**Министерство образования и науки**

**Российской Федерации**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет программной инженерии и компьютерной техники

Дисциплина: Информатика

**Отчёт по лабораторной работе №4**

Вариант 1

Выполнил студент группы Р3133 Анисимов Максим Дмитриевич

Проверил Балакшин Павел Валерьевич

Санкт-Петербург

2022 г

Оглавление

[Задание 3](#_Toc125552201)

[Основные этапы вычисления 3](#_Toc125552202)

[Shedule.xml 3](#_Toc125552203)

[Parser.py 3](#_Toc125552204)

[Результат выполнения программы 5](#_Toc125552205)

[Вывод 6](#_Toc125552206)

# 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.

Номер варианта 1: XML 🡪 JSON

# Основные этапы вычисления

# Shedule.xml

<root>  
 <day>Пн</day>  
 <shedule>  
 <Time>08:20-09:50</Time>  
 <Week>3, 5, 7, 9, 11, 13, 15, 17</Week>  
 <Room>2435/4 (бывш. 431в) ауд.</Room>  
 <Place>Кронверкский пр., д.49, лит.А</Place>  
 <Class>Программирование(Лаб)</Class>  
 <Teacher>Письмак Алексей Евгеньевич</Teacher>  
 <Format>Очно - дистанционный</Format>  
 </shedule>  
 <shedule>  
 <Time>10:00-11:30</Time>  
 <Week>3, 5, 7, 9, 11, 13, 15, 17</Week>  
 <Room>2435/4 (бывш. 431в) ауд.</Room>  
 <Place>Кронверкский пр., д.49, лит.А</Place>  
 <Class>Программирование(Лаб)</Class>  
 <Teacher>Письмак Алексей Евгеньевич</Teacher>  
 <Format>Очно - дистанционный</Format>  
 </shedule>  
</root>

# Parser.py

import queue  
  
class Tag:  
 def \_\_init\_\_(self, i, name):  
 self.end = i  
 self.name = name  
  
def Find\_key(d, key):  
 for i in d.keys():  
 if i == key:  
 return 1  
 return 0  
  
def Text\_correcter(text):  
 for i in range(len(text)):  
 if text[i] not in [" ", "\t", "\n"]:  
 text = text[i : ]  
 break  
 if len(text) != 0 and text[0] =="\"" and text[len(text) - 1] == "\"":  
 text = text[1 : len(text) - 1]  
 return text  
  
def deserialize(text):  
 text = Text\_correcter(text)  
 q = queue.LifoQueue()  
 if len(text) == 0 or text[0] != "<":  
 return text  
 block = []  
 for i in range(len(text)):  
 if text[i] == "<":  
 j = i + 1  
 while text[j] != ">":  
 j += 1  
 tag = text[i : j + 1]  
 if tag[ : 2] == "<?":  
 i = j - 1  
 continue  
 if tag[ : 2] == "</":  
 t = q.get()  
 if q.empty():  
 if t.name == "<root>":  
 return deserialize(text[t.end + 1 : i])  
 #block[t.name[1 : len(t.name) - 1]] = deserialize(text[t.end + 1 : i])  
 block.append([t.name[1 : len(t.name) - 1], deserialize(text[t.end + 1 : i])])  
 else:  
 q.put(Tag(j, tag))  
 answers = {}  
 keys = {}  
 for b in block:  
 if Find\_key(keys, b[0]):  
 keys[b[0]] += 1  
 else:  
 keys[b[0]] = 1  
 for b in block:  
 if keys[b[0]] == 1:  
 answers[b[0]] = b[1]  
 else:  
 a = []  
 for b1 in block:  
 if b1[0] == b[0]:  
 a.append(b1[1])  
 answers[b[0]] = a  
 return answers  
  
def serialize(p, tab = ""):  
 text = ""  
 if isinstance(p, str):  
 if p.isdigit() or p == "true" or p == "false":  
 return p  
 else:  
 return "\"" + p + "\""  
 if isinstance(p, dict):  
 text = "{\n"  
 for i in p.keys():  
 text += tab + "\t" + "\"" + i + "\"" + ": " + serialize(p[i], tab + "\t") + ",\n"  
 text = text[ : len(text) - 2] + "\n"  
 text += tab + "}"  
 if isinstance(p, list):  
 text = "[\n" + tab + "\t"  
 for i in p:  
 if len(p) == 1 and i == "":  
 text += "\n"  
 break  
 text += serialize(i, tab + "\t")  
 text += ",\n" + tab + "\t"  
 if text[len(text) - 1] == "\t":  
 text = text[ : len(text) - len(tab) - 3] + "\n"  
 text += tab + "]"  
 return text  
  
file = open('Schedule.xml', 'r', encoding='utf-8')  
s = file.read()  
p = deserialize(s)  
print(serialize(p))

# Результат выполнения программы

{

"day": "Пн",

"shedule": [

{

"Time": "08:20-09:50",

"Week": "3, 5, 7, 9, 11, 13, 15, 17",

"Room": "2435/4 (бывш. 431в) ауд.",

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

"Class": "Программирование(Лаб)",

"Teacher": "Письмак Алексей Евгеньевич",

"Format": "Очно - дистанционный"

},

{

"Time": "10:00-11:30",

"Week": "3, 5, 7, 9, 11, 13, 15, 17",

"Room": "2435/4 (бывш. 431в) ауд.",

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

"Class": "Программирование(Лаб)",

"Teacher": "Письмак Алексей Евгеньевич",

"Format": "Очно - дистанционный"

}

]

}

# Вывод

Во время выполнения программы я изучил работу с библиотекой queue. Также я узнал, как записываются форматы XML и JSON. На практике смог сделать парсер из xml в json.