Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники

Отчет
по лабораторной работе №4
«Исследование протоколов, форматов обмена информацией и языков разметки документов»

по дисциплине «Информатика»

вариант 35

Выполнил: Хоробрых Д.Е., группа P3116
Преподаватель: Машина Е.А.

Санкт-Петербург
~ 2022 ~

Оглавление

Задание

Вариант:

№ варианта	Исходный формат	Результирующий формат	День недели
35	XML	YAML	Суббота

1) Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый. Готовые библиотеки и регулярные выражения использовать нельзя.
2) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов. Переписать исходный код, применив найденные библиотеки. Регулярные выражения использовать нельзя. Сравнить полученные результаты и объяснить их сходство/различие.
3) Переписать исходный код, добавив в него использование регулярных выражений. Сравнить полученные результаты и объяснить их сходство/различие.
4) Используя свои программы из заданий 1–3, сравнить стократное время выполнения парсинга + конвертации в цикле. Проанализировать полученные результаты и объяснить их сходство/различие.
5) 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..

Задание 1

Код программы:

```
1. def parse_xml_to_yaml(filename: str):
2.     result = ''
3.     TAB = '  '
4.     with open(filename, 'r',  encoding='utf8') as f:
5.         text = f.read()
6.         now_tab_level = 0
7.         for line in map(str.strip, text.split('\n')):
8.             writing_tag = False
9.             now_tag = ''
10.            last_symbol = ''
11.            string = ''
12.            if line.startswith('<?'):
13.                continue
14.            if line.startswith('</'):
15.                now_tab_level -= 1
16.                continue
17.            for index, symbol in enumerate(line):
18.                if symbol == '<':
19.                    writing_tag = True
20.                elif last_symbol == '<' and symbol == '/':
21.                    result += TAB * now_tab_level + now_tag + ': ' + string
22.                    break
23.                elif symbol == '>':
24.                    writing_tag = False
25.                    if index == len(line) - 1:
26.                        result += TAB * now_tab_level + now_tag + ':'
27.                        now_tab_level += 1
28.                elif writing_tag:
29.                    now_tag += symbol
30.                else:
31.                    string += symbol
32.                last_symbol = symbol
33.            result += '\n'
34.        return result
35.
36.
37.with open('result.yaml', 'w', encoding='utf8') as f:
38.     f.write(parse_xml_to_yaml('day.xml'))
```

Содержание исходного файла day.xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
<schedule>
  <day>
    <day_date>сб, 29 октября</day_date>
    <lesson1>
      <time>
        <start>11:40</start>
        <end>13:10</end>
      </time>
      <body>
        <type>Лекция</type>
        <name>Математика (базовый уровень)</name>
        <info>
          <teacher>Правдин Константин Владимирович</teacher>
          <place>Ауд. 1404, Кронверкский пр., д.49, лит.А</place>
          <distant>Очный</distant>
        </info>
      </body>
    </lesson1>
    <lesson2>
      <time>
        <start>11:40</start>
        <end>13:10</end>
      </time>
      <body>
        <type>Лекция</type>
        <name>Математика (базовый уровень)</name>
        <info>
          <teacher>Правдин Константин Владимирович</teacher>
          <place>Ауд. 2201 (бывш. 206), Кронверкский пр., д.49,
лит.А</place>
          <distant>Очно-дистанционный</distant>
        </info>
      </body>
    </lesson2>
    <lesson3>
      <time>
        <start>13:30</start>
        <end>15:00</end>
      </time>
      <body>
        <type>Лекция</type>
        <name>Математика (базовый уровень)</name>
        <info>
          <teacher>Правдин Константин Владимирович</teacher>
          <place>Ауд. 1404, Кронверкский пр., д.49, лит.А</place>
          <distant>Очный</distant>
```

```
        </info>
      </body>
    </lesson3>
    <lesson4>
      <time>
        <start>13:30</start>
        <end>15:00</end>
      </time>
      <body>
        <type>Лекция</type>
        <name>Математика (базовый уровень)</name>
        <info>
          <teacher>Правдин Константин Владимирович</teacher>
          <place>Ауд. 2201 (бывш. 206), Кронверкский пр., д.49,
лит.А</place>
          <distant>Очно-дистанционный</distant>
        </info>
      </body>
    </lesson4>
    <lesson5>
      <time>
        <start>11:40</start>
        <end>13:10</end>
      </time>
      <body>
        <type>Лекция</type>
        <name>Математика (базовый уровень)</name>
        <info>
          <teacher>Правдин Константин Владимирович</teacher>
          <place>Ауд. 1404, Кронверкский пр., д.49, лит.А</place>
          <distant>Очный</distant>
        </info>
      </body>
    </lesson5>
  </day>
</schedule>
```

Содержание результирующего файла result.yaml:

```
schedule:
  day:
    day_date: сб, 29 октября
    lesson1:
      time:
        start: 11:40
        end: 13:10
      body:
        type: Лекция
```

```
      name: Математика (базовый уровень)
      info:
        teacher: Правдин Константин Владимирович
        place: Ауд. 1404, Кронверкский пр., д.49, лит.А
        distant: Очный
  lesson2:
    time:
      start: 11:40
      end: 13:10
    body:
      type: Лекция
      name: Математика (базовый уровень)
      info:
        teacher: Правдин Константин Владимирович
        place: Ауд. 2201 (бывш. 206), Кронверкский пр., д.49, лит.А
        distant: Очно-дистанционный
  lesson3:
    time:
      start: 13:30
      end: 15:00
    body:
      type: Лекция
      name: Математика (базовый уровень)
      info:
        teacher: Правдин Константин Владимирович
        place: Ауд. 1404, Кронверкский пр., д.49, лит.А
        distant: Очный
  lesson4:
    time:
      start: 13:30
      end: 15:00
    body:
      type: Лекция
      name: Математика (базовый уровень)
      info:
        teacher: Правдин Константин Владимирович
        place: Ауд. 2201 (бывш. 206), Кронверкский пр., д.49, лит.А
        distant: Очно-дистанционный
  lesson5:
    time:
      start: 11:40
      end: 13:10
    body:
      type: Лекция
      name: Математика (базовый уровень)
      info:
        teacher: Правдин Константин Владимирович
        place: Ауд. 1404, Кронверкский пр., д.49, лит.А
        distant: Очный
```

Задание 2

Для выполнения данного задания были выбраны следующие готовые библиотеки:

Xml_to_dict - для парсинга xml файла в словарь Python.

PyYAML - для конвертации словаря Python в формат YAML.

Исходный код программы:

```
1. from xml_to_dict import XMLtoDict
2. from yaml import dump
3.
4.
5. def task2_with_libs(filename: str):
6.     with open(filename) as f:
7.         dictionary = XMLtoDict().parse(f.read())
8.     with open('result.yaml', 'w') as f:
9.         dump(dictionary, f, allow_unicode=True)
10.
11.
12.task2_with_libs('day.xml')
```

Исходный и результирующий файлы аналогичны заданию 1.

По сравнению с первым вариантом выполнения задания, использование готовых сторонних библиотек позволило использовать намного меньше строк кода для реализации поставленной задачи, а также сократило время на создание и кода.

Задание 3

Для выполнения данного задания были произведены некоторые манипуляции с исходным файлом с помощью регулярных выражений.

Исходный код программы:

```
1. import re
2.
3.
4. def parser_xml_to_yaml_re(filename: str):
5.     with open(filename, encoding='utf8') as f:
6.         res = re.sub(re.compile(r'<\?.*>'), '', f.read())  # удаление
   служебных тегов
7.         res = re.sub(re.compile(r'</\w+>'), '', res)  # удаление
   закрывающихся тегов
8.         res = re.sub(re.compile(r'>'), r': ', res)  # приведение тегов к
   виду YAML
9.         res = re.sub(re.compile(r'<'), '', res)
10.        res = '\n'.join(filter(lambda x: x.strip(), res.split('\n')))  #
   удаление пустых строк
11.
12.    with open('result.yaml', 'w', encoding='utf-8') as f:
13.        f.write(res)
14.
15.
16.parser_xml_to_yaml_re('day.xml')
```

Исходный и результирующий файлы аналогичны заданию 1.

По сравнению с первым вариантом выполнения задания для реализации задачи с помощью регулярных выражений потребовалось также гораздо меньше строк кода, однако понадобилось время для продумывания оптимальных шаблонов и их написания.

Задание 4

В данном задании необходимо сравнить время выполнения трех вариантов программ при парсинге и конвертации 100 файлов в цикле.

Исходный код программы:

```
import datetime

from task1F import parse_xml_to_yaml
from task2 import task2_with_libs
from task3 import parser_xml_to_yaml_re

results = list()

start = datetime.datetime.now()
for i in range(100):
    parse_xml_to_yaml('day.xml')
results.append(datetime.datetime.now() - start)

start = datetime.datetime.now()
for i in range(100):
    task2_with_libs('day.xml')
results.append(datetime.datetime.now() - start)

start = datetime.datetime.now()
for i in range(100):
    parser_xml_to_yaml_re('day.xml')
results.append(datetime.datetime.now() - start)

print('Собственный код: ' + str(results[0].total_seconds()))
print('Использование библиотек: ' + str(results[1].total_seconds()))
print('Использование регулярных выражений: ' +
   str(results[2].total_seconds()))
```

Исходный и результирующий файлы аналогичны заданию 1.

Вывод программы:

```
Собственный код: 0.028428
Использование библиотек: 0.44911
Использование регулярных выражений: 0.048081
```

Итак, было получено, что наиболее быстрым вариантом парсинга и конвертации большого количества файлов является вариант программы, написанный без сторонних библиотек и регулярных выражений. Следующим по затратам времени оказался вариант с регулярными выражениями, а самым медленным (почти в 10 раз медленнее остальных) - вариант с использованием сторонних библиотек.

Данный результат являлся вполне ожидаемым, потому что внешние библиотеки не оптимизированы между собой и для реализации поставленной задачи необходимо переводить исходный файл в формат словаря Python.

Самым быстрым оказался собственный парсер, потому что в нем не используются никакие сторонние модули, только стандартные методы строк.

Задание 5

В данном задании необходимо написать программу для конвертации XML файла в любой другой (кроме JSON, YAML, XML, HTML) формат.

Мною был выбран формат CSV.

Исходный код программы:

```
import csv

from xml_to_dict import XMLtoDict


def task5(filename: str):
    with open(filename, 'r', encoding='utf8') as f:
        dictionary = XMLtoDict().parse(f.read())
        result = list()
        for key, value in dictionary.get('schedule').items():
            date = value['day_date']
            for k, v in value.items():
                if k == 'day_date':
                    date = v
                    continue
                start = v['time']['start']
                end = v['time']['end']
                type = v['body']['type']
                name = v['body']['name']
                teacher = v['body']['info']['teacher']
                place = v['body']['info']['place']
                distant = v['body']['info']['distant']

                result.append({'date': date,
                               'start': start,
                               'end': end,
                               'type': type,
                               'name': name,
                               'teacher': teacher,
                               'place': place,
                               'distant': distant})

    with open('result.csv', 'w', encoding='utf8') as f:
        d_writer = csv.DictWriter(f, result[0].keys())
        d_writer.writeheader()
        d_writer.writerows(result)


task5('day.xml')
```

Исходный файл аналогичен заданию 1.

Результирующий файл result.csv:

```
date,start,end,type,name,teacher,place,distant
"сб, 29 октября",11:40,13:10,Лекция,Математика (базовый уровень),Правдин
Константин Владимирович,"Ауд. 1404, Кронверкский пр., д.49, лит.А",Очный
"сб, 29 октября",11:40,13:10,Лекция,Математика (базовый уровень),Правдин
Константин Владимирович,"Ауд. 2201 (бывш. 206), Кронверкский пр., д.49,
лит.А",Очно-дистанционный
"сб, 29 октября",13:30,15:00,Лекция,Математика (базовый уровень),Правдин
Константин Владимирович,"Ауд. 1404, Кронверкский пр., д.49, лит.А",Очный
"сб, 29 октября",13:30,15:00,Лекция,Математика (базовый уровень),Правдин
Константин Владимирович,"Ауд. 2201 (бывш. 206), Кронверкский пр., д.49,
лит.А",Очно-дистанционный
"сб, 29 октября",11:40,13:10,Лекция,Математика (базовый уровень),Правдин
Константин Владимирович,"Ауд. 1404, Кронверкский пр., д.49, лит.А",Очный
```

Для решения данного варианта была использована библиотека из задания 2 для создания словаря Python и дальнейшего его парсинга в CSV.

Так для того, чтобы файл в формате XML конвертировать в формат CSV, необходимо, чтобы в файле XML описывалась одинаковая структура объектов, иначе перевод данного формата в CSV окажется бессмысленным.

Вывод

В ходе выполнения лабораторной работы я познакомился с разновидностями форматов обмена информацией и языков разметки, узнал о новых для себя форматах файлов, научился взаимодействовать с ними и реализовал несколько программ, позволяющих переводить файлы в XML формате в форматы YAML и CSV.

Список литературы

1) Лямин А.В., Череповская Е.Н. Объектно-ориентированное программирование. Компьютерный практикум. – СПб: Университет ИТМО, 2017. – 143 с. – Режим доступа: https://books.ifmo.ru/file/pdf/2256.pdf
2) Форма Бэкуса-Наура // Википедия - свободная энциклопедия URL: https://ru.wikipedia.org/wiki/Форма_Бэкуса_—_Наура (дата обращения: 23.10.2022).
3) Пишем изящный парсер на Питоне // Хабр URL: https://habr.com/ru/post/309242/ (дата обращения: 23.10.2022).