НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Информатика

Лабораторная работа № 4 “Исследование протоколов, форматов обмена информацией и языков разметки документов”

Выполнил студент:

Маликов Глеб Игоревич

Группа № Р3124

Преподаватель: Болдырева Елена Александровна

г. Санкт-Петербург

2022

Оглавление

[Обязательное задание 3](#_Toc119434147)

[Исходный код 3](#_Toc119434148)

[Ввод 4](#_Toc119434149)

[Вывод программы 6](#_Toc119434150)

[Дополнительное задание №1 9](#_Toc119434151)

[Исходный код 9](#_Toc119434152)

[Вывод программы 9](#_Toc119434153)

[Дополнительное задание №2 12](#_Toc119434154)

[Исходный код 12](#_Toc119434155)

[Вывод программы 13](#_Toc119434156)

[Дополнительное задание №3 17](#_Toc119434157)

[Исходный код 17](#_Toc119434158)

[Вывод программы 19](#_Toc119434159)

[Дополнительное задание №4 20](#_Toc119434160)

[Исходный код 20](#_Toc119434161)

[Вывод программы 22](#_Toc119434162)

[Вывод 24](#_Toc119434163)

[Список Литературы 25](#_Toc119434164)

# Обязательное задание

Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.

## Исходный код

def get\_tag\_text(text):

for i in range(len(text)):

if text[i] == "<":

start = i + 1

end = text.find(">", start, )

tussen = text[i + 1:end]

return tussen

return None

def get\_between\_tags(text):

end = text.find("<",)

tussen = text[:end]

return tussen

def replace\_w\_spaces(text):

if ' ' in text:

x = text.split('"')

x[1] = x[1].replace(' ', 'wEoN')

united = ''

for i in x:

if i == x[-1]:

return united

united += i + '"'

return united

else:

return text

def replace\_weon(text):

x = text.replace('wEoN', ' ')

return x

def xml\_to\_yaml(xmlinput: str):

yamlText = "---\n"

indentations = 0

temp = xmlinput.removeprefix("""<?xml version="1.0" encoding="UTF-8" ?>""")

while temp is not None:

between = get\_tag\_text(temp)

if between is None:

yamlText += "\n..."

yamlText = replace\_weon(yamlText)

return yamlText

to\_remove = "<" + between + ">"

end = temp.find(to\_remove) + len(to\_remove)

to\_remove = temp[0:end]

temp = temp.removeprefix(to\_remove)

if '"' in between:

between = replace\_w\_spaces(between)

between = between.split(" ")

comparador = []

for i in between:

if "/" in between[0]:

indentations -= 1

elif i == between[0]:

yamlText += (indentations \* " ") + i + ":\n"

elif len(between) > 1:

if comparador != between:

indentations += 1

i = i.split("=")

yamlText += (indentations \* " ") + i[0] + ": " + i[1] + "\n"

comparador = between

if (i == between[-1] or len(between) == 1) and not "/" in between[0]:

indentations += 1

temp = temp.lstrip()

if temp is not None and temp != '':

if temp[0] != '<':

data = get\_between\_tags(temp)

to\_remove = data

temp = temp.removeprefix(to\_remove)

data = data.lstrip()

if data != '':

yamlText += (indentations \* " ") + '"' + data + '"' + "\n"

yamlText += "\n..."

yamlText = replace\_weon(yamlText)

return yamlText

def add\_header\_xml(html):

xmlText = """<?xml version="1.0" encoding="UTF-8" ?>\n""" + html

return xmlText

def oblig(text):

x = add\_header\_xml(text)

y = xml\_to\_yaml(x)

outputFile = open("oblig.yml", "w")

outputFile.write(y)

## Ввод

<table id="2day" class="rasp\_tabl" cellspacing="0" cellpadding="0" border="0">

<tbody><tr><th class="day">

<script>$("#2day\_btn").show();</script>

<span>Вт</span></th> <td class="time">

<span>10:00-11:30</span>

<div>3, 7, 9, 11, 13, 15, 17</div>

<dd class="rasp\_aud\_mobile"></dd>

<dt class="rasp\_corp\_mobile"><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</td>

<td class="room">

<dl>

<dd></dd>

<dt><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</dl>

</td>

<td class="lesson">

<dl>

<dd>Основы профессиональной деятельности(Лек): актовый зал</dd><dt><b>

Клименков Сергей Викторович </b></dt>

</dl>

</td><td class="lesson-format">Очно - дистанционный </td>

</tr>

<tr><th class="day"></th> <td class="time">

<span>10:00-11:30</span>

<div>5</div>

<dd class="rasp\_aud\_mobile"></dd>

<dt class="rasp\_corp\_mobile"><i class="fa fa-map-marker"></i><span></span></dt>

</td>

<td class="room">

<dl>

<dd></dd>

<dt><i class="fa fa-map-marker"></i><span></span></dt>

</dl>

</td>

<td class="lesson">

<dl>

<dd>Основы профессиональной деятельности(Лек): ZOOM</dd><dt><b>

Клименков Сергей Викторович </b></dt>

</dl>

</td><td class="lesson-format">Дистанционный </td>

</tr>

<tr><th class="day"></th> <td class="time">

<span>10:00-11:30</span>

<div>2, 4, 6, 8, 10, 12, 14, 16</div>

<dd class="rasp\_aud\_mobile">1124 ауд.</dd>

<dt class="rasp\_corp\_mobile"><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</td>

<td class="room">

<dl>

<dd>1124 ауд.</dd>

<dt><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</dl>

</td>

<td class="lesson">

<dl>

<dd>Программирование(Лек)</dd>четная неделя<dt><b>

Письмак Алексей Евгеньевич </b></dt>

</dl>

</td><td class="lesson-format">Очно - дистанционный </td>

</tr>

<tr>

## Вывод программы

---

table:

id: "2day"

class: "rasp\_tabl"

cellspacing: "0"

cellpadding: "0"

border: "0"

tbody:

tr:

th:

class: "day"

script:

"$("#2day\_btn").show();"

span:

"Вт"

td:

class: "time"

span:

"10:00-11:30"

div:

"3, 7, 9, 11, 13, 15, 17"

dd:

class: "rasp\_aud\_mobile"

dt:

class: "rasp\_corp\_mobile"

i:

class: "fa fa-map-marker"

span:

"ул.Ломоносова, д.9, лит. М"

td:

class: "room"

dl:

dd:

dt:

i:

class: "fa fa-map-marker"

span:

"ул.Ломоносова, д.9, лит. М"

td:

class: "lesson"

dl:

dd:

"Основы профессиональной деятельности(Лек): актовый зал"

dt:

b:

"Клименков Сергей Викторович "

td:

class: "lesson-format"

"Очно - дистанционный "

tr:

th:

class: "day"

td:

class: "time"

span:

"10:00-11:30"

div:

"5"

dd:

class: "rasp\_aud\_mobile"

dt:

class: "rasp\_corp\_mobile"

i:

class: "fa fa-map-marker"

span:

td:

class: "room"

dl:

dd:

dt:

i:

class: "fa fa-map-marker"

span:

td:

class: "lesson"

dl:

dd:

"Основы профессиональной деятельности(Лек): ZOOM"

dt:

b:

"Клименков Сергей Викторович "

td:

class: "lesson-format"

"Дистанционный "

tr:

th:

class: "day"

td:

class: "time"

span:

"10:00-11:30"

div:

"2, 4, 6, 8, 10, 12, 14, 16"

dd:

class: "rasp\_aud\_mobile"

"1124 ауд."

dt:

class: "rasp\_corp\_mobile"

i:

class: "fa fa-map-marker"

span:

"ул.Ломоносова, д.9, лит. М"

td:

class: "room"

dl:

dd:

"1124 ауд."

dt:

i:

class: "fa fa-map-marker"

span:

"ул.Ломоносова, д.9, лит. М"

td:

class: "lesson"

dl:

dd:

"Программирование(Лек)"

"четная неделя"

dt:

b:

"Письмак Алексей Евгеньевич "

td:

class: "lesson-format"

"Очно - дистанционный "

tr:

...

# Дополнительное задание №1

1. Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
2. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.
3. Сравнить полученные результаты и объяснить их сходство/различие.

## Исходный код

import xmlplain

def first(text):

root = xmlplain.xml\_to\_obj(text, strip\_space=True, fold\_dict=True)

with open("1.yml", "w") as outf:

xmlplain.obj\_to\_yaml(root, outf)

## Вывод программы

table:

'@border': '0'

'@cellpadding': '0'

'@cellspacing': '0'

'@class': rasp\_tabl

'@id': 2day

tbody:

- tr:

- th:

'@class': day

script: $("#2day\_btn").show();

span: Вт

- td:

'@class': time

span: 10:00-11:30

div: 3, 7, 9, 11, 13, 15, 17

dd:

'@class': rasp\_aud\_mobile

dt:

'@class': rasp\_corp\_mobile

i:

'@class': fa fa-map-marker

span: ул.Ломоносова, д.9, лит. М

- td:

'@class': room

dl:

dd: ''

dt:

i:

'@class': fa fa-map-marker

span: ул.Ломоносова, д.9, лит. М

- td:

'@class': lesson

dl:

dd: 'Основы профессиональной деятельности(Лек): актовый зал'

dt:

b: "\n \

\ Клименков Сергей Викторович \

\ "

- td:

- '@class': lesson-format

- Очно - дистанционный

- tr:

- th:

'@class': day

- td:

'@class': time

span: 10:00-11:30

div: '5'

dd:

'@class': rasp\_aud\_mobile

dt:

'@class': rasp\_corp\_mobile

i:

'@class': fa fa-map-marker

span: ''

- td:

'@class': room

dl:

dd: ''

dt:

i:

'@class': fa fa-map-marker

span: ''

- td:

'@class': lesson

dl:

dd: 'Основы профессиональной деятельности(Лек): ZOOM'

dt:

b: "\n \

\ Клименков Сергей Викторович \

\ "

- td:

- '@class': lesson-format

- Дистанционный

- tr:

- th:

'@class': day

- td:

'@class': time

span: 10:00-11:30

div: 2, 4, 6, 8, 10, 12, 14, 16

dd:

- '@class': rasp\_aud\_mobile

- 1124 ауд.

dt:

'@class': rasp\_corp\_mobile

i:

'@class': fa fa-map-marker

span: ул.Ломоносова, д.9, лит. М

- td:

'@class': room

dl:

dd: 1124 ауд.

dt:

i:

'@class': fa fa-map-marker

span: ул.Ломоносова, д.9, лит. М

- td:

'@class': lesson

dl:

- dd: Программирование(Лек)

- четная неделя

- dt:

b: "\n \

\ Письмак Алексей Евгеньевич \

\ "

- td:

- '@class': lesson-format

- Очно - дистанционный

- tr: ''

Программа созданная с библиотекой, создаёт списки добавляя символ ‘-’ к тегам, а к атрибутам представленным в XML документе добавляет символ ‘@’. В остальном нет никаких заметных изменений.

# Дополнительное задание №2

1. Переписать исходный код, добавив в него использование регулярных выражений.
2. Сравнить полученные результаты и объяснить их сходство/различие.

## Исходный код

import re

def add\_header\_xml(html):

xmlText = """<?xml version="1.0" encoding="UTF-8" ?>\n""" + html

return xmlText

def get\_tag\_text(text):

pat = r"<(.\*?)>"

ret = re.search(pat, text)

if ret is None:

return None

ret = ret.group(0)

ret = ret[1:len(ret) - 1]

return ret

def get\_between\_tags(text):

pat = r"(.\*?)<"

ret = re.match(pat, text)

ret = ret.group(0)

ret = ret[:len(ret) - 1]

return ret

def replace\_w\_spaces(text):

if ' ' in text:

x = text.split('"')

x[1] = x[1].replace(' ', 'wEoN')

united = ''

for i in x:

if i == x[-1]:

return united

united += i + '"'

return united

else:

return text

def replace\_weon(text):

x = text.replace('wEoN', ' ')

return x

def xml\_to\_yaml(xmlinput: str):

yamlText = "---\n"

indentations = 0

temp = xmlinput.removeprefix("""<?xml version="1.0" encoding="UTF-8" ?>""")

while temp is not None:

between = get\_tag\_text(temp)

if between is None:

yamlText += "\n..."

yamlText = replace\_weon(yamlText)

return yamlText

to\_remove = "<" + between + ">"

end = temp.find(to\_remove) + len(to\_remove)

to\_remove = temp[0:end]

temp = temp.removeprefix(to\_remove)

if '"' in between:

between = replace\_w\_spaces(between)

between = between.split(" ")

comparador = []

for i in between:

if "/" in between[0]:

indentations -= 1

elif i == between[0]:

yamlText += (indentations \* " ") + i + ":\n"

elif len(between) > 1:

if comparador != between:

indentations += 1

i = i.split("=")

yamlText += (indentations \* " ") + i[0] + ": " + i[1] + "\n"

comparador = between

if (i == between[-1] or len(between) == 1) and not "/" in between[0]:

indentations += 1

temp = temp.lstrip()

if temp is not None and temp != '':

if temp[0] != '<':

data = get\_between\_tags(temp)

to\_remove = data

temp = temp.removeprefix(to\_remove)

data = data.lstrip()

if data != '':

yamlText += (indentations \* " ") + '"' + data + '"' + "\n"

yamlText += "\n..."

yamlText = replace\_weon(yamlText)

return yamlText

def second(text):

x = add\_header\_xml(text)

y = xml\_to\_yaml(x)

outputFile = open("2.yml", "w")

outputFile.write(y)

## Вывод программы

---

table:

id: "2day"

class: "rasp\_tabl"

cellspacing: "0"

cellpadding: "0"

border: "0"

tbody:

tr:

th:

class: "day"

script:

"$("#2day\_btn").show();"

span:

"Вт"

td:

class: "time"

span:

"10:00-11:30"

div:

"3, 7, 9, 11, 13, 15, 17"

dd:

class: "rasp\_aud\_mobile"

dt:

class: "rasp\_corp\_mobile"

i:

class: "fa fa-map-marker"

span:

"ул.Ломоносова, д.9, лит. М"

td:

class: "room"

dl:

dd:

dt:

i:

class: "fa fa-map-marker"

span:

"ул.Ломоносова, д.9, лит. М"

td:

class: "lesson"

dl:

dd:

"Основы профессиональной деятельности(Лек): актовый зал"

dt:

b:

"Клименков Сергей Викторович "

td:

class: "lesson-format"

"Очно - дистанционный "

tr:

th:

class: "day"

td:

class: "time"

span:

"10:00-11:30"

div:

"5"

dd:

class: "rasp\_aud\_mobile"

dt:

class: "rasp\_corp\_mobile"

i:

class: "fa fa-map-marker"

span:

td:

class: "room"

dl:

dd:

dt:

i:

class: "fa fa-map-marker"

span:

td:

class: "lesson"

dl:

dd:

"Основы профессиональной деятельности(Лек): ZOOM"

dt:

b:

"Клименков Сергей Викторович "

td:

class: "lesson-format"

"Дистанционный "

tr:

th:

class: "day"

td:

class: "time"

span:

"10:00-11:30"

div:

"2, 4, 6, 8, 10, 12, 14, 16"

dd:

class: "rasp\_aud\_mobile"

"1124 ауд."

dt:

class: "rasp\_corp\_mobile"

i:

class: "fa fa-map-marker"

span:

"ул.Ломоносова, д.9, лит. М"

td:

class: "room"

dl:

dd:

"1124 ауд."

dt:

i:

class: "fa fa-map-marker"

span:

"ул.Ломоносова, д.9, лит. М"

td:

class: "lesson"

dl:

dd:

"Программирование(Лек)"

"четная неделя"

dt:

b:

"Письмак Алексей Евгеньевич "

td:

class: "lesson-format"

"Очно - дистанционный "

tr:

...

По сравнению с изначальной программой, текст YAML не имеет никаких отличии так как замены в обеих программах были сделаны с одинаковой логикой.

# Дополнительное задание №3

1. Используя свою исходную программу из обязательного задания, программу из дополнительного задания №1 и программу из дополнительного задания №2, сравнить десятикратное время выполнения парсинга + конвертации в цикле.
2. Проанализировать полученные результаты и объяснить их сходство/различие.

## Исходный код

import time

from oblig import \*

from first import \*

from second import \*

x = """<table id="2day" class="rasp\_tabl" cellspacing="0" cellpadding="0" border="0">

<tbody><tr><th class="day">

<script>$("#2day\_btn").show();</script>

<span>Вт</span></th> <td class="time">

<span>10:00-11:30</span>

<div>3, 7, 9, 11, 13, 15, 17</div>

<dd class="rasp\_aud\_mobile"></dd>

<dt class="rasp\_corp\_mobile"><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</td>

<td class="room">

<dl>

<dd></dd>

<dt><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</dl>

</td>

<td class="lesson">

<dl>

<dd>Основы профессиональной деятельности(Лек): актовый зал</dd><dt><b>

Клименков Сергей Викторович </b></dt>

</dl>

</td><td class="lesson-format">Очно - дистанционный </td>

</tr>

<tr><th class="day"></th> <td class="time">

<span>10:00-11:30</span>

<div>5</div>

<dd class="rasp\_aud\_mobile"></dd>

<dt class="rasp\_corp\_mobile"><i class="fa fa-map-marker"></i><span></span></dt>

</td>

<td class="room">

<dl>

<dd></dd>

<dt><i class="fa fa-map-marker"></i><span></span></dt>

</dl>

</td>

<td class="lesson">

<dl>

<dd>Основы профессиональной деятельности(Лек): ZOOM</dd><dt><b>

Клименков Сергей Викторович </b></dt>

</dl>

</td><td class="lesson-format">Дистанционный </td>

</tr>

<tr><th class="day"></th> <td class="time">

<span>10:00-11:30</span>

<div>2, 4, 6, 8, 10, 12, 14, 16</div>

<dd class="rasp\_aud\_mobile">1124 ауд.</dd>

<dt class="rasp\_corp\_mobile"><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</td>

<td class="room">

<dl>

<dd>1124 ауд.</dd>

<dt><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</dl>

</td>

<td class="lesson">

<dl>

<dd>Программирование(Лек)</dd>четная неделя<dt><b>

Письмак Алексей Евгеньевич </b></dt>

</dl>

</td><td class="lesson-format">Очно - дистанционный </td>

</tr>

<tr> """

start\_time = time.time()

for i in range(1000):

oblig(x)

i += 1

print("Oblig: --- %s seconds ---" % (time.time() - start\_time))

start\_time = time.time()

for i in range(1000):

first(x)

i += 1

print("First: --- %s seconds ---" % (time.time() - start\_time))

start\_time = time.time()

for i in range(1000):

second(x)

i += 1

print("Second: --- %s seconds ---" % (time.time() - start\_time))

## Вывод программы

Oblig: --- 0.6864142417907715 seconds ---

First: --- 7.106864929199219 seconds ---

Second: --- 0.7123725414276123 seconds ---

Программа без библиотек и программа, использующая регулярные выражения выполняют работу в 10 раз быстрее чем программа с библиотекой xmlplain. Главная причина такой разницы в скорости работы программы с библиотекой xmlplain в том, что данная библиотека сначала сохраняет данные, полученные из документа xml как списки, словари и другие структуры данных и только потом сохраняет эти данные в документ yaml. Между программой без библиотек и программой с regex есть незначительная разница по времени, и при повторном выполнении программ, программа с regex может показать лучшую скорость выполнения.

# Дополнительное задание №4

1. Переписать исходную, 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. Проанализировать полученные результаты, объяснить особенности использованного формата.

## Исходный код

def get\_tag\_text(text):

for i in range(len(text)):

if text[i] == "<":

start = i + 1

end = text.find(">", start, )

ret = text[i + 1:end]

return ret

return None

def get\_between\_tags(text):

end = text.find("<", )

ret = text[:end]

return ret

x = """<tr><th class="day">

<script>$("#2day\_btn").show();</script>

<span>Вт</span></th> <td class="time">

<span>10:00-11:30</span>

<div>3, 7, 9, 11, 13, 15, 17</div>

<dd class="rasp\_aud\_mobile"></dd>

<dt class="rasp\_corp\_mobile"><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</td>

<td class="room">

<dl>

<dd></dd>

<dt><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</dl>

</td>

<td class="lesson">

<dl>

<dd>Основы профессиональной деятельности(Лек): актовый зал</dd><dt><b>

Клименков Сергей Викторович </b></dt>

</dl>

</td><td class="lesson-format">Очно - дистанционный </td>

</tr>

<tr><th class="day"></th> <td class="time">

<span>10:00-11:30</span>

<div>5</div>

<dd class="rasp\_aud\_mobile"></dd>

<dt class="rasp\_corp\_mobile"><i class="fa fa-map-marker"></i><span></span></dt>

</td>

<td class="room">

<dl>

<dd></dd>

<dt><i class="fa fa-map-marker"></i><span></span></dt>

</dl>

</td>

<td class="lesson">

<dl>

<dd>Основы профессиональной деятельности(Лек): ZOOM</dd><dt><b>

Клименков Сергей Викторович </b></dt>

</dl>

</td><td class="lesson-format">Дистанционный </td>

</tr>

<tr><th class="day"></th> <td class="time">

<span>10:00-11:30</span>

<div>2, 4, 6, 8, 10, 12, 14, 16</div>

<dd class="rasp\_aud\_mobile">1124 ауд.</dd>

<dt class="rasp\_corp\_mobile"><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</td>

<td class="room">

<dl>

<dd>1124 ауд.</dd>

<dt><i class="fa fa-map-marker"></i><span>ул.Ломоносова, д.9, лит. М</span></dt>

</dl>

</td>

<td class="lesson">

<dl>

<dd>Программирование(Лек)</dd>четная неделя<dt><b>

Письмак Алексей Евгеньевич </b></dt>

</dl>

</td><td class="lesson-format">Очно - дистанционный </td>

</tr>"""

def main(text):

tags = ''

data = ''

temp = text

while temp is not None:

tagText = get\_tag\_text(temp)

if tagText is None:

ret = tags + '\n' + data

return ret

elif '/' in tagText:

dataText = get\_between\_tags(temp)

if dataText is not None or dataText != '':

data += '"' + get\_between\_tags(temp) + '"' + ';'

else:

tags += tagText + ';'

to\_remove = "<" + tagText + ">"

end = temp.find(to\_remove) + len(to\_remove)

to\_remove = temp[0:end]

temp = temp.removeprefix(to\_remove)

ret = tags + '\n' + data

return ret

outputFile = open("4.csv", "w")

outputFile.write(main(x))

## Вывод программы

tr;th class="day";script;span;td class="time";span;div;dd class="rasp\_aud\_mobile";dt class="rasp\_corp\_mobile";i class="fa fa-map-marker";span;td class="room";dl;dd;dt;i class="fa fa-map-marker";span;td class="lesson";dl;dd;dt;b;td class="lesson-format";tr;th class="day";td class="time";span;div;dd class="rasp\_aud\_mobile";dt class="rasp\_corp\_mobile";i class="fa fa-map-marker";span;td class="room";dl;dd;dt;i class="fa fa-map-marker";span;td class="lesson";dl;dd;dt;b;td class="lesson-format";tr;th class="day";td class="time";span;div;dd class="rasp\_aud\_mobile";dt class="rasp\_corp\_mobile";i class="fa fa-map-marker";span;td class="room";dl;dd;dt;i class="fa fa-map-marker";span;td class="lesson";dl;dd;dt;b;td class="lesson-format";

"$("#2day\_btn").show();";"Вт";"";"10:00-11:30";"3, 7, 9, 11, 13, 15, 17";"";"";"ул.Ломоносова, д.9, лит. М";"";"

";"";"";"ул.Ломоносова, д.9, лит. М";"";"

";"

";"Основы профессиональной деятельности(Лек): актовый зал";"

Клименков Сергей Викторович ";"";"

";"

";"Очно - дистанционный ";"

";"";"10:00-11:30";"5";"";"";"";"";"

";"";"";"";"";"

";"

";"Основы профессиональной деятельности(Лек): ZOOM";"

Клименков Сергей Викторович ";"";"

";"

";"Дистанционный ";"

";"";"10:00-11:30";"2, 4, 6, 8, 10, 12, 14, 16";"1124 ауд.";"";"ул.Ломоносова, д.9, лит. М";"";"

";"1124 ауд.";"";"ул.Ломоносова, д.9, лит. М";"";"

";"

";"Программирование(Лек)";"

Письмак Алексей Евгеньевич ";"";"

";"

";"Очно - дистанционный ";"

";

CSV это текстовый формат, предназначенный для представления табличных данных, которые разделены между собой с помощью запятой и где каждая строка файла это одна строка в табличном виде. Данные имеющие зарезервированные символы должны быть заключены в кавычках.

# Вывод

В данной работе были переведены данные из формата XML в разные языки разметки. Языки разметки позволяют организованно сохранять данные и облегчают передачу данных между программами и людьми. Перевод форматов был реализован с помощью Python, и результаты сравнивались с выполнением перевода с регулярными выражениями и с библиотекой xmlplain.

# Список Литературы

*Python Software Foundation* (2022) re — Regular expression operations: https://docs.python.org/3/library/re.html

*Лямин А.В., Череповская Е.Н.* (2017) Объектно-ориентированное программирование. Компьютерный практикум. – СПб: Университет ИТМО. – 143 с.

*Guillon C.* (2018) – Xmlplain: https://guillon.github.io/xmlplain/