Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Отчет**

По лабораторной работа № 4 по дисциплине “Информатика”

Исследование протоколов,

форматов обмена информацией и языков разметки

документов

Вариант № 9

Студент:

Мирзаитов Тимур

Группа P3112

Преподаватель:

Рыбаков Степан Дмитриевич

Санкт-Петербург 2023

Оглавление

[Задание 1](#_Toc152031776)

[Основные этапы вычисления 2](#_Toc152031777)

# Задание

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 в новый путём простых замены метасимволов исходного формата на метасимволы результирующего формата.
2. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.
3. Дополнительное задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
   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. Объяснение должно быть отражено в отчёте.
4. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. Переписать исходный код, добавив в него использование регулярных выражений.
   2. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
5. Дополнительное задание №3 (позволяет набрать +25 процентов от максимального числа баллов БаРС за данную лабораторную).
6. 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.
7. b) Проверку осуществить как минимум для расписания с двумя учебными днями по два занятия в каждом.
8. с) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
9. Дополнительное задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле.
   2. Проанализировать полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
10. Дополнительное задание №5 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).
    1. 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.
    2. Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте.
11. Проверить, что все пункты задания выполнены и выполнены верно.
12. Написать отчёт о проделанной работе.
13. Подготовиться к устным вопросам на защите.

# Основные этапы вычисления

## main.py

### Листинг

**import** **time**

**for** xcvbn **in** range(**100**):

source\_file = open('D:\ИТМО 23-24\Информатика\Практика\Лаба №4\source.yml', 'r')

out\_file\_ = open('D:\ИТМО 23-24\Информатика\Практика\Лаба №4\out.json', 'w+')

out\_file = []

out\_file.append('{**\n**')

k = **1**

inBlock = **False**

**def** **outBlock**():

**global** k

**global** inBlock

s = (str)(out\_file.pop())

s = s[:-**2**]

out\_file.append(s + '**\n**')

out\_file.append('**\t**'\*(k-**1**) + '}]**\n**')

k-=**1**

inBlock = **False**

**for** line **in** source\_file:

line = line.strip('**\n**| ',)

**if**(line[-**1**] == ':'):

**if**(inBlock == **True**):

outBlock()

s = (str)(out\_file.pop())

s = s[:-**1**]

s += ',**\n**'

out\_file.append(s)

out\_file.append('**\t**'\*k + '"' + line.split(':')[**0**].replace(' ', '') + '"' + ':' +'**\n**')

out\_file.append('**\t**'\*k +'{**\n**')

k+=**1**

**if**(inBlock == **True** **and** line[**0**] != '-'):

out\_file.append('**\t**'\*k + '"' + line.split(':', **1**)[**0**] + '"' + ': ' + '"' + line.split(':', **1**)[**1**].replace('"', '').lstrip() + '",' + '**\n**')

**if**(line[**0**] == '-' **and** inBlock == **False**):

out\_file.pop()

k-=**1**

out\_file.append('**\t**'\*k + '[{**\n**')

inBlock = **True**

k+=**1**

**elif**(inBlock == **True** **and** line[**0**] == '-'):

s = (str)(out\_file.pop())

s = s[:-**2**]

out\_file.append(s + '**\n**')

out\_file.append('**\t**'\*(k-**1**) + '},' + '**\n**')

out\_file.append('**\t**'\*(k-**1**) + '{' + '**\n**')

outBlock()

**while** k>**0**:

out\_file.append('**\t**'\*(k-**1**) + '}**\n**')

k-=**1**

#print(out\_file)

out\_file\_.writelines(out\_file)

end = time.perf\_counter()

print(time.process\_time())

### Описание

Для решения задачи я ищу некоторые шаблоны синтаксиса yaml и сразу перевожу их в json добавляя в массив, который в конце добавляется в файл .json

## main\_regex.py

### Листинг

**import** **re**

**import** **time**

**for** i **in** range(**100**):

source\_file = open('D:\ИТМО 23-24\Информатика\Практика\Лаба №4\source.yml', 'r')

out\_file\_ = open('D:\ИТМО 23-24\Информатика\Практика\Лаба №4\out\_regex.json', 'w+')

out\_file = []

out\_file.append('{**\n**')

k = **1**

inBlock = **False**

**with** open(r"D:\ИТМО 23-24\Информатика\Практика\Лаба №4\source.yml", 'r') **as** fp:

**for** sizeOfFile, line **in** enumerate(fp):

**pass**

**for** i, line **in** enumerate(source\_file):

pattern = r"((.+): (.+))|((.+):)|(-)"

line = line.strip('**\n**| ')

line = line.replace('"', '')

new\_line = re.findall(pattern, line)

new\_line = list(new\_line[**0**])

#print(new\_line)

**if**(new\_line[-**1**] == ''):

new\_line.pop()

**if**(new\_line[-**1**] == ''):

new\_line.pop()

new\_line.pop()

new\_line.pop(**0**)

**if**(new\_line[**1**].isdigit()):

out\_file.append('**\t**'\*k + '"' + new\_line[**0**] + '": ' + new\_line[**1**] + ',**\n**')

**else**:

out\_file.append('**\t**'\*k + '"' + new\_line[**0**] + '": "' + new\_line[**1**] + '",**\n**')

**else**:

new\_line.pop(**0**)

new\_line.pop(**0**)

new\_line.pop(**0**)

new\_line.pop(**0**)

#print(new\_line)

out\_file.append('**\t**'\*k + '"' + new\_line[**0**] + '":**\n**')

out\_file.append('**\t**'\*k + '{**\n**')

k+=**1**

**else**:

**if**(inBlock == **False**):

out\_file.pop()

k-=**1**

out\_file.append('**\t**'\*k + '[{**\n**')

k+=**1**

inBlock = **True**

**else**:

s = out\_file.pop()

s = s[:-**2**]

out\_file.append(s + '**\n**')

k-=**1**

out\_file.append('**\t**'\*k + '},**\n**')

out\_file.append('**\t**'\*k + '{**\n**')

k+=**1**

**if**(i == sizeOfFile **and** inBlock):

s = out\_file.pop()

s = s[:-**2**]

out\_file.append(s + '**\n**')

k-=**1**

out\_file.append('**\t**'\*k + '}]**\n**')

**while** k>**0**:

out\_file.append('**\t**'\*(k-**1**) + '}**\n**')

k-=**1**

#print(out\_file)

out\_file\_.writelines(out\_file)

print(time.process\_time())

### Описание

Логика выполнения программы не изменилась, поменялся только способ нахождения шаблонов, теперь для их поиска использую regex, и работаю с найденными группами

## main\_lib.py

### Листинг

**import** **yaml**

**import** **json**

**import** **time**

**for** i **in** range(**100**):

**with** open("source.yml", 'r') **as** yaml\_in, open("out\_lib.json", "w") **as** json\_out:

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

json\_object = json.dumps(yaml\_object, indent=**2**)

json\_out.writelines(json\_object)

print(time.process\_time())

### Описание

Использую готовые библиотеки

## main\_csv.py

### Листинг

**import** **time**

**for** i **in** range(**100**):

source\_file = open('D:\ИТМО 23-24\Информатика\Практика\Лаба №4\source.yml', 'r')

out\_file\_ = open('D:\ИТМО 23-24\Информатика\Практика\Лаба №4\out\_csv.csv', 'w+')

out\_file = ['', '']

flag = **False**

FirstFlag = **True**

**for** line **in** source\_file:

line = line.strip('**\n**| ',)

line = line.replace('"', '')

#print(line)

**if** flag **and** line != '-' **and** FirstFlag:

line = line.split(':', **1**)

out\_file[**0**] += line[**0**].strip() + ', '

out\_file[**1**] += line[**1**].strip() + ', '

**if** flag **and** line != '-' **and** **not** FirstFlag:

line = line.split(':', **1**)

out\_file[**1**] += line[**1**].strip() + ', '

**if**(line == '-' **and** **not** flag):

flag = **True**

**elif**(line == '-' **and** flag):

out\_file[-**2**] = out\_file[-**2**][:-**2**] + '**\n**'

out\_file[-**1**] = out\_file[-**1**][:-**2**] + '**\n**'

FirstFlag = **False**

**elif**(line == '-' **and** flag **and** **not** FirstFlag):

out\_file[-**1**] = out\_file[-**1**][:-**2**] + '**\n**'

#print(out\_file)

out\_file\_.writelines(out\_file)

print(time.process\_time())

### Описание

Я выбрал перевод в csv, сначала выделил те данные которые мне нужно было перенести и затем построчно перенес их

## source.yml

schedule:

tuesday:

-

class: computer science

teacher: Rybakov Stepan Dmitrievich

start-time: "13:30"

end-time: "15:00"

address: Kronverksky pr., d.49, lit.A

room: 2127

-

class: computer science

teacher: Rybakov Stepan Dmitrievich

start-time: "15:20"

end-time: "16:50"

address: Kronverksky pr., d.49, lit.A

room: 2127

## out.json

{

"schedule":

{

"tuesday":

[{

"class": "computer science",

"teacher": "Rybakov Stepan Dmitrievich",

"start-time": "13:30",

"end-time": "15:00",

"address": "Kronverksky pr., d.49, lit.A",

"room": "2127"

},

{

"class": "computer science",

"teacher": "Rybakov Stepan Dmitrievich",

"start-time": "15:20",

"end-time": "16:50",

"address": "Kronverksky pr., d.49, lit.A",

"room": "2127"

}]

}

}

## out\_regex.json

{

"schedule":

{

"tuesday":

[{

"class": "computer science",

"teacher": "Rybakov Stepan Dmitrievich",

"start-time": "13:30",

"end-time": "15:00",

"address": "Kronverksky pr., d.49, lit.A",

"room": **2127**

},

{

"class": "computer science",

"teacher": "Rybakov Stepan Dmitrievich",

"start-time": "15:20",

"end-time": "16:50",

"address": "Kronverksky pr., d.49, lit.A",

"room": **2127**

}]

}

}

## out\_lib.json

{

"schedule": {

"tuesday": [

{

"class": "computer science",

"teacher": "Rybakov Stepan Dmitrievich",

"start-time": "13:30",

"end-time": "15:00",

"address": "Kronverksky pr., d.49, lit.A",

"room": **2127**

},

{

"class": "computer science",

"teacher": "Rybakov Stepan Dmitrievich",

"start-time": "15:20",

"end-time": "16:50",

"address": "Kronverksky pr., d.49, lit.A",

"room": **2127**

}

]

}

}

## out\_csv.csv

class, teacher, start-time, end-time, address, room

computer science, Rybakov Stepan Dmitrievich, 13:30, 15:00, Kronverksky pr., d.49, lit.A, 2127

computer science, Rybakov Stepan Dmitrievich, 15:20, 16:50, Kronverksky pr., d.49, lit.A, 2127,

## Время

main.py – ср. время 0,135416 сек

main\_regex.py – ср. время 0,17708 сек

main\_lib.py – ср. время 0.28031 сек

main\_csv.py – ср. время 0,127604 сек

Самое быстрое выполнение команды получилось при основном решении, самое удобное при этом было при regex решение.

# Заключение

В ходе данной лабораторной работы я узнал подробнее про json, yaml, формальные грамматики, впервые писал парсер.

# Список использованных источников

1. Методические пособия по информатике Балакшин П.В., Соснин В.В., Машина Е.А. Информатика. – СПб: Университет ИТМО, 2020. – 122 с.

<https://isu.ifmo.ru/pls/apex/f?p=2143:0:107301084256427:DWNLD_F:NO::FILE:AFCE39920F4F4287E79C1554D056FE9B>

1. ГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления» // Система стандартов по информации, библиотечному и

издательскому делу. Москва: Стандартинформ.

<https://cs.msu.ru/sites/cmc/files/docs/2021-11gost_7.32-2017.pdf>