Национальный исследовательский университет информационных технологий, механики и оптики  
Факультет программной инженерии и компьютерной техники

Лабораторная работа № 3  
«Исследование языков разметки документов»

Выполнил: Лысенко Данила Сергеевич  
Группа: P3110  
Вариант: 18

Преподаватель: Балакшин Павел Валерьевич

Санкт-Петербург  
2020

**Задание**

1. Определить номер варианта как остаток деления на 35 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, PROTOBUF.

4. Понять устройство страницы с расписанием для своей группы: http://www.ifmo.ru/ru/schedule/0/P3200/schedule.htm

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

6. Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый.

7. Нельзя использовать готовые библиотеки, кроме re (регулярные выражения в Python) и библиотеки для загрузки XML-файлов.

8. Необязательное задание для получения оценки «4» и «5» (позволяет набрать от 75 до 89 процентов от максимального числа баллов БаРС за данную лабораторную).

a) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.

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

9. Необязательное задание для получения оценки «5» (позволяет набрать от 90 до 100 процентов от максимального числа баллов БаРС за данную лабораторную).

a) Используя свою программу и найденные готовые библиотеки, сравнить десятикратное время выполнения парсинга + конвертации в цикле.

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

**Выполнение работы**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55 | **import** **json**  **import** **yaml**  **import** **time**  **import** **sys**  **import** **timeit**  **def** **work1**():  input = open('input.json', 'r', encoding="utf-8")  output = open('output.yaml', 'w', encoding="utf-8")  nextLine = input.readline()  totalLines = **0**  lines = list()  indent = -**2**  **while** nextLine:  lines.append(nextLine)  totalLines += **1**  nextLine = input.readline()  **for** i **in** range(**0**, totalLines):  **if** lines[i].lstrip()[**0**] == '{':  indent += **2**  **elif** lines[i].lstrip()[**0**] == '}':  indent -= **2**  **else**:  **if** lines[i].lstrip()[**0**] == '"':  temp = lines[i].split(':', maxsplit=**1**)  temp[**0**] = temp[**0**].replace('"', '').lstrip().expandtabs(**2**)  **if** temp[**1**] != '**\n**':  temp[**1**] = ' **\'**' + temp[**1**][**1**:len(temp[**1**])].replace('"', '**\'**')  **if** temp[**1**][len(temp[**1**]) - **2**] == ',':  temp[**1**] = temp[**1**].rstrip('**\n**')  temp[**1**] = temp[**1**].rstrip(',')  temp[**1**] = temp[**1**] + '**\n**'  output.write(indent \* ' ' + temp[**0**] + ':' + temp[**1**])  input.close()  output.close()  **def** **work2**():  input = open('input.json', 'r', encoding="utf-8")  jsonData = json.load(input)  output = open('output1.yaml', 'w+', encoding="utf-8")  yaml.dump(jsonData, output, allow\_unicode=True)  input.close()  output.close()  **if** \_\_name\_\_ == '\_\_main\_\_':  start\_time1 = time.time()  **for** i **in** range(**0**, **10**):  work1()  **print**("Written parser: --- %s seconds ---" % (time.time() - start\_time1))  start\_time2 = time.time()  **for** i **in** range(**0**, **10**):  work2()  **print**("Library parser: --- %s seconds ---" % (time.time() - start\_time2)) |

**Скорость выполнения:**

Written parser: --- 0.00600123405456543 seconds ---

Library parser: --- 0.03200793266296387 seconds ---

**Вывод**

В процессе выполнения лабораторной работы я изучил синтаксис языков разметки JSON и YAM, а также получи навыки работы со строками в языке программирования Python.