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 33.**

Выполнил студент

1-го курса гр. P3131

Мироненко Артем Дмитриевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель

Авксентьева Елена Юрьевна,

Доцент факультета ПИиКТ

Санкт-Петербург

~2023~

# Оглавление

[Задание 3](#_Toc150160469)

[Представление исходного кода программы 4](#_Toc150160470)

[Обязательное задание 6](#_Toc150160471)

[Дополнительное задание 1 10](#_Toc150160472)

[Дополнительное задание 2 10](#_Toc150160473)

[Дополнительное задание 3 12](#_Toc150160474)

[Дополнительное задание 4 14](#_Toc150160475)

[Контрольные вопросы 14](#_Toc150160476)

[Вывод 16](#_Toc150160477)

[Список литературы 16](#_Toc150160478)

# Задание

Понять устройство страницы с расписанием на примере расписания лектора: <https://itmo.ru/ru/schedule/3/125598/raspisanie_zanyatiy.htm>

Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного. При этом необходимо, чтобы в выбранном дне 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.

1. Обязательное задание (позволяет набрать до 45 процентов от максимального числа баллов БаРС за данную лабораторную):

написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла (YAML) 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

YAML 🡪 JSON (суббота)

2. Дополнительное задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). 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.

10.Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). Переписать исходный код, добавив в него использование регулярных выражений.

b) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

3. 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.

4. Дополнительное задание № 4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную). 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.

5. Дополнительное задание № 5 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную). 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.

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

# Представление исходного кода программы

Начальный файл в формате YAML - input\_yaml.yml

Суббота:

Расписание:

Пара1:

Время: 08:20-09:50

Неделя: 3, 7, 9, 11, 13, 15

Группа: PRLCOMP 1

Ауд: 1324 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ / PARALEL COMPUTING (ЛЕК)

Формат занятий: очно-дистанционный

Пара2:

Время: 10:00-11:30

Неделя: 3, 7

Группа: PRLCOMP 1

Ауд: 1324 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ / PARALEL COMPUTING (ЛЕК)

Формат занятий: очно-дистанционный

Пара3:

Время: 10:00-11:30

Неделя: 9, 11, 13

Группа: PRLCOMP 1.1

Ауд: 1324 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ / PARALEL COMPUTING (ЛЕК)

Формат занятий: очно-дистанционный

Пара4:

Время: 10:00-11:30

Неделя: 2, 4, 6, 8, 10, 12, 14, 16, 18

Группа: ИНФОРМ 3.4

Ауд: 2112 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ИНФОРМАТИКА (ЛАБ)

Формат занятий: очно-дистанционный

Пара5:

Время: 11:40-13:10

Неделя: 2, 4, 6, 8, 10, 12, 14, 16, 18

Группа: ИНФОРМ 3.4

Ауд: 2112 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ИНФОРМАТИКА (ЛАБ)

Формат занятий: очно-дистанционный

Пара6:

Время: 13:30-15:00

Неделя: 3, 7, 9, 11, 13

Группа: PRLCOMP 1.1

Ауд: 1324 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ / PARALEL COMPUTING (ЛЕК)

Формат занятий: очно-дистанционный

Пара7:

Время: 10:00-11:30

Неделя: 2, 4, 6, 8, 10, 12, 14, 16, 18

Группа: ИНФОРМ 3.5

Ауд: 2112 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ИНФОРМАТИКА (ЛАБ)

Формат занятий: очно-дистанционный

Пара8:

Время: 11:40-13:10

Неделя: 2, 4, 6, 8, 10, 12, 14, 16, 18

Группа: ИНФОРМ 3.5

Ауд: 2112 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ИНФОРМАТИКА (ЛАБ)

Формат занятий: очно-дистанционный

## Обязательное задание

import time

def spece\_count(str):

return len(str) - len(str.lstrip())

def ymljson(file,out\_f):

# file = r"input\_yaml.yml"

bufer = ""

root = {}

with open(file, 'r', encoding="UTF-8") as f:

bufer = f.readlines()

# print(bufer)

hier = []

for i in bufer:

key, value = i.split(":", 1)

layer, key = spece\_count(key) // 2, key.lstrip()

# print(layer,key)

hier.insert(layer, key) # Добавить эл по индексу layer

# print(hier[-1])

# print(key)

while hier[-1] != key:

hier.pop()

if value == '\n':

current\_tree = root

for node in hier:

if node in current\_tree:

current\_tree = current\_tree[node]

else:

current\_tree[node] = {}

else:

current\_tree = root

for node in hier:

if node in current\_tree:

current\_tree = current\_tree[node]

else:

current\_tree[node] = value.strip()

# print(str(root).replace('\'', '\"'))

a = str(root).replace('\'','\"')

with open(out\_f, 'w', encoding="UTF-8") as file\_out:

file\_out.write(a)

sttime = time.time()

for i in range(100):

ymljson(r"input\_yaml.yml",r"outJson.json")

endtime = time.time()

restime = endtime - sttime

print("Время выполнения, моей программы -", str(restime))

Полученный файл в формате JSON

{

"Суббота":

{

"Расписание":

{

"Пара1":

{

"Время":"08:20-09:50",

"Неделя":"3, 7, 9, 11, 13, 15",

"Группа":"PRLCOMP 1",

"Ауд":"1324 АУД.",

"Корпус":"Кронверский пр., д. 49, лит.А",

"Предмет":"ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ / PARALEL COMPUTING (ЛЕК)",

"Формат занятий":"очно-дистанционный"

},

"Пара2":

{

"Время":"10:00-11:30",

"Неделя":"3, 7",

"Группа":"PRLCOMP 1",

"Ауд":"1324 АУД.",

"Корпус":"Кронверский пр., д. 49, лит.А",

"Предмет":"ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ / PARALEL COMPUTING (ЛЕК)",

"Формат занятий":"очно-дистанционный"

},

"Пара3":

{

"Время":"10:00-11:30",

"Неделя":"9, 11, 13",

"Группа":"PRLCOMP 1.1",

"Ауд":"1324 АУД.",

"Корпус":"Кронверский пр., д. 49, лит.А",

"Предмет":"ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ / PARALEL COMPUTING (ЛЕК)",

"Формат занятий":"очно-дистанционный"

},

"Пара4":

{

"Время":"10:00-11:30",

"Неделя":"2, 4, 6, 8, 10, 12, 14, 16, 18",

"Группа":"ИНФОРМ 3.4",

"Ауд":"2112 АУД.",

"Корпус":"Кронверский пр., д. 49, лит.А",

"Предмет":"ИНФОРМАТИКА (ЛАБ)",

"Формат занятий":"очно-дистанционный"

},

"Пара5":

{

"Время":"11:40-13:10",

"Неделя":"2, 4, 6, 8, 10, 12, 14, 16, 18",

"Группа":"ИНФОРМ 3.4",

"Ауд":"2112 АУД.",

"Корпус":"Кронверский пр., д. 49, лит.А",

"Предмет":"ИНФОРМАТИКА (ЛАБ)",

"Формат занятий":"очно-дистанционный"

},

"Пара6":

{

"Время":"13:30-15:00",

"Неделя":"3, 7, 9, 11, 13",

"Группа":"PRLCOMP 1.1",

"Ауд":"1324 АУД.",

"Корпус":"Кронверский пр., д. 49, лит.А",

"Предмет":"ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ / PARALEL COMPUTING (ЛЕК)",

"Формат занятий":"очно-дистанционный"

},

"Пара7":

{

"Время":"10:00-11:30",

"Неделя":"2, 4, 6, 8, 10, 12, 14, 16, 18",

"Группа":"ИНФОРМ 3.5",

"Ауд":"2112 АУД.",

"Корпус":"Кронверский пр., д. 49, лит.А",

"Предмет":"ИНФОРМАТИКА (ЛАБ)",

"Формат занятий":"очно-дистанционный"

},

"Пара8":

{

"Время":"11:40-13:10",

"Неделя":"2, 4, 6, 8, 10, 12, 14, 16, 18",

"Группа":"ИНФОРМ 3.5",

"Ауд":"2112 АУД.",

"Корпус":"Кронверский пр., д. 49, лит.А",

"Предмет":"ИНФОРМАТИКА (ЛАБ)",

"Формат занятий":"очно-дистанционный"

}

}

}

}

**Исходный код программы:** [**https://github.com/tteemma/inf\_lab4/blob/master/MainTask/main.py**](https://github.com/tteemma/inf_lab4/blob/master/MainTask/main.py)

## Дополнительное задание 1

import yaml

import json

import time

def doptask1():

sttime = time.time()

with open("../input\_yaml.yml", "r", encoding="UTF-8") as yaml\_in, open("../out\_json.json", "w") as json\_out:

yaml\_object = yaml.safe\_load(yaml\_in)

json.dump(yaml\_object,json\_out)

endtime = time.time()

restime = endtime - sttime

print("Время выполнения используя готовые библиотеки - ", restime)

doptask1()

Конвертированные файлы отличаются только расположением фигурных скобок, что на суть контента не влияет.

Код программы: <https://github.com/tteemma/inf_lab4/blob/master/DopTask1/dopTask1.py>

## Дополнительное задание 2

import re

import time

def ymljson(input\_file, out\_f):

with open(input\_file, 'r', encoding='utf8') as inputFile:

data = inputFile.readlines()

pattern\_of\_day = r'Суббота:\n'

day\_repl = r'{\n"Суббота":\n'

lst = [

"Суббота:\n", " Расписание:\n", " Пара1:\n", " Пара2:\n", " Пара3:\n", " Пара4:\n", " Пара5:\n",

" Пара6:\n", " Пара7:\n", " Пара8:\n"

]

with open(out\_f, 'w', encoding='utf8') as out\_f:

for i in range(len(data)):

if data[i] in ["Суббота:\n"]:

rpl = re.sub(pattern\_of\_day, day\_repl, data[i])

out\_f.write(rpl)

elif data[i] in [" Расписание:\n"]:

rpl = re.sub(" Расписание:\n", '\t{\n\t"Расписание":\n\t\t{\n', data[i])

out\_f.write(rpl)

elif data[i] in [" Пара1:\n", " Пара2:\n", " Пара3:\n", " Пара4:\n", " Пара5:\n", " Пара6:\n",

" Пара7:\n", " Пара8:\n"]:

brace\_str = data[i].lstrip().split(':', maxsplit=1)

out\_f.write('\t\t"' + brace\_str[0] + '":' + brace\_str[1])

out\_f.write('\t\t\t{\n')

else:

if i + 1 == len(data):

brace\_str = data[i].lstrip().split(':', maxsplit=1)

a = brace\_str[1].split("\n")

out\_f.write('\t\t\t\t"' + brace\_str[0] + '":' + a[0].lstrip() + "\n")

elif i + 1 != len(data) and (data[i + 1] in lst):

brace\_str = data[i].lstrip().split(':', maxsplit=1)

a = brace\_str[1].split("\n")

out\_f.write('\t\t\t\t"' + brace\_str[0] + '":' + a[0].lstrip() + "\n")

else:

brace\_str = data[i].lstrip().split(':', maxsplit=1)

a = brace\_str[1].split("\n")

out\_f.write('\t\t\t\t"' + brace\_str[0] + '":' + a[0].lstrip() + ",\n")

if i + 1 != len(data) and data[i + 1] in lst:

out\_f.write('\t\t\t},\n')

out\_f.write("\t\t\t}\n\t\t}\n\t}\n}"'\n')

yamlFile = r"input\_yaml.yml"

jsonOutFile = r"output\_json.json"

st = time.time()

for i in range(100):

ymljson(yamlFile, jsonOutFile)

et = time.time()

tm = et - st

print("Время выполнения, с помощью регулярных выражений - " + str(tm))

Исходный код программы: <https://github.com/tteemma/inf_lab4/blob/master/DopTask2/dopTask2.py>

## Дополнительное задание 3

Для проверки того, что мой код работает и на других YAML файлах, я создал еще 2 файла с расписанием моего преподавателя по ОПД.

input\_yaml\_forTask3(1).yml:

Среда:

Расписание:

Пара1:

Время: 15:20-16:50

Неделя: четная неделя

Группа: ОснПрофДеят 1.1

Ауд: 1328 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ (ЛАБ)

Формат занятий: очно-дистанционный

Пара2:

Время: 17:00-18:30

Неделя: четная неделя

Группа: ОснПрофДеят 1.1

Ауд: 1328 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ (ЛАБ)

Формат занятий: очно-дистанционный

input\_yaml\_forTask3(2).yml:

Понедельник:

Расписание:

Понедельник:

Пара1:

Время: 17:00-18:30

Неделя: четная неделя

Группа: ОснПрофДеят 1.21

Ауд: 1338 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ (ЛАБ)

Формат занятий: очно

Пара2:

Время: 18:40-20:10

Неделя: четная неделя

Группа: ОснПрофДеят 1.21

Ауд: 1338 АУД.

Корпус: Кронверский пр., д. 49, лит.А

Предмет: ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ (ЛАБ)

Формат занятий: очно

Исходный код программы: <https://github.com/tteemma/inf_lab4/blob/master/MainTask/main.py>

## Дополнительное задание 4

Время стократного выполнения программы из обязательного задания – 0.023990392684936523 секунды

Время стократного выполнения программы из дополнительного задания №1 – 0.008080720901489258 секунды

Время стократного выполнения программы из дополнительного задания №2 – 0.032018423080444336

Заметим, что программы из обязательного задания и из доп. задания №2 выполняются с почти одинаковой скоростью, т.к. реализация конвертации в них отличается минимально.

А программа, использующая сторонние библиотеки работает на порядок быстрее. Скорее всего это связано с тем, что программа из обязательного задания не является примером хорошего кода. А используя библиотеки я получаю наилучший результат, ведь их делали знающие люди, как раз для того, чтобы уменьшить время выполнения программ.

# Контрольные вопросы

1. В чём разница между Markup и Markdown?

Markup (разметка): это общий термин, который включает в себя различные языки разметки, такие как HTML, XML, LaTeX и другие. Markup является более общим понятием и включает в себя разные языки с разными синтаксисами.

Markdown: это простой, легковесный язык разметки, который создан для простоты чтения и написания. Он использует простой синтаксис и часто преобразуется в HTML или другие форматы разметки.

1. В чём заключается особенность PROTOBUF по сравнению с другими форматами?

Поскольку структурированные данные хранятся в двоичном формате, они гораздо меньше текстовых форматов, таких как XML или JSON, что позволяет быстрее передавать их по сети.

Protobuf разработан так, чтобы его было легко расширять, что делает его идеальным для работы с быстро меняющимися структурами данных и новыми возможностями.

1. Чем формат CSV отличается от формата TSV?

В csv файлах данные разделены запятыми, а в TSV – табуляциями.

1. Чем обусловлено постоянное появление новых форматов представления данных?
2. Каким образом в формате XML представляются символы ‘>’ и ‘<’?

“<” (знак "меньше чем") представляется с помощью сущности “&lt;”.

“>” (знак "больше чем") представляется сущностью “&gt;”.

1. Что такое сериализация данных?

Сериализация — процесс перевода структур данных или состояния объекта в формат, который может быть сохранён или передан и реконструирован позже.

1. Каким образом в YAML обозначаются комментарии?

# comment text

1. Пояснить, как в языке разметки Markdown создать заголовки разных уровней, оформить код, вывести полужирный, курсивный и зачеркнутый текст?

Заголовки разных уровней оформляются с помощью решеток #, их количество определяет уровень заголовка.

Код оформляется внутри обратных апострофов или внутри тройных обратных апострофов.

Полужирный текст создается с помощью двойных звездочек \*\*

Курсивный – одинарные звездочки \*

Зачеркнутый – с помощью двух тильд ~~

1. Какие форматы обмена данных используются в современных

популярных мессенджерах (Viber, WhatsApp, Telegram и т.д.)?

WhatsApp, Telegram используют собственные протоколы (MTProto – Telegram), Viber – TSL, SRTP.

1. Как расшифровывается аббревиатура SVG?

Scalable Vector Graphics

1. Привести пример использования в языке HTML тега, который

создаёт гиперссылку на url.

<a href="https://www.example.com">

1. Какое число структур может представлять собой в закодированном виде JSON-текст?

6 структур (объект, массив, числа, строки, булевы значения, значения null)

# Вывод

Во время работы выполнения лабораторной работы я узнал о языках разметки YAML,XML, JSON, научился с ними работать и переводить один в другой с помощью средств языка Python. Научился работать с некоторыми библиотеками для парсинга в Python, такими как json, PyYAML.

# Список литературы

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. - экз. - Режим доступа:
2. <https://codebeautify.org/jsonvalidator>
3. <https://www.yamllint.com/>
4. Грошев А.С. Г89 Информатика: Учебник для вузов / А.С. Грошев. – Архангельск, Арханг. гос. техн. ун-т, 2010. -470с. -Режим доступа <https://narfu.ru/university/library/books/0690.pdf>