Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»



Лабораторная работа 4 «Исследование протоколов, форматов обмена информацией и языков разметки документов»

по дисциплине

«Информатика»

Вариант № 9

Группа: Р3114

Студент: Кондратьева К. М.

Преподаватель: Балакшин П. В.

г. Санкт Петербург, 2021

Оглавление

[Федеральное государственное автономное образовательное учреждение высшего образования 1](#_Toc90724226)

[«Национальный исследовательский университет ИТМО» 1](#_Toc90724227)

[Текст задания 2](#_Toc90724228)

[Выполнение 3](#_Toc90724229)

# Текст задания

1. Определить номер варианта как остаток деления на 36 порядкового номера в списке группы в ISU. В случае, если в данный день недели нет занятий, то увеличить номер варианта на восемь.

2. Понять устройство страницы с расписанием для своей группы:

https://itmo.ru/ru/schedule/0/P3110/schedule.htm

3. Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного.

4. 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.

5. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.

6. Дополнительное задание No1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

a) 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.

7. Дополнительное задание No2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

a) Переписать исходный код, добавив в него использование регулярных выражений.

b) Сравнить полученные результаты и объяснить их сходство/различие.

8. Дополнительное задание No3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

a) Используя свою исходную программу из обязательного задания, программу из дополнительного задания No1 и программу из дополнительного задания No2, сравнить десятикратное время выполнения парсинга + конвертации в цикле.

b) Проанализировать полученные результаты и объяснить их сходство/различие.

9.Дополнительное задание задание No4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную.

a) 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.

b) Проанализировать полученные результаты, объяснить особенности использованного формата.

# Выполнение

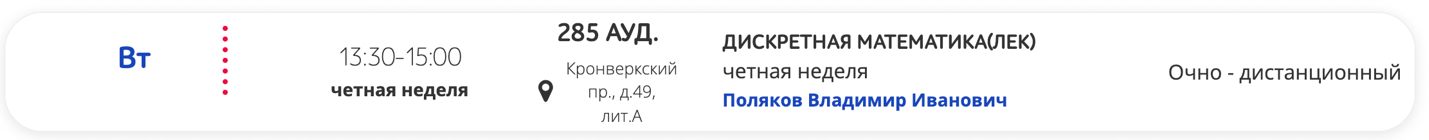
Вариант 9:



Понять устройство страницы с расписанием для своей группы:

https://itmo.ru/ru/schedule/0/P3110/schedule.htm

Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного



Файл timetable.json:

{

"timetable": {

"day": "Вт",

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

"week": "Чётная неделя",

"room": "285 ауд",

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

"lesson": "Дискретная математика (лек)",

"teacher": "Поляков Владимир Иванович",

"lesson format": "Очно-дистанционный"

}

}

Исходный код:

from timeit import Timer  
  
# libs for task 2  
import json  
import yaml  
  
# lib for task 3  
import re  
  
  
def task\_1():  
 json\_file = open("/Users/dead./Desktop/timetable.json", "r", encoding="utf-8")  
 yaml\_file = open("/Users/dead./Desktop/timetable1.yaml", 'w', encoding='utf-8')  
 json = json\_file.readlines()  
 i = -1  
 if json[0] == '{\n':  
 i = 0  
 x = len(json) - 1  
 empty = False  
 while i < x:  
 i += 1  
 flag = False  
 line = json[i]  
 k = ''  
 for j in range(2, len(line)-1):  
 if line[j] == '"' or line[j] == ',' or line[j] == '{':  
 k += ''  
 elif line[j] == '}':  
 flag = True  
 break  
 else:  
 k += line[j]  
 if flag:  
 pass  
 else:  
 k += '\n'  
 yaml\_file.write(k)  
 yaml\_file.close()  
 with open("/Users/dead./Desktop/timetable1.yaml", 'r', encoding='utf-8') as yaml\_file:  
 return yaml\_file.read()  
  
  
def task\_2():  
 with open("/Users/dead./Desktop/timetable.json", "r", encoding="utf-8") as json\_file:  
 config = json.load(json\_file)  
 with open("/Users/dead./Desktop/timetable2.yaml", 'w', encoding='utf-8') as yaml\_file:  
 yaml.dump(config, yaml\_file, allow\_unicode=True)  
 json\_file.close()  
 yaml\_file.close()  
 with open("/Users/dead./Desktop/timetable2.yaml", 'r', encoding='utf-8') as yaml\_file:  
 return yaml\_file.read()  
  
  
def task\_3():  
 json\_file = open("/Users/dead./Desktop/timetable.json", "r", encoding="utf-8")  
 yaml\_file = open("/Users/dead./Desktop/timetable3.yaml", 'w', encoding='utf-8')  
 line = json\_file.readlines()  
 for i in range(1, len(line)):  
 if re.search(r'\{|"|,', line[i]) and re.search(r'\w+', line[i]):  
 newline = re.sub(r'\{|"|,', '', line[i])  
 yaml\_file.write(newline)  
 json\_file.close()  
 yaml\_file.close()  
 with open("/Users/dead./Desktop/timetable3.yaml", 'r', encoding='utf-8') as yaml\_file:  
 return yaml\_file.read()  
  
  
print(task\_1())  
print(task\_2())  
print(task\_3())  
  
  
def task\_4():  
 print(Timer('from \_\_main\_\_ import task\_1; task\_1()').timeit(1))  
 print(Timer('from \_\_main\_\_ import task\_2; task\_2()').timeit(1))  
 print(Timer('from \_\_main\_\_ import task\_3; task\_3()').timeit(1))  
  
task\_4()