

Федеральное государственное автономное образовательное учреждение  
высшего образования «Национальный исследовательский университет  
ИТМО»

*Факультет программной инженерии и компьютерной техники*

## **Лабораторная работа №4**

**По дисциплине “Информатика”**

Вариант: 17

XML → YAML

Выполнил:  
Саранча Павел

Группа: Р3109

Преподаватель:  
Рудникова Тамара Владимировна

Санкт-Петербург, 2023г

## Оглавление

Задание: .....	2
Исходный XML файл с именем 'schedule.xml': .....	3
Обязательное задание: .....	3
Доп задание №1: .....	5
Доп задание №3: .....	7
Доп задание №4: .....	9
Доп задание №5, Перевод из XML в html: .....	9
Заключение: .....	11
Список литературы: .....	12

## Задание:

1. (Обязательное задание) Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый путём простой замены метасимволов исходного формата на метасимволы результирующего формата.
2. (Доп задание №1) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.
3. (Доп задание №3) Переписать исходный код таким образом, чтобы для решения задачи использовались формальные грамматики. То есть ваш код должен уметь осуществлять парсинг и конвертацию любых данных, представленных в исходном формате, в данные, представленные в результирующем формате: как с готовыми библиотеками из дополнительного задания №1.
4. (Доп задание №4) Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле.
5. (Доп задание №4) 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.

Исходный XML файл с именем 'schedule.xml':

```
1 <timetable>
2   <lang> ru </lang>
3   <day> Wednesday </day>
4   <lessons>
5     <lesson1>
6       <teacher> Балакшин Павел Валерьевич </teacher>
7       <time> 08:20-09:50 </time>
8       <group> ИНФОРМ 3 </group>
9       <room> Актальный зал </room>
10      <place> ул.Ломоносова, д.9, лит. М </place>
11      <subject> ИНФОРМАТИКА (ЛЕК): АКТОВЫЙ ЗАЛ</subject>
12      <lesson-format> Очно-дистанционный </lesson-format>
13    </lesson1>
14    <lesson2>
15      <teacher> Клименков Сергей Викторович </teacher>
16      <time> 10:00-11:30 </time>
17      <group> ОСН ПРОФ ДЕЯТ 1 </group>
18      <room> Актальный зал </room>
19      <place> ул.Ломоносова, д.9, лит. М </place>
20      <subject> ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ (ЛЕК) : АКТОВЫЙ ЗАЛ</subject>
21      <lesson-format> Очно-дистанционный </lesson-format>
22    </lesson2>
23  </lessons>
24 </timetable>
```

Обязательное задание:

1. `# -*- coding: utf-8 -*-`
2. `import time`
- 3.
4. `start_time = time.time()`
- 5.
6. `fin = open('schedule.xml', encoding='utf-8')`

```
7. fout = open('schedule.yaml', 'w+')
8.
9. read = 0
10. word = ""
11. for line in fin:
12.     xml = line
13.     yaml = ""
14.     for i in range(len(xml)):
15.
16.         if read != 2 and xml[i] == '\t' or xml[i] == ' ' or xml[i] == '\n':
17.             yaml = yaml + xml[i]
18.         elif xml[i] == '<':
19.             if xml[i + 1] != '/':
20.                 read = 1
21.             else:
22.                 read = -1
23.         elif xml[i] == '>':
24.             if read == 1:
25.                 yaml = yaml + ':'
26.                 read = 2
27.             elif read == -1:
28.                 if word != "":
29.                     if not word.isnumeric():
30.                         yaml = yaml + "" + word + ""
31.                         word = ""
32.                     else:
33.                         yaml = yaml + word
34.                         word = ""
35.                 read = 0
36.             elif read == 1:
37.                 yaml = yaml + xml[i]
38.             elif read == 2:
39.                 word = word + xml[i]
40.     fout.write(yaml)
41.
42. fout.close()
43.
44. end_time = time.time()
45.
46. # Стократное время выполнения программы
47. execution_time = (end_time - start_time) * 100
48. print(f"General Стократное время выполнения программы: {execution_time} секунд")
```

Вывод (Обязательного задание):

```
1 timetable:
2   lang: "ru"
3   day: "Wednesday"
4   lessons:
5     lesson1:
6       teacher: "БалакшинПавелВалерьевич"
7       time: "08:20-09:50"
8       group: "ИНФОРМ3"
9       room: "Актныйзал"
10      place: "ул.Ломоносова,д.9,лит.М"
11      subject: "ИНФОРМАТИКА(ЛЕК):АКТОВЫЙЗАЛ"
12      lesson-format: "Очно-дистанционный"
13
14     lesson2:
15       teacher: "КлименковСергейВикторович"
16       time: "10:00-11:30"
17       group: "ОСНПРОФДЕЯТ1"
18       room: "Актныйзал"
19       place: "ул.Ломоносова,д.9,лит.М"
20       subject: "ОСНОВЫПРОФЕССИОНАЛЬНОЙДЕЯТЕЛЬНОСТИ(ЛЕК):АКТОВЫЙЗАЛ"
21       lesson-format: "Очно-дистанционный"
22
```

Доп задание №1:

```
1. start_time = time.time()
2. # Чтение содержимого XML-файла
3. tree = ET.parse('schedule.xml')
4. root = tree.getroot()
5.
6. # Функция для рекурсивного преобразования элементов XML в словарь
7. def xml_to_dict(element):
8.     if len(element) == 0:
9.         return element.text
10.     result = {}
11.     for child in element:
12.         child_data = xml_to_dict(child)
13.         if child.tag in result:
14.             if type(result[child.tag]) is list:
15.                 result[child.tag].append(child_data)
16.             else:
17.                 result[child.tag] = [result[child.tag], child_data]
18.         else:
19.             result[child.tag] = child_data
20.     return result
21.
```



### Доп задание №3:

```
1. # -*- coding: utf-8 -*-
2. import time
3.
4.
5. PUNCTUATION = r'""!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~'
6.
7.
8. def get_open_tag_name(s: str) -> str:
9.     for i in range(len(s)):
10.         if s[i] == '>':
11.             return s[i+1]
12.
13.     raise ValueError('Key not found')
14.
15.
16. def get_close_tag_name(s: str) -> str:
17.     return s.replace('<', '</')
18.
19.
20. def get_text_without_tag(text: str, open_tag: str, close_tag: str):
21.     result = text.replace(open_tag, "").replace(close_tag, "").strip()
22.
23.     if result:
24.         if not result.isalnum() or not result.isascii():
25.             result = f'\{result}\n'
26.
27.     return result
28.
29.
30. def get_yaml_content(lines) -> str:
31.     open_tags = []
32.
33.     yaml_content = []
34.     for row in lines:
35.
36.         tab_lenght = '\t' * len(open_tags)
37.         tmp_row = row.strip()
38.
39.         if tmp_row[0] == '<' and tmp_row[1] not in PUNCTUATION:
40.             open_tag = get_open_tag_name(tmp_row)
41.             close_tag = get_close_tag_name(open_tag)
42.             text = get_text_without_tag(tmp_row, open_tag, close_tag)
43.
```

```

44.         if tmp_row[-len(close_tag):] != close_tag:
45.             open_tags.append(close_tag)
46.
47.         yml_content.append(f'{tab_lenght}{open_tag[1:-1]}: {text}')
48.
49.     elif tmp_row[0] == '<' and tmp_row[1] == '/':
50.         if open_tags[-1] != tmp_row:
51.             ValueError('Bad XML file')
52.         open_tags.pop()
53.
54.     if len(open_tags) != 0:
55.         ValueError('Bad XML file')
56.
57.     return '\n'.join(yml_content)
58.
59.
60. def main():
61.     start_time = time.time()
62.     with open('schedule.xml', mode='r', encoding='utf-8') as f:
63.         xml_lines = f.readlines()
64.
65.     yml_content = get_yaml_content(xml_lines)
66.
67.     with open('schedule.yaml', mode='w') as f:
68.         f.write(yml_content)
69.     end_time = time.time()
70.     # Стократное время выполнения программы
71.     execution_time = (end_time - start_time) * 100
72.     print(f"dop 3 Стократное время выполнения программы: {execution_time}
        секунд")
73.
74.
75. if __name__ == '__main__':
76.     main()

```



Вывод (Доп задание №3):

```
1 <timetable>
2   <lang> ru </lang>
3   <day> Wednesday </day>
4   <lessons>
5     <lesson1>
6       <teacher> Балакшин Павел Валерьевич </teacher>
7       <time> 08:20-09:50 </time>
8       <group> ИНФОРМ 3 </group>
9       <room> Актальный зал </room>
10      <place> ул.Ломоносова, д.9, лит. М </place>
11      <subject> ИНФОРМАТИКА (ЛЕК): АКТОВЫЙ ЗАЛ</subject>
12      <lesson-format> Очно-дистанционный </lesson-format>
13    </lesson1>
14    <lesson2>
15      <teacher> Клименков Сергей Викторович </teacher>
16      <time> 10:00-11:30 </time>
17      <group> ОСН ПРОФ ДЕЯТ 1 </group>
18      <room> Актальный зал </room>
19      <place> ул.Ломоносова, д.9, лит. М </place>
20      <subject> ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ (ЛЕК) : АКТОВЫЙ ЗАЛ</subject>
21      <lesson-format> Очно-дистанционный </lesson-format>
22    </lesson2>
23  </lessons>
24</timetable>
```

Доп задание №4:

```
1 General Стократное время выполнения программы: 0.09968280792236328 секунд
2
3 dop1 Стократное время выполнения программы: 0.4000425338745117 секунд
4
5 dop3 Стократное время выполнения программы: 0.10018348693847656 секунд
6
```

Доп задание №5, Перевод из XML в html:

1. `# -*- coding: utf-8 -*-`
2. `import xml.etree.ElementTree as ET`
3. `from html import escape`
- 4.
- 5.
6. `def convert_xml_to_html(xml_file_path, html_file_path):`
7. `# Парсинг XML файла`
8. `tree = ET.parse(xml_file_path)`
9. `root = tree.getroot()`
- 10.
11. `# Открытие HTML файла для записи`
12. `with open(html_file_path, 'w', encoding='utf-8') as html_file:`
13. `# Запись начальных тегов HTML`

```

14.     html_file.write('<!DOCTYPE  html>\n<html>\n<head>\n<title>XML  to
HTML Conversion</title>\n</head>\n<body>\n')
15.
16.     # Рекурсивная функция для обхода элементов XML и их преобразования
    в HTML
17.     def convert_element_to_html(element):
18.         nonlocal html_file
19.
20.         # Запись открывающего тега
21.         html_file.write(f'<{escape(element.tag)}>')
22.
23.         # Запись атрибутов
24.         for key, value in element.attrib.items():
25.             html_file.write(f' {escape(key)}="{escape(value)}"')
26.
27.         # Рекурсивный вызов для дочерних элементов
28.         for child in element:
29.             convert_element_to_html(child)
30.
31.         # Запись закрывающего тега
32.         html_file.write(f'</{escape(element.tag)}>\n')
33.
34.         # Начало обхода элементов XML
35.         for child in root:
36.             convert_element_to_html(child)
37.
38.         # Запись закрывающих тегов HTML
39.         html_file.write('</body>\n</html>')
40.
41. # Пример использования
42. convert_xml_to_html('schedule.xml', 'schedule.html')

```

Вывод (Доп задание №5):

```
1      <!DOCTYPE html>
2      <html>
3      <head>
4          <title>XML to HTML Conversion</title>
5      </head>
6      <body>
7          <lang></lang>
8          <day></day>
9      <lessons><lesson1><teacher></teacher>
10         <time></time>
11         <group></group>
12         <room></room>
13         <place></place>
14         <subject></subject>
15         <lesson-format></lesson-format>
16     </lesson1>
17     <lesson2><teacher></teacher>
18         <time></time>
19         <group></group>
20         <room></room>
21         <place></place>
22         <subject></subject>
23         <lesson-format></lesson-format>
24     </lesson2>
25 </lessons>
26 </body>
27 </html>
28
```

### Заключение:

В процессе выполнения этой работы я узнал об особенностях структуре таких форматов, как XML, YAML и html, совершал преобразования между этими форматами, познакомился с парсингом на языке python.

## Список литературы:

1. Балакшин П.В., Соснин В.В., Калинин И.В., Малышева Т.А., Раков С.В., Рущенко Н.Г., Дергачев А.М. Информатика: лабораторные работы и тесты: Учебно-методическое пособие / Рецензент: Поляков В.И. - Санкт-Петербург: Университет ИТМО, 2019. - 56 с. - экз. - Режим доступа:  
[https://books.ifmo.ru/book/2248/informatika:\\_laboratornye\\_raboty\\_i\\_testy:\\_uchebno-metodicheskoe\\_posobie\\_/recenzent:\\_polyakov\\_v.i..htm](https://books.ifmo.ru/book/2248/informatika:_laboratornye_raboty_i_testy:_uchebno-metodicheskoe_posobie_/recenzent:_polyakov_v.i..htm)
1. Грошев А.С. Г89 Информатика: Учебник для вузов / А.С. Грошев. – Архангельск, Арханг. гос. техн. ун-т, 2010. -470с. -Режим доступа <https://narfu.ru/university/library/books/0690.pdf>
2. Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем: Учебник для вузов. 2-е изд. – СПб.: Питер, 2011. – 688 с.: ил.
3. Алексеев Е.Г., Богатырев С.Д. Информатика. Мультимедийный электронный учебник. – Режим доступа: <http://inf.e-alekseev.ru/text/toc.html>