Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Факультет Программной Инженерии и Компьютерной техники

Лабораторная работа №4

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант 17

Выполнила:

Фонарева Виктория Сергеевна

Группа P3110

Проверил:

Рыбаков Степан Дмитриевич

**Оглавление**

[Задание 3](#_Toc183601856)

[Задание 1 4](#_Toc183601857)

[Задание 2 5](#_Toc183601858)

[Задание 3 7](#_Toc183601859)

[Задание 4 8](#_Toc183601860)

[Задание 5 11](#_Toc183601861)

[Задание 6 12](#_Toc183601862)

# Задание

1. Написать программу на языке Python 3.x или любом другом, которая бы осуществляла парсинг и конвертацию исходного файла в новый путём простой замены метасимволов
2. Дополнительное задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
   2. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.
   3. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
3. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. Переписать исходный код, добавив в него использование регулярных выражений.
   2. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
4. Дополнительное задание № 3 (позволяет набрать +25 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. 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.
   2. Проверку осуществить как минимум для расписания с двумя учебными днями по два занятия в каждом.
   3. Сравнить полученные результаты и объяснить их сходство
5. Дополнительное задание № 4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле.
   2. Проанализировать полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

# Задание 1

Код решения:

|  |
| --- |
| *inp* = open('schedule.xml', "r", encoding="utf-8")  out = open("schedule.yaml", "w", encoding='utf-8')  level = 0  s = ""  for tag in inp:  if tag.*count*('<') == 1:  if tag.*count*('/') != 0:  level -= 1  continue  name = ''  i = 0  while(tag[i] != '<'):  i += 1  i += 1  while (tag[i] != '>'):  name += tag[i]  i += 1  s += ' '\*level + name + ": \n"  level += 1  else:  name = ''  value = ''  i = 0  while (tag[i] != '<'):  i += 1  i += 1  while (tag[i] != '>'):  name += tag[i]  i += 1  i += 1  while (tag[i] != '<'):  value += tag[i]  i += 1  s += ' '\*level + name + ": " + value + "\n"  out.*write*(s) |

Результат:

|  |
| --- |
| schedule:  wednesday:  lesson1:  time: 08:20 - 09:50  subject: Информатика  lecturer: Балакшин Павел Валерьевич  room: Актовый зал (1216/0)  address: ул. Ломоносова, д.9, лит. М  lesson2:  time: 10:00 - 11:30  subject: Основы профессиональной деятельности  lecturer: Клименков Сергей Викторович  room: Актовый зал (1216/0)  address: ул. Ломоносова, д.9, лит. М  lesson3:  time: 15:20 - 16:50  subject: Английский язык B1.1  lecturer: Морозова Анна Михайловна  room: 3213  address: ул. Ломоносова, д.9, лит. Е  lesson4:  time: 17:00 - 18:30  subject: Английский язык B1.1  lecturer: Морозова Анна Михайловна  room: 3213  address: ул. Ломоносова, д.9, лит. Е  saturday:  lesson1:  time: 08:20 - 09:50  subject: Математический анализ (продвинутый уровень)  lecturer: Холодова Светлана Евгеньевна  room: 1405  address: Кронверкский пр., д.49, лит. А  lesson2:  time: 10:00 - 11:30  subject: Математический анализ (продвинутый уровень)  lecturer: Холодова Светлана Евгеньевна  room: 1405  address: Кронверкский пр., д.49, лит. А  lesson3:  time: 13:30 - 15:00  subject: Основы профессиональной деятельности  lecturer: Остапенко Иван Вадимович  room: 2116  address: Кронверкский пр., д.49, лит. А  lesson4:  time: 15:20 - 16:50  subject: Основы профессиональной деятельности  lecturer: Остапеко Иван Вадимович  room: 2116  address: Кронверкский пр., д.49, лит. А |

# Задание 2

Код решения:

|  |
| --- |
| *inp* = open('schedule.xml', "r", encoding="utf-8")  out = open("schedule.yaml", "w", encoding='utf-8')  import xml.*etree*.*ElementTree*  import yaml  tree = xml.*etree*.*ElementTree*.*parse*('schedule.xml')  root = tree.*getroot*()  def parse\_xml(el):  res = {}  for d in el:  if len(d) != 0: # Если есть вложенные элементы  res[d.*tag*] = parse\_xml(d)  else:  res[d.*tag*] = d.*text*  return res  yaml\_res = yaml.*dump*({root.*tag*: parse\_xml(root)}, allow\_unicode=True)  print(yaml\_res) |

Результат:

|  |
| --- |
| schedule:  saturday:  lesson1:  address: Кронверкский пр., д.49, лит. А  lecturer: Холодова Светлана Евгеньевна  room: '1405'  subject: Математический анализ (продвинутый уровень)  time: 08:20 - 09:50  lesson2:  address: Кронверкский пр., д.49, лит. А  lecturer: Холодова Светлана Евгеньевна  room: '1405'  subject: Математический анализ (продвинутый уровень)  time: 10:00 - 11:30  lesson3:  address: Кронверкский пр., д.49, лит. А  lecturer: Остапенко Иван Вадимович  room: '2116'  subject: Основы профессиональной деятельности  time: 13:30 - 15:00  lesson4:  address: Кронверкский пр., д.49, лит. А  lecturer: Остапеко Иван Вадимович  room: '2116'  subject: Основы профессиональной деятельности  time: 15:20 - 16:50  wednesday:  lesson1:  address: ул. Ломоносова, д.9, лит. М  lecturer: Балакшин Павел Валерьевич  room: Актовый зал (1216/0)  subject: Информатика  time: 08:20 - 09:50  lesson2:  address: ул. Ломоносова, д.9, лит. М  lecturer: Клименков Сергей Викторович  room: Актовый зал (1216/0)  subject: Основы профессиональной деятельности  time: 10:00 - 11:30  lesson3:  address: ул. Ломоносова, д.9, лит. Е  lecturer: Морозова Анна Михайловна  room: '3213'  subject: Английский язык B1.1  time: 15:20 - 16:50  lesson4:  address: ул. Ломоносова, д.9, лит. Е  lecturer: Морозова Анна Михайловна  room: '3213'  subject: Английский язык B1.1  time: 17:00 - 18:30 |

Сходства результатов выполнения кода 1 и 2 задания

Результат содержит одинаковые данные, одинаковую структуру.

Различия результатов выполнения кода 1 и 2 задания.

Разный порядок тегов (во втором случае они отсортированы по алфавиту). Числовые значения стали обернуты в кавычки.

# Задание 3

Код решения:

|  |
| --- |
| import re  res = ''  pattern = r'<(\w+)>(.\*?)<\/\1>'  def parse(tag:str, content:str, level:int):  if content.*count*('<') == 0:  return (' '\*level + tag + ": " + content)  s = ''  matches = re.*findall*(pattern,content)  s += ' ' \* level + tag + ':\n'  for i in range(0,len(matches)):  s += parse(matches[i][0],matches[i][1],level+1)  if (i != len(matches) - 1):  s += '\n'  return s  inp = open('schedule.xml', "r", encoding="utf-8")  out = open("schedule.yaml", "w", encoding='utf-8')  s = inp.*read*()  s = s.*replace*('\n','')  matches = re.*findall*(pattern,s)  out.write(parse(matches[0][0],matches[0][1],0)) |

Результат:

|  |
| --- |
| schedule:  wednesday:  lesson1:  time: 08:20 - 09:50  subject: Информатика  lecturer: Балакшин Павел Валерьевич  room: Актовый зал (1216/0)  address: ул. Ломоносова, д.9, лит. М  lesson2:  time: 10:00 - 11:30  subject: Основы профессиональной деятельности  lecturer: Клименков Сергей Викторович  room: Актовый зал (1216/0)  address: ул. Ломоносова, д.9, лит. М  lesson3:  time: 15:20 - 16:50  subject: Английский язык B1.1  lecturer: Морозова Анна Михайловна  room: 3213  address: ул. Ломоносова, д.9, лит. Е  lesson4:  time: 17:00 - 18:30  subject: Английский язык B1.1  lecturer: Морозова Анна Михайловна  room: 3213  address: ул. Ломоносова, д.9, лит. Е  saturday:  lesson1:  time: 08:20 - 09:50  subject: Математический анализ (продвинутый уровень)  lecturer: Холодова Светлана Евгеньевна  room: 1405  address: Кронверкский пр., д.49, лит. А  lesson2:  time: 10:00 - 11:30  subject: Математический анализ (продвинутый уровень)  lecturer: Холодова Светлана Евгеньевна  room: 1405  address: Кронверкский пр., д.49, лит. А  lesson3:  time: 13:30 - 15:00  subject: Основы профессиональной деятельности  lecturer: Остапенко Иван Вадимович  room: 2116  address: Кронверкский пр., д.49, лит. А  lesson4:  time: 15:20 - 16:50  subject: Основы профессиональной деятельности  lecturer: Остапеко Иван Вадимович  room: 2116  address: Кронверкский пр., д.49, лит. А |

Сходства результатов выполнения кода 2 и 3 задания.

Результат содержит одинаковые данные, одинаковую структуру.

Различия результатов выполнения кода 2 и 3 задания.

Разный порядок тегов (во первом случае они отсортированы по алфавиту). Числовые значения в 3 задании, в отличие от 2, не обернуты в кавычки.

# Задание 4

Код решения:

|  |
| --- |
| import re  res = ''  pattern = r'<(\w+)>(.\*?)<\/\1>'  opened\_tag = 'OPEN\_TAG'  closed\_tag = 'CLOSED\_TAG'  text = 'TEXT'  whitespace = 'WHITESPACE'  structure\_attributes\_name = 'attributes'  lex = {opened\_tag: r'<(\w+)((\s[^"=\s]+="\w+")\*)>',  closed\_tag: r'<\/(\w+)>',  text: r'([^<]+)',  whitespace: r'(\s+)'}  tags = []  class Tag:  def \_\_init\_\_(self, name, content: list):  self.*name* = name  self.*content* = content  self.*extend\_list\_tags* = []  def \_\_str\_\_(self):  return f"{self.*name*}: {self.*content*}"  def \_\_repr\_\_(self):  return self.*\_\_str\_\_*()  def regular\_exp\_element(tag):  pass  def parse(s: str, pos: int):  level = 0  opened\_tags = []  while pos < len(s):  match = re.*match*(lex[whitespace], s[pos:])  if match:  pos += match.*end*()  continue  match = re.*match*(lex[opened\_tag], s[pos:])  if match:  tag = Tag(match.*group*(1), [])  attributes = match.*group*(2)  if attributes:  attr\_list\_tag = Tag(structure\_attributes\_name, [])  attributes = attributes[1:] #убираем первый пробел  for full\_attr in attributes.*split*(' '):  full\_attr\_splitted = full\_attr.*split*('=')  name = full\_attr\_splitted[0]  value = full\_attr\_splitted[1]  value = value[1:len(value) - 1]  attr\_tag = Tag(name, value)  attr\_list\_tag.*content*.*append*(attr\_tag)  tag.*content*.*append*(attr\_list\_tag)  opened\_tags.*append*(tag)  pos += match.*end*()  continue  match = re.*match*(lex[text], s[pos:])  if match:  founded\_text = match.*group*(1)  pos += match.*end*()  match = re.*match*(lex[closed\_tag], s[pos:])  if not match:  print("Неправильный XML формат")  return  pos += match.*end*()  tag = Tag(match.*group*(1), founded\_text)  opened\_tags.*pop*()  last\_tag = opened\_tags[-1]  last\_tag.*content*.*append*(tag)  continue  match = re.*match*(lex[closed\_tag], s[pos:])  if match:  cur\_tag = opened\_tags[-1]  if len(opened\_tags) == 1:  tags.*append*(opened\_tags[-1]) #добавляем высший  opened\_tags.*pop*()  if len(opened\_tags) != 0:  last\_tag = opened\_tags[-1]  analogy\_name = False  for last\_tag\_child in last\_tag.*content*:  last\_tag\_child: Tag = last\_tag\_child  if last\_tag\_child.*name* == cur\_tag.*name*:  last\_tag\_child.*extend\_list\_tags*.*append*(cur\_tag)  analogy\_name = True  break  if not analogy\_name:  last\_tag.*content*.*append*(cur\_tag)  pos += match.*end*()  continue  pos += 1  def xml\_to\_yaml(tag: Tag, level: int):  s = ' ' \* level + tag.*name* + ":" + '\n'  if not len(tag.*extend\_list\_tags*):  for child in tag.*content*:  if (isinstance(child, str)):  return level \* ' ' + tag.*name* + ": " + tag.*content* + '\n'  s += xml\_to\_yaml(child, level + 1)  return s  i = 0  for child in tag.*content*:  child:Tag = child  s += ' '\*(level+1) + ('-' if i == 0 else ' ') + xml\_to\_yaml(child, level + 1)[2\*(level+1)-1:]  i += 1  for ex\_tag in tag.*extend\_list\_tags*:  i = 0  for ex\_tag\_child in ex\_tag.*content*:  ex\_tag\_child: Tag = ex\_tag\_child  s += (' ' \* (level + 1) + ('-' if i == 0 else ' ') +  xml\_to\_yaml(ex\_tag\_child, level + 1)[2 \* (level + 1) - 1:])  i += 1  return s  inp = open('sc.xml', "r", encoding="utf-8")  out = open("schedule.yaml", "w", encoding='utf-8')  s = inp.*read*()  s = s.*replace*('\n', '')  parse(s, 0)  out.*write*(xml\_to\_yaml(tags[0],0)) |

Результат:

|  |
| --- |
| schedule:  wednesday:  lesson1:  time: 08:20 - 09:50  subject: Информатика  lecturer: Балакшин Павел Валерьевич  room: Актовый зал (1216/0)  address: ул. Ломоносова, д.9, лит. М  lesson2:  time: 10:00 - 11:30  subject: Основы профессиональной деятельности  lecturer: Клименков Сергей Викторович  room: Актовый зал (1216/0)  address: ул. Ломоносова, д.9, лит. М  lesson3:  time: 15:20 - 16:50  subject: Английский язык B1.1  lecturer: Морозова Анна Михайловна  room: 3213  address: ул. Ломоносова, д.9, лит. Е  lesson4:  time: 17:00 - 18:30  subject: Английский язык B1.1  lecturer: Морозова Анна Михайловна  room: 3213  address: ул. Ломоносова, д.9, лит. Е  saturday:  lesson1:  time: 08:20 - 09:50  subject: Математический анализ (продвинутый уровень)  lecturer: Холодова Светлана Евгеньевна  room: 1405  address: Кронверкский пр., д.49, лит. А  lesson2:  time: 10:00 - 11:30  subject: Математический анализ (продвинутый уровень)  lecturer: Холодова Светлана Евгеньевна  room: 1405  address: Кронверкский пр., д.49, лит. А  lesson3:  time: 13:30 - 15:00  subject: Основы профессиональной деятельности  lecturer: Остапенко Иван Вадимович  room: 2116  address: Кронверкский пр., д.49, лит. А  lesson4:  time: 15:20 - 16:50  subject: Основы профессиональной деятельности  lecturer: Остапеко Иван Вадимович  room: 2116  address: Кронверкский пр., д.49, лит. А |

Сходства результатов выполнения кода 2 и 4 задания

Результат содержит одинаковые данные, одинаковую структуру.

Различия результатов выполнения кода 2 и 4 задания

Разный порядок тегов (в первом случае они отсортированы по алфавиту). Числовые значения в 4 задании, в отличие от 2, не обернуты в кавычки.

# Задание 5

Время парсинга+конвертации для 1 задания: 0.0010190010070800781 секунд.

Время парсинга+конвертации для 2 задания: 0.14359688758850098 секунд.

Время парсинга+конвертации для 3 задания: 0.008018732070922852 секунд.

Время парсинга+конвертации для 4 задания: 0.04431462287902832 секунд.

Анализ

Время выполнения 1-го и 3-го заданий не сильно отличаются потому что: в 1м задании используется построчный анализ xml файла, который быстро работает, а в 3 задании используется одно регулярное выражение и данные о тегах не сохраняются. Для 2го и 4го задания время отличается. Это может быть связано с тем, что готовая библиотека использует более сложные алгоритмы и хранение данных, чем код, написанный в 4 задании. 3 задание и 4 используют регулярные выражения, но в 3 задании не обрабатываются сложные структуры XML. В 4 задании обрабатываются атрибуты и одинаковые элементы, это добавляет дополнительную нагрузку.

# Заключение

В ходе лабораторной работы я изучила форматы представления данных. Научилась преобразовывать XML в yaml.

# Список использованных источников

Дейтел, П. Дж., Дейтел, Х., Нието, Т. Как программировать на XML / П. Дж. Дейтел, Х. Дейтел, Т. Нието. — М.: Бином, 2008. — 576 с.

Yaml – король мета-описаний // Хабр. — URL: https://habr.com/ru/articles/834270 (дата обращения: 27.11.2024).