

Санкт-Петербургский Национальный Исследовательский
Университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

По информатике

Вариант 13

Выполнил:

Студент группы Р3116
Никулин Никита Павлович

Преподаватель:

Рудникова Тамара Владимировна



г. Санкт-Петербург

2021 г.

Задание

Порядок выполнения работы

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 протоколов и форматов обмена информацией между системами: JSON, YAML, XML.
4. Понять устройство страницы с расписанием для своей группы:
<https://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.
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 задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). а) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов. б) Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать. в) Сравнить полученные результаты и объяснить их сходство/различие.
9. Дополнительное задание задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). а) Переписать исходный код, добавив в него использование регулярных выражений. б) Сравнить полученные результаты и объяснить их сходство/различие.
10. Дополнительное задание задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). а) 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. б) Проанализировать полученные результаты и объяснить их сходство/различие.

11. Дополнительное задание задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную. а) 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. б) Проанализировать полученные результаты, объяснить особенности использованного формата.
12. Проверить, что все пункты задания выполнены и выполнены верно.
13. Написать отчёт о проделанной работе.
14. Подготовиться к устным вопросам на защите.

Рисунок 1 - Задание 13

№ варианта	Исходный формат	Результирующий формат	День недели
0	YAML	XML	Понедельник
1	JSON	XML	Понедельник
2	XML	JSON	Понедельник
3	JSON	YAML	Понедельник
4	YAML	JSON	Понедельник
5	XML	YAML	Понедельник
6	YAML	XML	Вторник
7	JSON	XML	Вторник
8	XML	JSON	Вторник
9	JSON	YAML	Вторник
10	YAML	JSON	Вторник
11	XML	YAML	Вторник
12	YAML	XML	Среда
13	JSON	XML	Среда

Основные этапы вычисления

```
{
  "Lessons": [
    {
      "time": "10.00 - 11.30",
      "room": "461",
      "lesson": "Математика",
      "lesson-format": "Очно - дистанционный",
      "week": "Нечётная",
      "teacher": "Синетова Мадина Мансуровна"
    },
    {
      "time": "11.40 - 13.10",
      "room": "306",
      "lesson": "Программирование",
      "lesson-format": "Очно - дистанционный",
      "week": "Нечётная",
      "teacher": "Сорокин Роман Борисович"
    },
    {
      "time": "13.30 - 15.00",
```

```

"room": "306",
"lesson": "Программирование",
"lesson-format": "Очно - дистанционный",
"week": "Нечётная",
"teacher": "Сорокин Роман Борисович"
},
{
"time": "10.00 - 11.30",
"room": "409",
"lesson": "Математика",
"lesson-format": "Очно - дистанционный",
"week": "Чётная",
"teacher": "Синетова Мадина Мансуровна"
},
{
"time": "11.40 - 13.10",
"room": "303/2",
"lesson": "Основы Профессиональной Деятельности",
"lesson-format": "Очно - дистанционный",
"week": "Чётная",
"teacher": "Ткешелашвили Нино Мерабиевна"
},
{
"time": "13.30 - 15.00",
"room": "303/2",
"lesson": "Основы Профессиональной Деятельности",
"lesson-format": "Очно - дистанционный",
"week": "Чётная",
"teacher": "Ткешелашвили Нино Мерабиевна"
}
]
}

```

1)

```

file = open("input.json", "r")
A = file.read()
file.close()

```

```

file = open("output.xml", "w")
file.write("<?xml version='1.0' encoding='UTF-8' ?>")

```

```

begin = True
opened = False
L = list()
mass = list()
record = True
array = list()
array.append("")
flag = False
for i in A:
    if i != '{' and i != '}' and i != '[' and i != ']' and i != '\"' and i != ',' and i != ':' and i != '\n':
        if begin is False and record is True:
            mass.append(i)
            file.write(i)
        elif i == '\"':
            if record:

```

```

    if begin:
        begin = False
        file.write('<')
    else:
        begin = True
        file.write('>')
        L.append("".join(mass))
        mass.clear()
elif i == ':':
    record = False
elif i == ',' and record is False:
    record = True
    file.write('</' + L[-1] + '>')
    L.pop(-1)
elif i == '}':
    if opened:
        file.write('</' + array[-1] + '>')
    elif record and len(L) > 0:
        file.write('</' + L[-1] + '>')
        L.pop(-1)
elif i == '[':
    record = True
    array.append(L[-1])
    L.pop(-1)
    opened = True
elif i == ']':
    opened = False
elif i == '\n':
    if record is False:
        file.write('</' + L[-1] + '>\n')
        L.pop(-1)
        record = True
    else:
        file.write('\n')
elif i == '{':
    record = True
    if opened and len(array) == 2:
        if flag:
            file.write('<' + array[-1] + '>')
        else:
            flag = True
file.close()

```

2)

```

import json
import time
from json2xml import json2xml

```

```

def read_json():
    return json.loads(open('input.json', encoding='UTF-8').read())

```

```

def write_xml(str):
    f = open('output.xml', 'w', encoding='UTF-8')
    f.write(str)
    f.close()

```

```

additional_one_time = time.time()

for _ in range(10):
    obj = read_json()

    write_xml(json2xml.Json2xml(obj, attr_type=False).to_xml())

additional_one_time = time.time() - additional_one_time

```

3)

```

import re
import time

```

```

# функция открытия JSON файла
def open_json(file_name):
    return open(file_name, encoding='UTF-8').read().split('\n')

```

```

# функция записи в XML файл
def write_xml(list):
    f = open('Schedule_2.xml', 'w', encoding='UTF-8')
    f.write('\n'.join(list))
    f.close()

```

```

# функция обрезки JSON массива
def js_cutter(js):
    js = js[1:len(js) - 1]
    for j in range(len(js)):
        js[j] = js[j][4:]
    return js

```

```

# функция, создающая массив из количества отступов каждой строки
def padding_checker():
    for a in range(len(start_json)):
        if start_json[a].startswith(' '):
            padding[a] = start_json[a].count(' ')

```

```

# функция нахождения тега
def find_tag(index):
    return re.search(r'["\w-]*"', start_json[index])[0][1:-1]

```

```

# функция нахождения текста
def find_text(index):
    return re.search(r'["\w:.,() -]*"', start_json[index])[0][3:-1]

```

```

additional_two_time = time.time()
for _ in range(10):
    start_json = open_json('Schedule.json')
    start_json = js_cutter(start_json)

    res_xml = list() # список, который будет содержать строки нашего результирующего файла
    res_xml.append('<?xml version="1.0" encoding="utf-8"?>') # первая строка XML файла

```

```
massiveFlag = False # флаг, отвечающий за вход в массив внутри JSON файла
listOfEndings = list() # список, который будет содержать окончания открытых тегов
```

```
padding = [0] * len(start_json)
padding_checker() # создаем массив количества отступов
```

```
for i in range(len(start_json)): # начинаем бегать по строкам списка JSON
    tempString = "" # переменная, в которой хранится временная строка для последующей записи
    if '"' in start_json[i]:
        tag = find_tag(i)
        # ищем теги, закрывающиеся на той же строке
        if padding[i] == padding[i + 1] or start_json[i + 1].lstrip() == '}' or start_json[i + 1].lstrip() == '}',':
            text = find_text(i)
            if massiveFlag:
                tempString += '\t' * (padding[i] - 1) + '<' + tag + '>' + text + '</' + tag + '>'
            else:
                tempString += '\t' * padding[i] + '<' + tag + '>' + text + '</' + tag + '>'
            res_xml.append(tempString)
        # ищем теги, закрывающиеся с конца файла
        elif padding[i] != padding[i + 1] and '[' not in start_json[i]:
            text_s = '\t' * padding[i] + '<' + tag + '>'
            text_e = '\t' * padding[i] + '</' + tag + '>'
            res_xml.append(text_s)
            listOfEndings.append(text_e)
        # ищем открытие массива
        elif '[' in start_json[i]:
            massiveFlag = True
            massiveTag = tag
            massivePadding = i
            tempString += ' ' * padding[massivePadding] + '<' + tag + '>'
            res_xml.append(tempString)

    # ищем закрытие массива
    if ']' in start_json[i]:
        massiveFlag = False
        tempString += ' ' * padding[massivePadding] + '</' + massiveTag + '>'
        res_xml.append(tempString)
    # ищем переходы в массиве
    if massiveFlag:
        if '}' in start_json[i] and '{' in start_json[i + 1]:
            tempString += ' ' * padding[massivePadding] + '</' + massiveTag + '>'
            res_xml.append(tempString)
            tempString = ' ' * padding[massivePadding] + '<' + massiveTag + '>'
            res_xml.append(tempString)
    # переворачиваем и присоединяем список окончаний тегов
    listOfEndings.reverse()
    res_xml += listOfEndings

write_xml(res_xml) # записываем XML в файл
```

```
additional_two_time = time.time() - additional_two_time
```

4) Время первой программы: 0.003985021 sec
Время второй программы: 0.25328723 sec
Время третьей программы: 0.002957582 sec

Список источников

1) <https://jsonformatter.org/>

2) <https://github.com/DeBug758/InformaticsLab4>

Вывод

Я научился писать парсеры, работать с ними и узнал много нового об JSON и XML