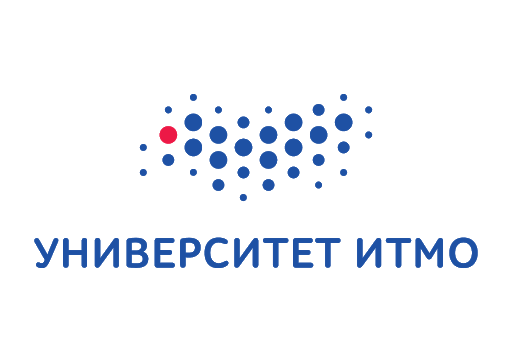
**Федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»  
Факультет программной инженерии и компьютерной техники  
Кафедра вычислительной техники**

**Лабораторная работа: № 3**

**Предмет: Информатика**

**«Исследование языков разметки документов»**

**Вариант: 21**

**Выполнил:**

Студент группы Р3112

*Оскилко М.В.*

**Преподаватель**:

[*Малышева Т. А*](https://isu.ifmo.ru/pls/apex/f?p=2143:3:100583859049304::NO::PID:165275)

**САНКТ-ПЕТЕРБУРГ**

**2020г.**

1. **Задания варианта:  
   YAML -> XML в четверг.**
2. **Исходной код в YAML:**

timetable\_P3112\_for\_thursday:

first\_lesson\_all\_weeks:

day: 'Чт'

time:

time\_of\_lesson: '08:20-09:50'

even\_or\_add\_week: ' '

room:

audience: '466 ауд.'

place: 'Кронверкский пр., д.49, лит.А'

lesson:

what\_lesson: 'Математика (базовый уровень)(Лек)'

teacher: 'Возианова Анна'

lesson\_format: 'Очно - дистанционный'

second\_lesson\_even\_weeks:

day: ' '

time:

time\_of\_lesson: '10:00-11:30'

even\_or\_add\_week: 'четная неделя'

room:

audience: 'АКТОВЫЙ ЗАЛ'

place: 'Кронверкский пр., д.49, лит.А'

lesson:

what-lesson: 'ПРОГРАММИРОВАНИЕ(ЛЕК)'

teacher: 'Письмак Алексей Евгеньевич'

lesson\_format: 'Очно - дистанционный'

second\_lesson\_odd\_weeks:

day: ' '

time:

time\_of\_lesson: '10:00-11:30'

even\_or\_add\_week: 'нечетная неделя'

room:

audience: 'АКТОВЫЙ ЗАЛ'

place: 'Кронверкский пр., д.49, лит.А'

lesson:

what\_lesson: 'ИНФОРМАТИКА(ЛЕК)'

teacher: 'Балакшин Павел Валерьевич'

lesson-format: 'Очно - дистанционный'

third\_lesson\_even\_weeks:

day: ' '

time:

time\_of\_lesson: '11:40-13:10'

even\_or\_add\_week: 'нечетная неделя'

room:

audience: 'АКТОВЫЙ ЗАЛ'

place: 'Кронверкский пр., д.49, лит.А'

lesson:

what\_lesson: 'ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛЕК)'

teacher: 'Клименков Сергей Викторович'

lesson\_format: 'Очно - дистанционный'

1. **Преобразованный код в XML:**

<timetable\_P3112\_for\_thursday>

<first\_lesson\_all\_weeks>

<day>Чт</day>

<time>

<time\_of\_lesson>08:20-09:50</time\_of\_lesson>

<even\_or\_add\_week></even\_or\_add\_week>

</time>

<room>

<audience>466 ауд.</audience>

<place>Кронверкский пр., д.49, лит.А</place>

</room>

<lesson>

<what\_lesson>Математика (базовый уровень)(Лек)</what\_lesson>

<teacher>Возианова Анна</teacher>

</lesson>

<lesson\_format>Очно - дистанционный</lesson\_format>

</first\_lesson\_all\_weeks>

<second\_lesson\_even\_weeks>

<day></day>

<time>

<time\_of\_lesson>10:00-11:30</time\_of\_lesson>

<even\_or\_add\_week>четная неделя</even\_or\_add\_week>

</time>

<room>

<audience>АКТОВЫЙ ЗАЛ</audience>

<place>Кронверкский пр., д.49, лит.А</place>

</room>

<lesson>

<what-lesson>ПРОГРАММИРОВАНИЕ(ЛЕК)</what-lesson>

<teacher>Письмак Алексей Евгеньевич</teacher>

</lesson>

<lesson\_format>Очно - дистанционный</lesson\_format>

</second\_lesson\_even\_weeks>

<second\_lesson\_odd\_weeks>

<day></day>

<time>

<time\_of\_lesson>10:00-11:30</time\_of\_lesson>

<even\_or\_add\_week>нечетная неделя</even\_or\_add\_week>

</time>

<room>

<audience>АКТОВЫЙ ЗАЛ</audience>

<place>Кронверкский пр., д.49, лит.А</place>

</room>

<lesson>

<what\_lesson>ИНФОРМАТИКА(ЛЕК)</what\_lesson>

<teacher>Балакшин Павел Валерьевич</teacher>

</lesson>

<lesson-format>Очно - дистанционный</lesson-format>

</second\_lesson\_odd\_weeks>

<third\_lesson\_even\_weeks>

<day></day>

<time>

<time\_of\_lesson>11:40-13:10</time\_of\_lesson>

<even\_or\_add\_week>нечетная неделя</even\_or\_add\_week>

</time>

<room>

<audience>АКТОВЫЙ ЗАЛ</audience>

<place>Кронверкский пр., д.49, лит.А</place>

</room>

<lesson>

<what\_lesson>ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛЕК)</what\_lesson>

<teacher>Клименков Сергей Викторович</teacher>

</lesson>

<lesson\_format>Очно - дистанционный</lesson\_format>

</third\_lesson\_even\_weeks>

</timetable\_P3112\_for\_thursday>

1. **Код на Python, который преобразует YAML в XML сразу:**

#Считываю данные из yaml файла

file = open('src.yaml', "r", encoding='utf-8')  
string = file.read()  
file.flush()  
  
#Замена : на \ для последующих преобразований + разбиение String на массив строк  
string = string.replace(": ", "\\")  
array = string.split("\n")  
#Содержит тэги, которые в последствии надо будет закрыть по очереди.  
lastbig = []  
#Кол-во " " - пробелов в данной строке  
countx = 0  
#Кол-во " " - пробелов в предыдущей строке  
county = 0  
#Выходной xml код  
addstring = ""  
  
for i in range(0, len(array)):  
 #делит строку на 2 элемента  
 arrs\_of\_strings = array[i].split("\\")  
 county = countx  
 countx = arrs\_of\_strings[0].count(" ")  
 #Убирает ненужные символы.  
 if len(arrs\_of\_strings) > 1:  
 arrs\_of\_strings[1] = arrs\_of\_strings[1].replace("' '", "\\")  
 arrs\_of\_strings[1] = arrs\_of\_strings[1].replace("'", "")  
 #Если значение идущее после “:" было пустым, то удаляет элемент  
 if arrs\_of\_strings[1] == "":  
 del arrs\_of\_strings[1]  
 else:  
 arrs\_of\_strings[0] = arrs\_of\_strings[0].replace(":", "")  
 #Это нужно, чтобы выводить закрывающие элементы, которые имеют дочерние элементы.  
 if county > countx:  
 addstring += " " \* lastbig[len(lastbig)-1].count(" ") + \  
 "</" + lastbig[len(lastbig)-1].replace(" ", "") + ">" + "\n"  
 lastbig.pop()  
 #Выводит элементы, которые не имеют дочерних элементов  
 if len(arrs\_of\_strings) > 1:  
 addstring += " " \* countx + "<" + arrs\_of\_strings[0].replace(" ", "") + ">" \  
 + arrs\_of\_strings[1].replace("\\", "") + "</" + arrs\_of\_strings[0].replace(" ", "") + ">" + "\n"  
 #Программа видит, что появился элемент, которые будет иметь дочерние элементы, запоминает его,  
 #чтобы потом вывести его.  
 else:  
 lastbig.append(arrs\_of\_strings[0])  
 addstring += " " \* lastbig[len(lastbig)-1].count(" ") + \  
 "<" + lastbig[len(lastbig)-1].replace(" ", "") + ">" + "\n"  
  
#Дописывает закрывающие элементы  
while len(lastbig) > 1:  
 addstring += " " \* lastbig[len(lastbig)-1].count(" ") + \  
 "</" + lastbig[len(lastbig)-1].replace(" ", "") + ">" + "\n"  
 lastbig.pop()  
addstring += " " \* lastbig[len(lastbig)-1].count(" ") + \  
 "</" + lastbig[len(lastbig)-1].replace(" ", "") + ">"  
lastbig.pop()  
file\_write = open("bin.xml", "w")  
file\_write.write(addstring)

1. **Код на Python с готовой библиотекой.**

import xmlplain  
  
# Read the YAML file  
with open("src.yaml", "r", encoding='utf-8') as inf:  
 root = xmlplain.obj\_from\_yaml(inf)  
  
# Output back XML  
with open("bin2.xml", "w", encoding='utf-8') as outf:  
 xmlplain.xml\_from\_obj(root, outf, pretty=True)

1. **Результат:**

<?xml version="1.0" encoding="UTF-8"?>

<timetable\_P3112\_for\_thursday>

<first\_lesson\_all\_weeks>

<day>Чт</day>

<time>

<time\_of\_lesson>08:20-09:50</time\_of\_lesson>

<even\_or\_add\_week> </even\_or\_add\_week>

</time>

<room>

<audience>466 ауд.</audience>

<place>Кронверкский пр., д.49, лит.А</place>

</room>

<lesson>

<what\_lesson>Математика (базовый уровень)(Лек)</what\_lesson>

<teacher>Возианова Анна</teacher>

</lesson>

<lesson\_format>Очно - дистанционный</lesson\_format>

</first\_lesson\_all\_weeks>

<second\_lesson\_even\_weeks>

<day> </day>

<time>

<time\_of\_lesson>10:00-11:30</time\_of\_lesson>

<even\_or\_add\_week>четная неделя</even\_or\_add\_week>

</time>

<room>

<audience>АКТОВЫЙ ЗАЛ</audience>

<place>Кронверкский пр., д.49, лит.А</place>

</room>

<lesson>

<what-lesson>ПРОГРАММИРОВАНИЕ(ЛЕК)</what-lesson>

<teacher>Письмак Алексей Евгеньевич</teacher>

</lesson>

<lesson\_format>Очно - дистанционный</lesson\_format>

</second\_lesson\_even\_weeks>

<second\_lesson\_odd\_weeks>

<day> </day>

<time>

<time\_of\_lesson>10:00-11:30</time\_of\_lesson>

<even\_or\_add\_week>нечетная неделя</even\_or\_add\_week>

</time>

<room>

<audience>АКТОВЫЙ ЗАЛ</audience>

<place>Кронверкский пр., д.49, лит.А</place>

</room>

<lesson>

<what\_lesson>ИНФОРМАТИКА(ЛЕК)</what\_lesson>

<teacher>Балакшин Павел Валерьевич</teacher>

</lesson>

<lesson-format>Очно - дистанционный</lesson-format>

</second\_lesson\_odd\_weeks>

<third\_lesson\_even\_weeks>

<day> </day>

<time>

<time\_of\_lesson>11:40-13:10</time\_of\_lesson>

<even\_or\_add\_week>нечетная неделя</even\_or\_add\_week>

</time>

<room>

<audience>АКТОВЫЙ ЗАЛ</audience>

<place>Кронверкский пр., д.49, лит.А</place>

</room>

<lesson>

<what\_lesson>ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛЕК)</what\_lesson>

<teacher>Клименков Сергей Викторович</teacher>

</lesson>

<lesson\_format>Очно - дистанционный</lesson\_format>

</third\_lesson\_even\_weeks>

</timetable\_P3112\_for\_thursday>

1. **Различия:**

После выполнения данных программ можно увидеть, что есть небольшие различия, например, если у меня тэги <day> не содержат в себе ничего, то библиотека оставляет там пробел – “ “. В конце текста есть перенос строки. А также в самом начале есть строка, указывающая, что данный файл является xml, с кодировкой utf-8.

Мой код сразу же преобразует yaml в xml без каких-либо отдельных преобразований. Библиотека же сначала преобразует yaml в строку, похожую на некий объект, который сама создает, а потом уже этот объект преобразует в xml файл.

1. **Сравнение десятикратного времени исполнения:**

Я использовал функцию time() из библиотеки time, которая возвращает время в секундах с начала эпохи.

В начале обоих кодов вставил

import time

#Запоминает время начало кода

start\_time = time.time()

#Потом добавил цикл

For jh int range(10)

#Напечатал время

print(time.time() – start\_time)

Я получил такие результаты:

1. 0.024004220962524414 – для моей собственной программы
2. 0.39308667182922363 – для готовой библиотеки.

Как видно из результатов, готовая библиотека работает намного дольше моей. Это происходит из-за того, что я сразу обрабатываю строку и подаю ее на вывод(добавляю к пременной) и потом сразу создаю новый xml файл. Библиотека же сначала преобразует все в 1 формат, а только потом из этого формата сделает xml формат и выведет его в файл.

1. **Вывод:**

Выполняю данную работу я познакомился с основным синтаксисом языка программирования Python. Как работать со строкой и преобразовывать ее, считывать и создавать файлы. Я однозначно понял, что использовать готовые библиотеки для решения задач намного проще, чем придумывать что-то свое.