Министерство науки и высшего образования Российской Федерации

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 №336765

Группа: P3131

Выполнил: Хайкин О. И.

Проверил:

к.т.н. преподаватель Белозубов А.В.

Санкт-Петербург

2021г

Оглавление

[Задание 3](#_Toc87352711)

[Обязательное задание (позволяет набрать до 65 процентов от максимального числа баллов БаРС за данную лабораторную): 3](#_Toc87352712)

[Дополнительное задание задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную): 3](#_Toc87352713)

[Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную): 3](#_Toc87352714)

[Дополнительное задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную): 3](#_Toc87352715)

[Дополнительное задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную: 3](#_Toc87352716)

[Выполнение работы 5](#_Toc87352717)

[Исходный файл (timetable.json) 5](#_Toc87352718)

[Задание №1 5](#_Toc87352719)

[Получившийся файл 7](#_Toc87352720)

[Доп. задание №1 7](#_Toc87352721)

[Получившийся файл 7](#_Toc87352722)

[Доп. задание №2 8](#_Toc87352723)

[Получившийся файл 10](#_Toc87352724)

[Доп. задание №3 10](#_Toc87352725)

[Доп. задание №4 10](#_Toc87352726)

[Получившийся файл 11](#_Toc87352727)

[Заключение 13](#_Toc87352728)

# Задание

## Обязательное задание (позволяет набрать до 65 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. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.

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

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

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

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

## Дополнительное задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную):

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. Проанализировать полученные результаты и объяснить их сходство/различие.

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

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. Проанализировать полученные результаты, объяснить особенности использованного формата.

# Выполнение работы

## Исходный файл (timetable.json)

{

"timetable": {

"day": "THURSDAY",

"lessons": [

{

"Subject1": {

"time": "13:30-15:00",

"cabinet": "285",

"adress": "Кронверкский пр., д.49, лит.А",

"name": "ДИСКРЕТНАЯ МАТЕМАТИКА(ЛЕК)",

"week": ["нечет"],

"teacher": "Поляков Владимир Иванович",

"format": "Очно-дистанционный"

}

},

{

"Subject2": {

"time": "15:20-16:50",

"cabinet": "302",

"adress": "Кронверкский пр., д.49, лит.А",

"name": "ДИСКРЕТНАЯ МАТЕМАТИКА(ЛЕК)",

"week": ["чет", "нечет"],

"teacher": "Поляков Владимир Иванович",

"format": "Очно-дистанционный"

}

}

]

}

}

## Задание №1

a=open("timetable.json", encoding="utf8")

b=a.read().split("\"")

a.close()

for i in range (len(b)):

b[i]=b[i].strip()

b[i]=b[i].replace("\n", "")

tabcount=-1

arrcount=0

arr\_start=0

c=[]

check=0

obj\_mode=0

arr\_mode=0

for i in range (len(b)):

if obj\_mode==1 and "}," in b[i]:

b[i+1]="- "+b[i+1]

arr\_start=1

if "]" in b[i] and "," in b[i]:

b[i]=""

arr\_mode=0

continue

if "," in b[i]:

b[i]=b[i].replace(",", "")

if ":" in b[i]:

b[i]=b[i].replace(":", ": ")

if "]" in b[i]:

arr\_mode=0

b[i]=b[i].replace("]", "")

if arr\_mode==1 and not b[i].startswith("-") and b[i]!="":

b[i]="- "+b[i]

if "[" in b[i]:

b[i+1]="- "+b[i+1]

arr\_start=1

if i+2<len(b) and "{" in b[i+2]:

obj\_mode=1

if not "{" in b[i]:

arr\_mode=1

b[i]=b[i].replace("[", "")

if "{" in b[i]:

for j in range (b[i].count("{")):

tabcount +=1

b[i]=b[i].replace("{", "")

if "}" in b[i]:

for j in range (b[i].count("}")):

tabcount -=1

b[i]=b[i].replace("}", "")

if i+1<len(b) and b[i+1].startswith(":") or (arr\_mode==1 and not ":" in b[i]) and b[i]!="":

if arr\_start!=1 and obj\_mode!=1:

for j in range (tabcount):

b[i]="\t"+b[i]

else:

arr\_start=0

for j in range (tabcount-1):

b[i]="\t"+b[i]

if i+2<len(b) and b[i+2].startswith(":") or arr\_mode==1 and b[i]!="":

b[i]=b[i]+"\n"

c.append(b[i])

c="".join(c)

print(c)

res=open("timetable.yaml", 'w', encoding="utf8")

res.write(c)

Получившийся файл

timetable:

day: THURSDAY

lessons:

- Subject1:

time: 13: 30-15: 00

cabinet: 285

adress: Кронверкский пр. д.49 лит.А

name: ДИСКРЕТНАЯ МАТЕМАТИКА(ЛЕК)

week:

- нечет

teacher: Поляков Владимир Иванович

format: Очно-дистанционный

- Subject2:

time: 15: 20-16: 50

cabinet: 302

adress: Кронверкский пр. д.49 лит.А

name: ДИСКРЕТНАЯ МАТЕМАТИКА(ЛЕК)

week:

- чет

- нечет

teacher: Поляков Владимир Иванович

format: Очно-дистанционный

## Доп. задание №1

import json

import yaml

a=open("timetable.json", encoding="utf8")

c=yaml.dump(json.load(a), allow\_unicode=True)

print(c)

res=open("timetable.yaml", 'w', encoding="utf8")

res.write(c)

res.close()

Получившийся файл

timetable:

day: THURSDAY

lessons:

- Subject1:

adress: Кронверкский пр., д.49, лит.А

cabinet: '285'

format: Очно-дистанционный

name: ДИСКРЕТНАЯ МАТЕМАТИКА(ЛЕК)

teacher: Поляков Владимир Иванович

time: 13:30-15:00

week:

- нечет

- Subject2:

adress: Кронверкский пр., д.49, лит.А

cabinet: '302'

format: Очно-дистанционный

name: ДИСКРЕТНАЯ МАТЕМАТИКА(ЛЕК)

teacher: Поляков Владимир Иванович

time: 15:20-16:50

week:

- чет

- нечет

Единственное различие этого файла с файлом, получившимся в результате работы программы из задания №1 – положение массивов “week”. Порядок ключей в данном случае не важен, поэтому оба варианта являются верными.

## Доп. задание №2

import re

a=open("timetable.json", encoding="utf8")

c=a.read()

c=re.sub('([\]"}])(,)', '\g<1>\n', c)

c=re.sub('"(:)', '\g<0> ', c)

c=re.sub("{", "\n{\n", c)

c=re.sub("}", "\n}", c)

c=re.sub('\]', '\n]', c)

c=re.sub('\[', '\n[\n', c)

c=re.sub('\n\n', '\n', c)

tabcount=-1

def parse(c):

c=c.split("\n")

d=""

for i in range (len(c)):

c[i]=c[i].strip()

while '' in c:

c.remove('')

tabcount=-1

for i in range (len(c)):

if c[i]=="{":

tabcount+=1

c[i]=""

elif c[i]=="}":

tabcount-=1

c[i]=""

elif c[i]=="[":

c[i]=""

if c[i+1]!="{":

while d!="]":

i=i+1

d=c[i]

c[i]="- "+d

d=""

else:

arrcount=1

arrend=0

arrtab=tabcount

arrobj=[]

for j in range (i, len(c)):

if c[j]=="[":

arrcount+=1

elif c[j]=="]":

arrcount-=1

if arrtab==tabcount:

arrobj.append(c[j])

if arrcount==0:

arrend=j

break

str="\n".join(arrobj)

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

l=0

objtab=0

objects=[]

for j in range (len(str)):

if str[j]=="{":

objtab+=1

elif str[j]=="}":

objtab-=1

if objtab==0:

objects.append(str[l:j+1])

l=j+1

for j in range (len(objects)):

objects[j]=parse(objects[j])

objects[j]="\n".join(objects[j])

if objects[j].startswith("\n"):

objects[j]=objects[j][0:1]+"- "+objects[j][1:len(objects[j])]

objects[j]=re.sub('^[^{}-]', '\t\g<0>', objects[j], 0, re.MULTILINE)

str="\n".join(objects)

str=str.split("\n")

k=0

for j in range (i, arrend-1):

c[j]=str[j-i]

c[i]=""

elif c[i]=="- ]" or c[i]=="]":

c[i]=""

elif c[i]!="":

for j in range (tabcount):

c[i]="\t"+c[i]

return(c)

c=parse(c)

c="\n".join(c)

c=re.sub(r'^\t\*\n', '', c, 0, re.MULTILINE)

c=re.sub(r'"(.\*?[^\\])"', '\g<1>', c)

print(c)

res=open("timetable.yaml", 'w', encoding="utf8")

res.write(c)

res.close()

Получившийся файл

timetable:

day: THURSDAY

lessons:

- Subject1:

time: 13:30-15:00

cabinet: 285

adress: Кронверкский пр., д.49, лит.А

name: ДИСКРЕТНАЯ МАТЕМАТИКА(ЛЕК)

week:

- нечет

teacher: Поляков Владимир Иванович

format: Очно-дистанционный

- Subject2:

time: 15:20-16:50

cabinet: 302

adress: Кронверкский пр., д.49, лит.А

name: ДИСКРЕТНАЯ МАТЕМАТИКА(ЛЕК)

week:

- чет

- нечет

teacher: Поляков Владимир Иванович

format: Очно-дистанционный

Различий с файлом из 1-го задания нет.

## Доп. задание №3

Десятикратное время выполнения кода из задания №1: ≈0.75 сек.

Десятикратное время выполнения кода из доп. задания №1: ≈0.74 сек.

Десятикратное время выполнения кода из доп. задания №2: ≈ 0.62 сек.

Причиной выигрыша в скорости программы из доп. задания №2 является факт того, что эта программа не является полноценным парсером JSON’а: в ней не учитывается всё то, что учитывается в парсинге с использованием библиотек Python’а, из-за чего программе из доп. задания №1 нужно больше времени. Программа из задания №1 не использует регулярные выражения, из-за чего работает медленнее.

## Доп. задание №4

import re

a=open("timetable.json", encoding="utf8")

c=a.read().split()

print(c)

dotcount=0

for i in range (len(c)-1):

if c[i]=="{" or (c[i]=="[" and c[i+1]!="{"):

dotcount+=1

c[i]=""

if c[i]=="[":

c[i]=""

if c[i]=="}" or c[i]=="]" or c[i]=="},":

dotcount-=1

c[i]=""

if c[i]=="\t":

c[i]=""

c[i]=" "+c[i]

if ":" in c[i]:

for j in range (dotcount):

c[i]="\*"+c[i]

c[i]="\n"+c[i]

c="".join(c)

c=re.sub(r'"(.\*?[^\\])"', '\g<1>', c)

c=re.sub(r'\[(.\*?[^\\])\]', '\g<1>', c)

c=re.sub(r',\n', '\n', c)

c=c[0:len(c)-1]

c="h1. Timetable\n"+c

print(c)

res=open("timetable.textile", 'w', encoding="utf8")

res.write(c)

res.close()

Получившийся файл

h1. Timetable

\* day: THURSDAY

\* lessons:

\*\* Subject1:

\*\*\* time:

\*\*\* 13:30-15:00

\*\*\* cabinet: 285

\*\*\* adress: Кронверкский пр., д.49, лит.А

\*\*\* name: ДИСКРЕТНАЯ МАТЕМАТИКА(ЛЕК)

\*\*\* week: нечет

\*\*\* teacher: Поляков Владимир Иванович

\*\*\* format: Очно-дистанционный

\*\* Subject2:

\*\*\* time:

\*\*\* 15:20-16:50

\*\*\* cabinet: 302

\*\*\* adress: Кронверкский пр., д.49, лит.А

\*\*\* name: ДИСКРЕТНАЯ МАТЕМАТИКА(ЛЕК)

\*\*\* week: чет, нечет

\*\*\* teacher: Поляков Владимир Иванович

\*\*\* format: Очно-дистанционный

Формат Textile используется для форматирования текста в блогах или системах управления содержимым, представляя собой удобный и читаемый язык разметки.

# Заключение

В ходе выполнения данной работы я узнал о различных языках разметки и написал программу, переводящую JSON-файл в YAML-файл