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 информаций

и языков разметки документов»

«ИНФОРМАТИКА»

Вариант № 13

**Выполнил:**

Студент группы P3116

Ткачев Илья

Андреевич

**Преподаватель:**

Машина Екатерина

Алексеевна

2022, Санкт-Петербург

Содержание

Задание 3

Ход работы 4

Обязательное задание 6

Дополнительное задание № 1 8

Дополнительное задание № 2 10

Дополнительное задание № 3 11

Дополнительное задание № 4 12

Вывод 7

Задание

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. Изучить форму Бэкуса-Наура.

3. Изучить особенности языков разметки/форматов JSON, YAML, XML.

4. Понять устройство страницы с расписанием для своей группы: http://itmo.ru/ru/schedule/0/P3110/schedule.htm

5. 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.

6. 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.

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

8. Дополнительное задание №1 (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.

9. Дополнительное задание №2 (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.

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

a) Используя свою исходную программу из обязательного задания, программу из дополнительного задания №1 и программу из дополнительного задания №2, сравнить

стократное время выполнения парсинга + конвертации в цикле.

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

11.Дополнительное задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).

c) 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.

d) Проанализировать полученные результаты, объяснить особенности использования формата.

12.Проверить, что все пункты задания выполнены и выполнены верно.

13.Написать отчёт о проделанной работе.

14.Подготовиться к устным вопросам на защите

Ход работы

Исходный файл tableXML.xml:

<?xml version="1.0" encoding="windows-1251"?>

<root>

<today>Вторник, 8 ноября</today>

<lessons>

<lesson1>

<time>

<start>8:20</start>

<end>9:50</end>

</time>

<lessonName>Иностранный язык</lessonName>

<teacherName>Гаипова Азиза Атахановна</teacherName>

<place>ауд. 3401, ул.Ломоносова, д.9, лит. Е</place>

<educationForm>Очно</educationForm>

</lesson1>

<lesson2>

<time>

<start>10:00</start>

<end>11:30</end>

</time>

<lessonName>Иностранный язык</lessonName>

<teacherName>Гаипова Азиза Атахановна</teacherName>

<place>ауд. 3401, ул.Ломоносова, д.9, лит. Е</place>

<educationForm>Очно</educationForm>

</lesson2>

</lessons>

</root>

Обязательное задание

osnova.py

fileXML = open('tableXML.xml', 'r', encoding='windows-1251')  
fileJSON = open('tableJSON.json', 'w')  
  
orderHeaders = ['']  
newHeader = ''  
fileJSON.write('{\n')  
linesXML = fileXML.readlines()  
fileXML.close()  
for i in range(1, len(linesXML)):  
 temp = linesXML[i].split('<')  
 a = '\t' + temp[0]  
 fileJSON.write(a)  
 if len(temp) == 2:  
 for j in temp[1]:  
 if j == '>': break  
 newHeader += j  
 if newHeader == ('/' + orderHeaders[len(orderHeaders) - 1]):  
 fileJSON.write('}')  
 orderHeaders.pop()  
 if (i < len(linesXML) - 1) and not(linesXML[i + 1].\_\_contains\_\_(orderHeaders[len(orderHeaders) - 1])):  
 fileJSON.write(',')  
 newHeader = ''  
 else:  
 orderHeaders.append(newHeader)  
 fileJSON.write('"' + newHeader + '": {')  
 newHeader = ''  
 else:  
 for j in temp[1]:  
 if j == '>': break  
 newHeader += j  
 fileJSON.write('"' + newHeader + '": "' + temp[1].split('>')[1] + '"')  
 if not(linesXML[i + 1].\_\_contains\_\_(orderHeaders[len(orderHeaders) - 1])): fileJSON.write(',')  
 newHeader = ''  
 fileJSON.write('\n')  
fileJSON.write('}')  
  
fileJSON.close()

Результат tableJSON.json:

{  
 "root": {  
 "today": "Вторник, 8 ноября",  
 "lessons": {  
 "lesson1": {  
 "time": {  
 "start": "8:20",  
 "end": "9:50"  
 },  
 "lessonName": "Иностранный язык",  
 "teacherName": "Гаипова Азиза Атахановна",  
 "place": "ауд. 3401, ул.Ломоносова, д.9, лит. Е",  
 "educationForm": "Очно"  
 },  
 "lesson2": {  
 "time": {  
 "start": "10:00",  
 "end": "11:30"  
 },  
 "lessonName": "Иностранный язык",  
 "teacherName": "Гаипова Азиза Атахановна",  
 "place": "ауд. 3401, ул.Ломоносова, д.9, лит. Е",  
 "educationForm": "Очно"  
 }  
 }  
 }  
}

Данные правильно перевелись в json формат.

Дополнительное задание № 1

dop\_1.py

import xmltodict  
import json  
  
with open('tableXML.xml') as fd:  
 doc = xmltodict.parse(fd.read())  
text = json.dumps(doc)  
  
tableJSON = u'{\n\t'  
numberTubs = 1  
for i in range(1, len(text)):  
 if text[i] == '{':  
 tableJSON += '{\n'  
 numberTubs += 1  
 tableJSON += ('\t' \* numberTubs)  
 elif (i < len(text) - 1) and (i > 0) and (text[i - 1] + text[i] + text[i + 1] == ', "'):  
 continue  
 elif (i < len(text) - 2) and (text[i] + text[i + 1] + text[i + 2] == ', "'):  
 tableJSON += ',\n'  
 tableJSON += ('\t' \* numberTubs)  
 elif text[i] == '}':  
 tableJSON += '\n'  
 numberTubs -= 1  
 tableJSON += ('\t' \* numberTubs + '}')  
 else:  
 tableJSON += text[i]  
  
fileNew = open('dop1\_JSON.json', 'w')  
fileNew.write(tableJSON)  
fileNew.close()

Результат dop1\_JSON.json:

{  
 "root": {  
 "today": "\u0412\u0442\u043e\u0440\u043d\u0438\u043a, 8 \u043d\u043e\u044f\u0431\u0440\u044f",  
 "lessons": {  
 "lesson1": {  
 "time": {  
 "start": "8:20",  
 "end": "9:50"  
 },  
 "lessonName": "\u0418\u043d\u043e\u0441\u0442\u0440\u0430\u043d\u043d\u044b\u0439 \u044f\u0437\u044b\u043a",  
 "teacherName": "\u0413\u0430\u0438\u043f\u043e\u0432\u0430 \u0410\u0437\u0438\u0437\u0430 \u0410\u0442\u0430\u0445\u0430\u043d\u043e\u0432\u043d\u0430",  
 "place": "\u0430\u0443\u0434. 3401, \u0443\u043b.\u041b\u043e\u043c\u043e\u043d\u043e\u0441\u043e\u0432\u0430, \u0434.9, \u043b\u0438\u0442. \u0415",  
 "educationForm": "\u041e\u0447\u043d\u043e"  
 },  
 "lesson2": {  
 "time": {  
 "start": "10:00",  
 "end": "11:30"  
 },  
 "lessonName": "\u0418\u043d\u043e\u0441\u0442\u0440\u0430\u043d\u043d\u044b\u0439 \u044f\u0437\u044b\u043a",  
 "teacherName": "\u0413\u0430\u0438\u043f\u043e\u0432\u0430 \u0410\u0437\u0438\u0437\u0430 \u0410\u0442\u0430\u0445\u0430\u043d\u043e\u0432\u043d\u0430",  
 "place": "\u0430\u0443\u0434. 3401, \u0443\u043b.\u041b\u043e\u043c\u043e\u043d\u043e\u0441\u043e\u0432\u0430, \u0434.9, \u043b\u0438\u0442. \u0415",  
 "educationForm": "\u041e\u0447\u043d\u043e"  
 }  
 }  
 }  
}

Результат совпадает с результатом в основном задании. Только здесь текст представлен в юникоде.

Дополнительное задание № 2

dop\_2.py

import re  
  
fileXML = open('tableXML.xml', 'r', encoding='windows-1251')  
fileJSON = open('dop2\_JSON.json', 'w')  
  
orderHeaders = ['']  
newHeader = ''  
fileJSON.write('{\n')  
linesXML = fileXML.readlines()  
for i in range(1, len(linesXML)):  
 temp = re.split(r'[<>]', linesXML[i])  
 fileJSON.write('\t' + temp[0])  
 if len(temp) == 3:  
 if orderHeaders[len(orderHeaders) - 1] == temp[1]:  
 orderHeaders.pop()  
 fileJSON.write('}')  
 if (i < len(linesXML) - 1) and not(re.fullmatch(r'\t\*</.\*>\n?', linesXML[i + 1])):  
 fileJSON.write(',')  
 fileJSON.write('\n')  
  
 else:  
 fileJSON.write(f'"{temp[1]}": ' + '{\n')  
 orderHeaders.append('/' + temp[1])  
 else:  
 fileJSON.write(f'"{temp[1]}": "{temp[2]}"')  
 if not re.fullmatch(r'\t\*</.\*\n', linesXML[i + 1]):  
 fileJSON.write(',')  
 fileJSON.write('\n')  
  
fileJSON.write('}')  
  
fileJSON.close()

Результат dop2\_JSON.json:

{  
 "root": {  
 "today": "Вторник, 8 ноября",  
 "lessons": {  
 "lesson1": {  
 "time": {  
 "start": "8:20",  
 "end": "9:50"  
 },  
 "lessonName": "Иностранный язык",  
 "teacherName": "Гаипова Азиза Атахановна",  
 "place": "ауд. 3401, ул.Ломоносова, д.9, лит. Е",  
 "educationForm": "Очно"  
 },  
 "lesson2": {  
 "time": {  
 "start": "10:00",  
 "end": "11:30"  
 },  
 "lessonName": "Иностранный язык",  
 "teacherName": "Гаипова Азиза Атахановна",  
 "place": "ауд. 3401, ул.Ломоносова, д.9, лит. Е",  
 "educationForm": "Очно"  
 }  
 }  
 }  
}

Результат совпадает с результатом в основном задании.

Дополнительное задание № 3

dop\_3.py

from datetime import datetime  
import os  
  
time100\_osnova = datetime.today() - datetime.today()  
time100\_dop\_1 = datetime.today() - datetime.today()  
time100\_dop\_2 = datetime.today() - datetime.today()  
  
for i in range(100):  
 t = datetime.today()  
 os.system('python osnova.py')  
 time100\_osnova += (datetime.today() - t)  
  
 t = datetime.today()  
 os.system('python dop\_1.py')  
 time100\_dop\_1 += (datetime.today() - t)  
  
 t = datetime.today()  
 os.system('python dop\_2.py')  
 time100\_dop\_2 += (datetime.today() - t)  
  
  
print(time100\_osnova)  
print(time100\_dop\_1)  
print(time100\_dop\_2)

0:00:04.121003

0:00:14.456653

0:00:05.488636

Быстрее всего выполнился файл без библиотек и регулярных выражений.

Дольше выполняется файл с использованием библиотек, так как в него загружаются данные библиотеки, а также библиотека переводит данные так, что после их ещё нужно обработать.

А третий файл, с регулярными выражениями, работает чуть дольше чем первый, потому что он работает как первый за исключением, что для работы со строками используются регулярные выражения, а они сравнивают всю строку. Это и замедляет программу.

Дополнительное задание № 4

dop\_4.py

import re  
  
fileXML = open('tableXML.xml', 'r', encoding='windows-1251')  
  
orderHeaders = []  
newHeader = ''  
linesXML = fileXML.readlines()  
fileXML.close()  
  
firstStr = ''  
secondStr = ''  
  
for i in range(1, len(linesXML) - 1):  
 temp0 = re.split(r'<|>', linesXML[i])  
 temp1 = re.split(r'<|>', linesXML[i + 1])  
 if len(temp0[0]) < len(temp1[0]):  
 orderHeaders.append(temp0[1])  
 elif len(temp0) == 5:  
 if firstStr != '':  
 firstStr += ','  
 secondStr += ','  
 firstStr += str(orderHeaders).replace("', '", '/')[2:-2] + f'/{temp0[1]}'  
 if temp0[2].\_\_contains\_\_(','):  
 secondStr += f'"{temp0[2]}"'  
 else:  
 secondStr += temp0[2]  
 if len(temp0[0]) > len(temp1[0]):  
 orderHeaders.pop()  
  
  
fileCSV = open('dop\_4CSV.csv', 'w', encoding='utf-8')  
fileCSV.write(firstStr)  
fileCSV.write('\n')  
fileCSV.write(secondStr)  
fileCSV.close()

Результат dop4\_CSV.csv:

root/today,root/lessons/lesson1/time/start,root/lessons/lesson1/time/end,root/lessons/lesson1/lessonName,root/lessons/lesson1/teacherName,root/lessons/lesson1/place,root/lessons/lesson1/educationForm,root/lessons/lesson2/time/start,root/lessons/lesson2/time/end,root/lessons/lesson2/lessonName,root/lessons/lesson2/teacherName,root/lessons/lesson2/place,root/lessons/lesson2/educationForm  
"Вторник, 8 ноября",8:20,9:50,Иностранный язык,Гаипова Азиза Атахановна,"ауд. 3401, ул.Ломоносова, д.9, лит. Е",Очно,10:00,11:30,Иностранный язык,Гаипова Азиза Атахановна,"ауд. 3401, ул.Ломоносова, д.9, лит. Е",Очно

Вывод

Я познакомился с форматами представления данных. Сделал программы с разными способами конвертации одного формата в другой. Узнал как вызывать выполнение программы в консоли при помощи команд python.