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

по информатике

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

Вариант №23

Выполнил:

Эрбаев Ильдус

Группа: P3112

Приняла: Рудникова

Тамара Владимировна

г. Санкт-Петербург, 2021 г.

Оглавление

[Задание: 3](#_Toc87923378)

[Выполнение: 4](#_Toc87923385)

[Обязательное задание: 4](#_Toc87923386)

[Содержимое программы 4](#_Toc87923387)

[Сформированный YAML-файл после работы парсера 5](#_Toc87923388)

[Дополнительное задание №1: 6](#_Toc87923389)

[Содержимое программы 6](#_Toc87923390)

[Сформированный YAML-файл после работы парсера: 7](#_Toc87923391)

[Дополнительное задание №2: 7](#_Toc87923392)

[Содержимое программы 7](#_Toc87923393)

[Сформированный YAML-файл после работы парсера: 8](#_Toc87923394)

[Дополнительное задание №3: 9](#_Toc87923395)

[Сравнить десятикратное время выполнения парсинга + конвертации в цикле: 9](#_Toc87923396)

[Проанализировать полученные результаты и объяснить их сходство/различие: 9](#_Toc87923397)

[Вывод: 10](#_Toc87923398)

# 

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

Дополнительное задание 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.

Дополнительное задание 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.

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

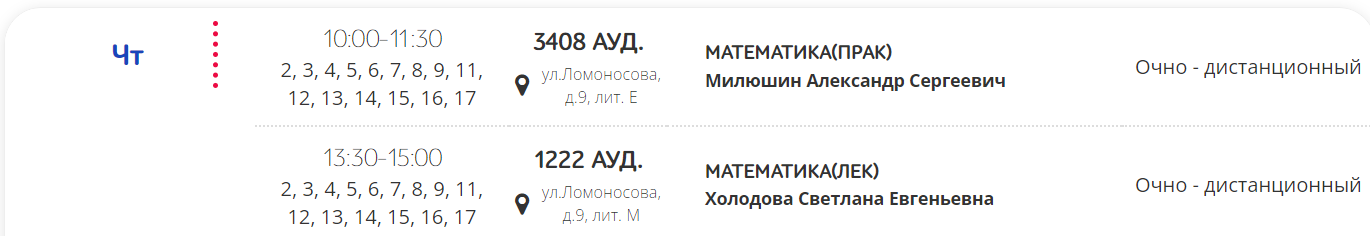
a) Используя свою исходную программу из обязательного задания, программу из дополнительного задания №1 и программу из дополнительного задания №2, сравнить десятикратное время выполнения парсинга + конвертации в цикле.

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

|  |  |  |
| --- | --- | --- |
| Исходный формат | Результирующий формат | День недели |
| XML | YAML | Четверг |

# Выполнение:

Расписание на четверг:



Расписание на четверг в формате XML:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | <schedule>  <day name="Четверг">  <lesson name="Математика (Практика)">  <time time="10:00-11:30" />  <place audience="3408" building="ул.Ломоносова, д.9, лит. Е" />  <teacher name="Милюшин Александр Сергеевич" />  <format format="Очно - дистанционный"/>  </lesson>  <lessоn name="Математика (Лекция)">  <time time="13:30-15:00" />  <place audience="1222" building="ул.Ломоносова, д.9, лит. М" />  <teacher name="Холодова Светлана Евгеньевна" />  <format format="Очно - дистанционный" />  </lessоn>  </day>  </schedule> |

## Обязательное задание:

Содержимое программы:

|  |  |
| --- | --- |
| 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  56  57  58  59  60  61  62 | file = open('thursday.xml', 'r', encoding="utf-8")  xml = file.read()  yaml = "---**\n**"  depth, i = **0**, **0**  data = []  **while** i < len(xml):  **if** xml[i:i+**2**] == '</' **or** xml[i:i+**2**] == '/>':  depth -= **2**  **elif** xml[i] == '<':  name = ''  i += **1**  **while** xml[i] != ' ' **and** xml[i] != '>':  name += xml[i]  i += **1**  **if** xml[i] == ' ':  data.append([depth, name])  depth += **1**  **while** xml[i] != '/' **and** xml[i] != '>':  **if** xml[i] == ' ':  name = ''  i += **1**  **while** xml[i] != ' ' **and** xml[i] != '=' **and** xml[i] != '/' **and** xml[i] != '>':  name += xml[i]  i += **1**  **if** xml[i] == '/' **or** xml[i] == '>':  depth -= **1**  **break**  **while** xml[i] != '=':  i += **1**  information = ''  **while** xml[i] != '"':  i += **1**  i += **1**  **while** xml[i] != '"':  information += xml[i]  i += **1**  data.append([depth, name, information])  **else**:  i += **1**  **else**:  i += **1**  information = ''  **while** xml[i] != '<':  information += xml[i]  i += **1**  **if** xml[i+**1**] == '/':  i += **1**  data.append([depth, name, information])  **else**:  data.append([depth, name])  depth += **1**  i += **1**  **for** i **in** range(len(data)):  yaml += (' '\*data[i][**0**])+str(data[i][**1**])+':'  **if** len(data[i]) == **3**:  yaml += ' '+str(data[i][**2**])+'**\n**'  **else**:  yaml += '**\n**'  file = open('thursday.yaml', 'w')  file.write(yaml)  file.close() |

Сформированный YAML-файл после работы парсера:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | **---**  schedule:  lesson:  name: Математика (Практика)  time:  time: 10:00-11:30  place:  audience: 3408  building: ул.Ломоносова, д.9, лит. Е  teacher:  name: Милюшин Александр Сергеевич  format:  format: Очно - дистанционный  lessоn:  name: Математика (Лекция)  time:  time: 13:30-15:00  place:  audience: 1222  building: ул.Ломоносова, д.9, лит. М  teacher:  name: Холодова Светлана Евгеньевна  format:  format: Очно - дистанционный |

## Дополнительное задание №1:

Содержимое программы:

|  |  |
| --- | --- |
| 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 | **import** **xmltodict**  **with** open('thursday.xml', encoding="utf-8") **as** xml\_data:  xml\_parsed = xmltodict.parse(xml\_data.read())  **def** **pars**(cur\_depth, list, yaml):  **for** j **in** list:  **if** len(str(j)) != len(j):  **for** jj **in** j:  **if** jj[**0**] == '@':  yaml += ('**\n**'+' '\*cur\_depth+str(jj[**1**:])+':')  **else**:  yaml += ('**\n**'+' '\*cur\_depth+str(jj)+':')  yaml = pars(cur\_depth+**1**, j[jj], yaml)  **continue**  **try**:  list[j]  **except** **TypeError**:  yaml += (' '+str(list))  **break**    **if** j[**0**] == '@':  yaml += ('**\n**'+' '\*cur\_depth+str(j[**1**:])+':')  **else**:  yaml += ('**\n**'+' '\*cur\_depth+str(j)+':')  yaml = pars(cur\_depth+**1**, list[j], yaml)  **return** yaml  yaml = pars(**0**, xml\_parsed, '---')  file = open('thursday.yaml', 'w')  file.write(yaml)  file.close() |

### Сформированный YAML-файл после работы парсера:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | **---**  schedule:  lesson:  name: Математика (Практика)  time:  time: 10:00-11:30  place:  audience: 3408  building: ул.Ломоносова, д.9, лит. Е  teacher:  name: Милюшин Александр Сергеевич  format:  format: Очно - дистанционный  lessоn:  name: Математика (Лекция)  time:  time: 13:30-15:00  place:  audience: 1222  building: ул.Ломоносова, д.9, лит. М  teacher:  name: Холодова Светлана Евгеньевна  format:  format: Очно - дистанционный |

## Дополнительное задание №2:

Содержимое программы:

|  |  |
| --- | --- |
| 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  56  57  58  59  60  61  62  63  64  65  66  67  68  69 | **import** **re**  file = open('thursday.xml', 'r', encoding="utf-8")  list = re.findall(r'[^\t]\*', file.read())  xml = ''  **for** i **in** list:  xml += str(i)  yaml = "---**\n**"  depth, i = **0**, **0**  data = []  **while** i < len(xml):  **if** xml[i:i+**2**] == '</' **or** xml[i:i+**2**] == '/>':  depth -= **2**  **elif** xml[i] == '<':  name = ''  i += **1**  **while** xml[i] != ' ' **and** xml[i] != '>':  name += xml[i]  i += **1**  **if** xml[i] == ' ':  data.append([depth, name])  depth += **1**  **while** xml[i] != '/' **and** xml[i] != '>':  **if** xml[i] == ' ':  name = ''  i += **1**  **while** xml[i] != ' ' **and** xml[i] != '=' **and** xml[i] != '/' **and** xml[i] != '>':  name += xml[i]  i += **1**  **if** xml[i] == '/' **or** xml[i] == '>':  depth -= **1**  **break**  **while** xml[i] != '=':  i += **1**  information = ''  **while** xml[i] != '"':  i += **1**  i += **1**  **while** xml[i] != '"':  information += xml[i]  i += **1**  data.append([depth, name, information])  **else**:  i += **1**  **else**:  i += **1**  information = ''  **while** xml[i] != '<':  information += xml[i]  i += **1**  **if** xml[i+**1**] == '/':  i += **1**  data.append([depth, name, information])  **else**:  data.append([depth, name])  depth += **1**  i += **1**  **for** i **in** range(len(data)):  yaml += (' '\*data[i][**0**])+str(data[i][**1**])+':'  **if** len(data[i]) == **3**:  yaml += ' '+str(data[i][**2**])+'**\n**'  **else**:  yaml += '**\n**'  file = open('thursday.yaml', 'w')  file.write(yaml)  file.close() |

### Сформированный YAML-файл после работы парсера:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | **---**  schedule:  lesson:  name: Математика (Практика)  time:  time: 10:00-11:30  place:  audience: 3408  building: ул.Ломоносова, д.9, лит. Е  teacher:  name: Милюшин Александр Сергеевич  format:  format: Очно - дистанционный  lessоn:  name: Математика (Лекция)  time:  time: 13:30-15:00  place:  audience: 1222  building: ул.Ломоносова, д.9, лит. М  teacher:  name: Холодова Светлана Евгеньевна  format:  format: Очно - дистанционный |

## Дополнительное задание №3:

### Сравнить десятикратное время выполнения парсинга + конвертации в цикле:

При помощи библиотеки time нам удалось получить десятикратное время выполнения программы:

0.006581922149658203 - Исходная программа.

0.001950979232788086 - Программа с использованием готовых библиотек.

0.007043941116333008 - Программа с использованием регулярных выражений.

Как мы видим, больше всего отличилась программа с использованием библиотеки xmltodict, она в 3,4 раза быстрее исходной программы и в 3,6 раза быстрее программы с использованием регулярных выражений. Исходная программа же быстрее программы с использованием регулярных выражений на 7 процентов (в 1,07 раз).

### Проанализировать полученные результаты и объяснить их сходство/различие:

При выполнении простых парсингов все программы справляются на отлично, но использование библиотек, во-первых, значительно уменьшает время на выполнение за счёт специальных алгоритмов, во-вторых, с ними проще работать, в-третьих, программа гораздо функциональна по сравнению с исходной программой и программой с использованием регулярных выражений. Программа с использованием регулярных выражений работает немного медленнее исходной, так как все мы знаем, что регулярные выражения не самые быстрые, но с ней проще и приятнее работать.

# Вывод:

В ходе выполнения лабораторной работы я познакомился с работой Python’а, многими его функциями, поближе с библиотекой регулярных выражений, узнал об устройстве и строении парсингов.