </type>  
 <weeks>  
 Нечётная  
 </weeks>  
 <begin\_time>  
 13:30  
 </begin\_time>  
 <end\_time>  
 15:00  
 </end\_time>  
 <teacher>  
 Белоусов Александр Сергеевич  
 </teacher>  
 <building>  
 Кронверкский пр., д. 49, лит. А  
 </building>  
 <classroom>  
 1404  
 </classroom>  
 </class1>  
 <class2>  
 <title>  
 История  
 </title>  
 <type>  
 Практика  
 </type>  
 <weeks>  
 Нечётная  
 </weeks>  
 <begin\_time>  
 17:00  
 </begin\_time>  
 <end\_time>  
 18:30  
 </end\_time>  
 <teacher>  
 Белоусов Александр Сергеевич  
 </teacher>  
 <building>  
 Кронверкский пр., д. 49, лит. А  
 </building>  
 <classroom>  
 2414  
 </classroom>  
 </class2>  
 </thursday>  
</schedule>

Сравнение результатов обязательного задания и 3 дополнительного

Код для третьего долполнительного задания способен осуществить парсинг и конвертацию файла с любой вложенностью, а обязательный – только для четырех уровней вложенности

Результаты совпадают

# Дополнительное задание 4

import time  
from base import yaml\_to\_xml  
from dop\_1 import yaml\_to\_xml\_1  
from dop\_2 import yaml\_to\_xml\_2  
from dop\_3 import yaml\_to\_xml\_3  
  
iterations = 100  
yaml\_file\_name = 'schedule.yml'  
  
# основной вариант  
in\_yaml = ''  
with open(yaml\_file\_name, 'r', encoding='utf-8') as f:  
 in\_yaml = f.read().split('\n')  
start\_time = time.time()  
for \_ in range(iterations):  
 yaml\_to\_xml(in\_yaml)  
end\_time = time.time()  
total\_time = end\_time - start\_time  
print(f"Function yaml\_to\_xml took {total\_time:.6f} seconds for {iterations} iterations.")  
  
# вариант с использованием yaml и xmltodict  
in\_yaml1 = ''  
with open(yaml\_file\_name, 'r', encoding='utf-8') as f1:  
 in\_yaml1 = f1.read()  
  
start\_time = time.time()  
for \_ in range(iterations):  
 yaml\_to\_xml\_1(in\_yaml1)  
end\_time = time.time()  
total\_time = end\_time - start\_time  
print(f"Function yaml\_to\_xml\_1 took {total\_time:.6f} seconds for {iterations} iterations.")  
  
# вариант с использованием регулярных выражений  
with open(yaml\_file\_name, 'r', encoding='utf-8') as f3:  
 in\_yaml = f3.read().split('\n')  
start\_time = time.time()  
for \_ in range(iterations):  
 yaml\_to\_xml\_2(in\_yaml)  
end\_time = time.time()  
total\_time = end\_time - start\_time  
print(f"Function yaml\_to\_xml\_2 took {total\_time:.6f} seconds for {iterations} iterations.")  
  
# вариант с использованием формальных грамматик  
with open(yaml\_file\_name, 'r', encoding='utf-8') as f2:  
 in\_yaml = f2.read()  
start\_time = time.time()  
for \_ in range(iterations):  
 yaml\_to\_xml\_3(in\_yaml)  
end\_time = time.time()  
total\_time = end\_time - start\_time  
print(f"Function yaml\_to\_xml\_3 took {total\_time:.6f} seconds for {iterations} iterations.")

Результат

Function yaml\_to\_xml took 0.005040 seconds for 100 iterations.

Function yaml\_to\_xml\_1 took 0.324151 seconds for 100 iterations.

Function yaml\_to\_xml\_2 took 0.030055 seconds for 100 iterations.

Function yaml\_to\_xml\_3 took 0.009002 seconds for 100 iterations.

Сравнение полученных результатов

Программа обязательного задания работает быстрее всего, потому что она заточена на структуру входного файла, но не является универсальной. Доп 3 работает чуть дольше, потому что используется одно регулярное. Доп 2 на третьем месте, потому что используется множество регулярных выражений. Доп 1 работает дольше всех, потому что является универсальным парсером, в библиотеке реализованы все тонкости перевода YAML в XML.

## Дополнительное задание 5

def yaml\_to\_csv(text):  
 csv\_output = "Day,Class,Title,Type,Weeks,Begin Time,End Time,Teacher,Building,Classroom\n" # CSV заголовок  
 current\_day = None  
 current\_class = None  
 details\_parts = []  
  
 for i in text:  
 cnt\_space = len(i) - len(i.lstrip(' ')) # Количество пробелов перед текстом (тегом)  
 num\_tag = cnt\_space // 2 # Номер тега  
  
 i = i.split(':')  
 tag\_name = i[0].strip() # Получаем имя тега  
  
 if num\_tag == 0: # schedule  
 continue # Пропускаем, так как это заголовок  
  
 elif num\_tag == 1: # day  
 current\_day = tag\_name # Сохраняем текущий день  
  
 elif num\_tag == 2: # class  
 current\_class = tag\_name # Сохраняем текущий класс  
  
 elif num\_tag == 3: # details (например, описание или предмет)  
 if len(i) > 1:  
 details = i[1].strip() # Получаем значение  
 if details.find(',') > -1:  
 details\_parts.append( '"' + details + '"') # если значение содержит запятые  
 else:  
 details\_parts.append(details)  
 if len(details\_parts) == 8: # Ожидаем 8 элементов  
 # Добавляем строку в CSV, используя текущий день и класс  
 csv\_output += f"{current\_day},{current\_class}," + ','.join(details\_parts) + "\n"  
 details\_parts = []  
  
 return csv\_output  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 in\_yaml = ''  
 with open('schedule.yml', 'r', encoding='utf-8') as f:  
 in\_yaml = f.read().splitlines()  
  
 out\_csv = yaml\_to\_csv(in\_yaml) # Изменение функции на yaml\_to\_csv  
 print(out\_csv)  
  
 with open('1.csv', 'w', encoding='utf-8') as f: # Изменение файла на 1.csv  
 f.write(out\_csv)

Результат

Day,Class,Title,Type,Weeks,Begin Time,End Time,Teacher,Building,Classroom  
tuesday,class1,Мат. анализ,Практика,"Чётная, Нечётная",10,11,Юрова Татьяна Сергеевна,"Кронверкский пр., д. 49, лит. А",2414  
tuesday,class2,Лин. алгебра,Практика,"Чётная, Нечётная",11,13,Ефремов Илья Дмитриевич,"Кронверкский пр., д. 49, лит. А",1316  
tuesday,class3,Англ. язык А2,Практика,"Чётная, Нечётная",15,16,Яковлева Ольга Александровна,"ул. Ломоносова, д. 9, лит. Е",3211  
tuesday,class4,Англ. язык А2,Практика,"Чётная, Нечётная",17,18,Яковлева Ольга Александровна,"ул. Ломоносова, д. 9, лит. Е",3211  
thursday,class1,История,Лекция,Нечётная,13,15,Белоусов Александр Сергеевич,"Кронверкский пр., д. 49, лит. А",1404  
thursday,class2,История,Практика,Нечётная,17,18,Белоусов Александр Сергеевич,"Кронверкский пр., д. 49, лит. А",2414

Анализ

Заметим, что в формате CSV файлы занимают меньше места. Парсинг программ совпадает

# Вывод

В ходе выполнения лабораторной работы, я познакомилась с форматами YAML, XML, CSV. Также был получен опыт преобразования форматов между собой с помощью библиотек и самописного алгоритма. Кроме того, был повторно изучен материал для работы с регулярными выражениями.

# Источники

1. Балакшин П.В., Соснин В.В., Калинин И.В., Малышева Т.А., Раков С.В., Рущенко Н.Г., Дергачев А.М. Информатика: лабораторные работы и тесты: Учебно-методическое пособие / Рецензент: Поляков В.И. - Санкт-Петербург: Университет ИТМО, 2019. - 56 с.
2. Грошев А.С. Г89 Информатика: Учебник для вузов / А.С. Грошев. – Архангельск, Арханг. гос. техн. ун-т, 2010. -470с.