Университет ИТМО

Информатика

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

ФИО студента: Готовко Алексей Владимирович

Номер варианта: 13533 (номер ИСУ: 335151)

Направление подготовки: 09.03.04 (СППО)

Учебная группа: P3119

ФИО преподавателя: Балакшин Павел Валерьевич

Санкт-Петербург, 2021 г.

Оглавление

1. **Задание** 3
2. **Выполнение работы** 4
3. **Вывод** 8

1. Задание

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

1.2 Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.

1.3 Переписать исходный код, добавив в него использование регулярных выражений.

1.4 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.

1.5 Переписать исходную, чтобы она осуществляла парсинг и конвертацию исходного файла в любой другой формат (кроме JSON, YAML, XML, HTML): PROTOBUF, TSV, CSV, WML и т.п.

Вариант задания:

JSON – YAML - CSV

2. Выполнение работы

```python
# json2yaml converter (no libs, no regexes)
import time

def main():
  out = ["---\n"]
  flagNewBlock = False
  with open("timetable.json", 'r', encoding="utf-8") as f_in:
    line = f_in.readline()
    while line:
      # format tabulation and dashes
      if flagNewBlock:
        flagNewBlock = False
        tabs = line.count('\t')
        line = line.replace('\t', '')
        spaces = ' ' * (tabs - 2)
        line = spaces + '- ' + line
      else:
        line = line.replace('\t', ' ')
        if '{' in line:
          flagNewBlock = True
      # format symbols
      for c in "\'\",{}[]":
        line = line.replace(c, '')
      # format colons
      line = line.split(' ')
      for idx, word in enumerate(line):
        if len(word) < 3:
          continue
        flagNewLine, flagColon = False, False
        if word[-1] == '\n':
          flagNewLine = True
          word = word[:-1]
        if word[-1] == ':':
          flagColon = True
          word = word[:-1]
        if ':' in word:
          word = '\'' + word + '\''
          if flagColon:
            word += ':'
          if flagNewLine:
            word += '\n'
          line[idx] = word
      line = ' '.join(line)

      out.append(line)
      line = f_in.readline()

  for line in out:
    s = line.split()
    if not s:
      out.remove(line)

  with open("timetable.yaml", 'w', encoding="utf-8") as f_out:
    f_out.write(''.join(out))
```

```python
# json2yaml (with libs, no regexes)
import json, yaml

def main():
    with open("timetable.json", 'r', encoding="utf-8") as jsonFile:
        jsonData = json.load(jsonFile)
    with open("task1.yaml", 'w', encoding="utf-8") as yamlFile:
        yamlData = yaml.dump(jsonData, allow_unicode=True)
        yamlFile.write(yamlData)

```

```python
# json2yaml converter (no libs, with regexes)
import re

def main():
  out = ["---\n"]
  flagNewBlock = False
  with open("timetable.json", 'r', encoding="utf-8") as f_in:
    line = f_in.readline()
    while line:
      # format tabulation and dashes
      if flagNewBlock:
        flagNewBlock = False
        line = re.sub(r'\t+',  ' ' * (line.count('\t') - 2) + '- ', line)
      else:
        line = line.replace('\t', ' ')
        if '{' in line:
          flagNewBlock = True
      # format symbols
      line = re.sub(r'[\'\",{}\[\]]', '', line)
      # format colons
      line = line.split(' ')
      for idx, word in enumerate(line):
        if len(word) < 3:
          continue
        flagNewLine, flagColon = False, False
        if word[-1] == '\n':
          flagNewLine = True
          word = word[:-1]
        if word[-1] == ':':
          flagColon = True
          word = word[:-1]
        if ':' in word:
          word = '\'' + word + '\''
          if flagColon:
            word += ':'
          if flagNewLine:
            word += '\n'
          line[idx] = word
      line = ' '.join(line)

      out.append(line)
      line = f_in.readline()

  for line in out:
    s = re.sub(r'\s+', '', line)
    if not s:
      out.remove(line)

  with open("task2.yaml", 'w', encoding="utf-8") as f_out:
    f_out.write(''.join(out))
```

Проверка времени исполнения:

```
import time
import json2yaml_no_lib
import json2yaml_regex
import json2yaml_lib


start = time.process_time()
for i in range(100):
    open("timetable.yaml", 'w').close()
    json2yaml_no_lib.main()
print(f'No libs, no regex: {time.process_time() - start}')

start = time.process_time()
for i in range(100):
    open("task1.yaml", 'w').close()
    json2yaml_regex.main()
print(f'No libs, with regex: {time.process_time() - start}')

start = time.process_time()
for i in range(100):
    open("task2.yaml", 'w').close()
    json2yaml_lib.main()
print(f'With lib: {time.process_time() - start}')

```

Было использовано стократное, а не десятикратное, время выполнения для большей наглядности результата

```
No libs, no regex: 0.078125
No libs, with regex: 0.0625
With lib: 0.1875
```

Исходный файл (JSON)

```yaml
---
- Monday:
 - discipline: Дискретная математика (практика)
   lecturer: Поляков Владимир Иванович
   start_time: '11:40'
   end_time: '13:10'
   place: Кронверкский пр. д.49 лит.А
   lecture_room: 369А
   week_parity: четная неделя
   studying_format: Очно - дистанционный
 - discipline: Дискретная математика (практика)
   lecturer: Поляков Владимир Иванович
   start_time: '13:30'
   end_time: '15:00'
   place: Кронверкский пр. д.49 лит.А
   lecture_room: 369А
   week_parity: четная неделя
   studying_format: Очно - дистанционный
```

Полученные файлы (YAML и CSV):

```json
{
  "Monday": [
    {
      "discipline": "Дискретная математика (практика)",
      "lecturer": "Поляков Владимир Иванович",
      "start_time": "11:40",
      "end_time": "13:10",
      "place": "Кронверкский пр., д.49, лит.А",
      "lecture_room": "369А",
      "week_parity": "четная неделя",
      "studying_format": "Очно - дистанционный"
    },

    {
      "discipline": "Дискретная математика (практика)",
      "lecturer": "Поляков Владимир Иванович",
      "start_time": "13:30",
      "end_time": "15:00",
      "place": "Кронверкский пр., д.49, лит.А",
      "lecture_room": "369А",
      "week_parity": "четная неделя",
      "studying_format": "Очно - дистанционный"
    }
    ]
}
```

```csv
,Monday

0,"{'discipline': 'Дискретная математика (практика)', 'lecturer': 'Поляков Владимир Иванович', 'start_time': '11:40', 'end_time': '13:10', 'place': 'Кронверкский пр., д.49, лит.А', 'lecture_room': '369А', 'week_parity': 'четная неделя', 'studying_format': 'Очно - дистанционный'}"

1,"{'discipline': 'Дискретная математика (практика)', 'lecturer': 'Поляков Владимир Иванович', 'start_time': '13:30', 'end_time': '15:00', 'place': 'Кронверкский пр., д.49, лит.А', 'lecture_room': '369А', 'week_parity': 'четная неделя', 'studying_format': 'Очно - дистанционный'}"

```

3. Вывод

В процессе выполнения лабораторной работы удалось познакомиться с протоколами, форматами обмена информацией и языками разметки; реализовать парсер-конвертер между форматами JSON и YAML.

Список литературы

- Презентация лекции №4 *(https://isu.ifmo.ru/pls/apex/f?p=2002:0:121542816407093:DWNLD_F:NO::FILE:BC31BC6D79279D03C961166D5D910478)*
- Черновик методического пособия «Информатика» *(https://vk.com/doc-31201840_566998093)*