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

Чураков Александр Алексеевич

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

Преподаватель

Авксентьева Елена Юрьевна,

Доцент факультета ПИиКТ

Санкт-Петербург 2023

Оглавление

[Задание 3](#_Toc150160228)

[Основные этапы вычисления (листинг программы) 3](#_Toc150160229)

[Начальный файл yaml – init\_yaml.yml 3](#_Toc150160230)

[Обязательное задание 3](#_Toc150160231)

[Дополнительное задание №1 3](#_Toc150160232)

[Дополнительное задание №2 3](#_Toc150160233)

[Дополнительное задание №3 3](#_Toc150160234)

[Дополнительное задание №4 4](#_Toc150160235)

[Контрольные вопросы 5](#_Toc150160236)

[Заключение 7](#_Toc150160237)

[Список использованной литературы 7](#_Toc150160238)

# 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.

# Основные этапы вычисления (листинг программы)

## Начальный файл yaml – init\_yaml.yml

Суббота:  
 Расписание:  
 Пара1:  
 Время: "08:20-09:50"  
 Неделя: "3, 7, 9, 11, 13, 15"  
 Группа: "PRLCOMP 1"  
 Ауд./Корпус: "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./Корпус: "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./Корпус: "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./Корпус: "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./Корпус: "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./Корпус: "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./Корпус: "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./Корпус: "2112 АУД."  
 Корпус: "Кронверский пр., д. 49, лит.А"  
 Предмет: "ИНФОРМАТИКА (ЛАБ)"  
 Формат занятий: "очно-дистанционный"

Файл out\_json.json (изначально пустой)

## Обязательное задание

Исходный код - <https://github.com/awesoma31/ITMO_Labs/blob/main/INF/lab4/tasks/main.py>

import time  
  
  
def yaml2json(input\_file, output\_file):  
 with open(input\_file, 'r', encoding='utf8') as in\_file:  
 data = in\_file.readlines()  
 numb\_lines = len(data)  
  
 out\_file = open(output\_file, 'w', encoding='windows-1251')  
 out\_file.write("{\n")  
  
 lst = [  
 "Суббота:\n", " Расписание:\n", " Пара1:\n", " Пара2:\n", " Пара3:\n", " Пара4:\n", " Пара5:\n",  
 " Пара6:\n", " Пара7:\n", " Пара8:\n"  
 ]  
  
 for i in range(0, numb\_lines):  
 if data[i] in ["Суббота:\n"]:  
 if data[0] == data[i]:  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 out\_file.write('"' + sup\_string[0] + '":' + sup\_string[1])  
 else:  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 out\_file.write(' }\n }\n\n"' + sup\_string[0] + '":' + sup\_string[1])  
  
 elif data[i] in [" Расписание:\n"]:  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 out\_file.write('\t{\n')  
 out\_file.write('\t"' + sup\_string[0] + '":' + sup\_string[1])  
 out\_file.write('\t\t{\n')  
  
 elif data[i] in [" Пара1:\n", " Пара2:\n", " Пара3:\n", " Пара4:\n", " Пара5:\n", " Пара6:\n",  
 " Пара7:\n", " Пара8:\n"]:  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 out\_file.write('\t\t"' + sup\_string[0] + '":' + sup\_string[1])  
 out\_file.write('\t\t\t{\n')  
  
 else:  
 if i + 1 == numb\_lines:  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 a = sup\_string[1].split("\n")  
 out\_file.write('\t\t\t\t"' + sup\_string[0] + '":' + a[0].lstrip() + "\n")  
 elif i + 1 != numb\_lines and (data[i + 1] in lst):  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 a = sup\_string[1].split("\n")  
 out\_file.write('\t\t\t\t"' + sup\_string[0] + '":' + a[0].lstrip() + "\n")  
 else:  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 a = sup\_string[1].split("\n")  
 out\_file.write('\t\t\t\t"' + sup\_string[0] + '":' + a[0].lstrip() + ",\n")  
 if i + 1 != numb\_lines and data[i + 1] in lst:  
 out\_file.write('\t\t\t},\n')  
  
 out\_file.write("\t\t\t}\n\t\t}\n\t}\n}"'\n')  
 out\_file.close()  
  
  
IN\_YAML = r"C:\Users\gwert\Documents\ITMO\_Labs\INF\lab4\tasks\data\init\_yaml.yml"  
OUT\_JSON = r"C:\Users\gwert\Documents\ITMO\_Labs\INF\lab4\tasks\data\end\_json.json"  
  
st = time.time()  
for i in range(100):  
 yaml2json(IN\_YAML, OUT\_JSON)  
et = time.time()  
ext = et - st  
print("Время выполнения, используя свой парсер - " + str(ext))

Пример полученного JSON файла представлен на Рисунке 1

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок

## Дополнительное задание №1

Исходный код - <https://github.com/awesoma31/ITMO_Labs/blob/main/INF/lab4/tasks/yaml2json_lib.py>

import json  
import time  
import yaml  
  
IN\_YAML = r"C:\Users\gwert\Documents\ITMO\_Labs\INF\lab4\tasks\data\init\_yaml.yml"  
OUT\_JSON\_LIB = r"C:\Users\gwert\Documents\ITMO\_Labs\INF\lab4\tasks\data\lib\_json.json"  
  
st = time.time()  
  
for i in range(100):  
 with open(IN\_YAML, 'r', encoding='utf8') as yaml\_in, open(OUT\_JSON\_LIB, "w") as json\_out:  
 yaml\_object = yaml.safe\_load(yaml\_in)  
 json.dump(yaml\_object, json\_out)  
  
et = time.time()  
ext = et - st  
  
print("Время выполнения, используя библиотеки - " + str(ext))

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

## Дополнительное задание №2

Исходный код - <https://github.com/awesoma31/ITMO_Labs/blob/main/INF/lab4/tasks/yaml2json_regexp.py>

import re  
import time  
  
  
def yml2json\_regexp(input\_file, out\_f):  
 with open(input\_file, 'r', encoding='utf8') as in\_file:  
 data = in\_file.readlines()  
  
 day\_pattern = 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='windows-1251') as out\_f:  
 for i in range(len(data)):  
 if data[i] in ["Суббота:\n"]:  
 rpl = re.sub(day\_pattern, 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"]:  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 out\_f.write('\t\t"' + sup\_string[0] + '":' + sup\_string[1])  
 out\_f.write('\t\t\t{\n')  
  
 else:  
 if i + 1 == len(data):  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 a = sup\_string[1].split("\n")  
 out\_f.write('\t\t\t\t"' + sup\_string[0] + '":' + a[0].lstrip() + "\n")  
 elif i + 1 != len(data) and (data[i + 1] in lst):  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 a = sup\_string[1].split("\n")  
 out\_f.write('\t\t\t\t"' + sup\_string[0] + '":' + a[0].lstrip() + "\n")  
 else:  
 sup\_string = data[i].lstrip().split(':', maxsplit=1)  
 a = sup\_string[1].split("\n")  
 out\_f.write('\t\t\t\t"' + sup\_string[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')  
  
  
IN\_YAML = r"data\init\_yaml.yml"  
OUT\_JSON\_REGEXP = r"data\regexp.json"  
  
st = time.time()  
for i in range(100):  
 yml2json\_regexp(IN\_YAML, OUT\_JSON\_REGEXP)  
  
et = time.time()  
tm = et - st  
  
print("Время выполнения, используя регулярки - " + str(tm))

Результат не отличается от полученного в обязательном задании файла, потому что при написании кода я стремился к получению такого же результата.

## Дополнительное задание №3

import time  
  
  
def count\_spaces(row):  
 return len(row) - len(row.lstrip())  
  
  
def my\_yaml2json(in\_file, out\_f):  
 res = {}  
 with open(in\_file, 'r', encoding="utf8") as f:  
 buffer = f.readlines()  
  
 hierarchy = []  
  
 for i in buffer:  
 key, value = i.split(":", 1)  
 layer, key = count\_spaces(key) // 2, key.lstrip()  
 hierarchy.insert(layer, key)  
  
 while hierarchy[-1] != key:  
 hierarchy.pop()  
 if value == '\n':  
 cur\_tree = res  
 for j in hierarchy:  
 if j in cur\_tree:  
 cur\_tree = cur\_tree[j]  
 else:  
 cur\_tree[j] = {}  
 else:  
 cur\_tree = res  
 for j in hierarchy:  
 if j in cur\_tree:  
 cur\_tree = cur\_tree[j]  
 else:  
 cur\_tree[j] = value.strip()  
  
 a = str(res).replace('\'', '\"')  
 with open(out\_f, 'w', encoding='utf8') as out\_f:  
 out\_f.write(a)  
  
  
test1file = r"data\test1.yml"  
test2file = r"data\test2.yml"  
OUT\_FILE = r"data\dop3.yml"  
  
st = time.time()  
  
for i in range(100):  
 my\_yaml2json(test1file, OUT\_FILE)  
  
et = time.time()  
ext = et - st  
print("Время выполнения доп. задания 3 - ", str(ext))

В результирующем файле данные будут представлены в виде 1 строки, но это также возможно, т. к. в json уровни разделяются не отступами, а скобками.

## Дополнительное задание №4

Время стократного выполнения программы из обязательного задания – 0.046010494232177734 секунды

Время стократного выполнения программы из дополнительного задания №1 – 0.461458683013916 секунды

Время стократного выполнения программы из дополнительного задания №2 – 0.048011064529418945

Время стократного выполнения программы из дополнительного задания №2 – 0.1125345230102539

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 медленнее. Скорее всего это связано с тем, что программа из обязательного задания не является универсальным парсером и более поверхностно обрабатывает исходный файл.

# 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?

Т. к. в разных областях применение может быть удобнее использовать более уникальные форматы, оптимизированные под конкретную задачу.

1. 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)

# Заключение

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

# Список использованной 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:

<https://books.ifmo.ru/book/2248/informatika:_laboratornye_raboty_i_testy:_uchebno-metodicheskoe_posobie_/_recenzent:_polyakov_v.i..htm>

1. 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>