**Федеральное государственное автономное образовательное учреждение высшего образования**

**Национальный исследовательский университет ИТМО**

Факультет программной инженерии и компьютерной техники

Направление подготовки: 09.03.01, Информатика и вычислительная техника

Дисциплина «Информатика»

**Отчет**

**По лабораторной работе №4**

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

**Вариант №8**

Студент

Бушмелев Константин Алексеевич,   
группа P3132

Преподаватель

к.т.н. преподаватель Белозубов Александр Владимирович

г. Санкт-Петербург, 2022 г.

Оглавление

[Оглавление 2](#_Toc118876560)

[Задание 2](#_Toc118876561)

[Основное задание 3](#_Toc118876562)

[Дополнительное задание 1 3](#_Toc118876563)

[Дополнительное задание 2 3](#_Toc118876564)

[Дополнительное задание 3 3](#_Toc118876565)

[Дополнительное задание 4 3](#_Toc118876566)

[Заключение 3](#_Toc118876567)

[Список литературы 3](#_Toc118876568)

Задание

1. 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.
2. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.
3. 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.
4. 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. b) Сравнить полученные результаты и объяснить их сходство/различие.
5. 5. Дополнительное задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. a) Используя свою исходную программу из обязательного задания, программу из дополнительного задания №1 и программу из дополнительного задания №2, сравнить стократное время выполнения парсинга + конвертации в цикле.
   2. b) Проанализировать полученные результаты и объяснить их сходство/различие.
6. 6. Дополнительное задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. a) 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.
   2. b) Проанализировать полученные результаты, объяснить особенности использования формата.

Основное задание

import codecs  
import time  
  
start\_time1 = time.time()  
for i in range(100):  
 f = codecs.open("data/myItmo.json", "r", "utf-8")  
 s = ''''''  
 for v in f:  
 s += v  
 yaml = s.replace('{', '')  
 yaml = yaml.replace('}', '')  
 yaml = yaml.replace('"', '')  
 yaml = yaml.replace(',', '')  
 yaml = '---' + '\n' + yaml + '\n' + '---'  
end\_time1 = time.time()

Дополнительное задание 1

import codecs  
import json  
import yaml  
import time  
  
start\_time2 = time.time()  
for i in range(100):  
 f = codecs.open('data/myItmo.json', 'r', 'utf-8')  
 s = ''''''  
 for v in f:  
 s += v  
  
 json\_data = json.loads(s)  
 json\_data = json.dumps(json\_data)  
  
 yaml\_data = yaml.safe\_load(json\_data)  
 with open('data/myitmo.yaml', 'w', encoding='utf-8') as file:  
 yaml.dump(yaml\_data, file, allow\_unicode=True)  
 output = codecs.open("data/myitmo.yaml", "r", "utf-8")  
end\_time2 = time.time()

print(output.read())

Yaml файлы, сконвертированные в ходе выполнения заданий 1 и 2, сходны между собой конструкцией, вложенностью. Различие заключается в том, что в файле, сгенерированном с помощью библиотек, отсутствует открывающая и закрывающая конструкция “--- … ---”. В файле, сгенерированном самостоятельно, присутствуют лишние пустые строки.

Дополнительное задание 2

import codecs  
import re  
import time  
  
start\_time3 = time.time()  
for i in range(100):  
 f = codecs.open("data/myItmo.json", "r", "utf-8")  
 s = ''''''  
 for v in f:  
 s += v  
  
 new\_yaml = ''''''  
 k = 0 # количество табуляций  
 pattern = re.compile('\{[^{]\*?\{')  
 sfound = re.search(pattern, s)[0]  
 while True:  
 s = '{' + s.replace(sfound, '', 1)  
 while True:  
 try:  
 re.search('**\"**.+**\"**\s\*:\s\***\"**.+**\"**', sfound)[0]  
 except:  
 try:  
 key = re.search('**\"**[\w\d:-]+**\"**', sfound)[0]  
 new\_yaml += '\n' + '\t' \* k + key.replace("\"", "") + ':'  
 finally:  
 break  
 else:  
 key = re.search('**\"**[\w\d:-]+**\"**', sfound)[0]  
 new\_yaml += '\n' + '\t' \* k + key.replace("\"", "") + ':'  
 sfound = sfound.replace(key, '', 1)  
 value = re.search('**\"**.+**\"**', sfound)[0]  
 new\_yaml += value.replace("\"", "")  
 sfound = sfound.replace(value, '', 1)  
 try:  
 sfound = re.search(pattern, s)[0]  
 k = s.count('}') - s.count('{')  
 except:  
 break  
  
while True:  
 try:  
 re.search('**\"**.+**\"**\s\*:\s\***\"**.+**\"**', s)[0]  
 except:  
 break  
 else:  
 key = re.search('**\"**[\w\d:-]+**\"**', s)[0]  
 new\_yaml += '\n' + '\t' \* k + key.replace("\"", "") + ':'  
 s = s.replace(key, '', 1)  
 value = re.search('**\"**.+**\"**', s)[0]  
 new\_yaml += value.replace("\"", "")  
 s = s.replace(value, '', 1)  
end\_time3 = time.time()  
print(new\_yaml)

Первая и третья реализации парсинга и конвертации различаются. В первой нужный формат получается посредством удаления лишних символов, характерных для синтаксиса json формата, тогда как в третьей Yaml файл формируется построчно, одновременно с обработкой входного файла.

Дополнительное задание 3

from JSONToYAML import start\_time1, end\_time1  
  
def TimeToTask(start, end):  
 return (end - start) \* 100  
t1 = TimeToTask(start\_time1, end\_time1)  
print("Время выполнение первой программы: ", t1)  
  
from Dop1 import start\_time2, end\_time2  
  
t2 = TimeToTask(start\_time2, end\_time2)  
print("Время выполнение второй программы: ", t2)  
  
from Dop2 import start\_time3, end\_time3  
  
t3 = TimeToTask(start\_time3, end\_time3)  
print("Время выполнение третьей программы: ", t3)

Изображение выглядит как текст

Автоматически созданное описание

Время выполнения существенно различается. Первая работает быстрее всего, так как она использует только метод замены строк. Вторая работает заметно дольше, так как в библиотеки включена обработка однострочных и многострочных файлов, наследование и так далее, то есть ее возможности существенно шире.

Дополнительное задание 4

import codecs  
import json  
import re  
  
f = codecs.open("data/myItmo.json", "r", "utf-8")  
s = ''''''  
for v in f:  
 s += v  
jsondata = json.loads(s)  
csvdata = ''''''  
csvdatakey = []  
csvdatavalue = []  
k = 0 # количество табуляций  
pattern = re.compile('\{[^{]\*?\{')  
sfound = re.search(pattern, s)[0]  
while True:  
 s = '{' + s.replace(sfound, '', 1)  
 while True:  
 try:  
 re.search('**\"**.+**\"**\s\*:\s\***\"**.+**\"**', sfound)[0]  
 except:  
 break  
 else:  
 key = re.search('**\"**[\w\d:-]+**\"**', sfound)[0]  
 csvdatakey.append(key)  
 sfound = sfound.replace(key, '', 1)  
 value = re.search('**\"**.+**\"**', sfound)[0]  
 csvdatavalue.append(value)  
 sfound = sfound.replace(value, '', 1)  
 try:  
 sfound = re.search(pattern, s)[0]  
 except:  
 break  
while True:  
 try:  
 re.search('**\"**.+**\"**\s\*:\s\***\"**.+**\"**', s)[0]  
 except:  
 break  
 else:  
 key = re.search('**\"**[\w\d:-]+**\"**', s)[0]  
 csvdatakey.append(key)  
 s = s.replace(key, '', 1)  
 value = re.search('**\"**.+**\"**', s)[0]  
 csvdatavalue.append(value)  
 s = s.replace(value, '', 1)  
  
csvdatakey2 = ['"NumberOfLesson"']  
csvdatavalue2 = [['"1"'], ['"2"'], ['"3"'], ['"4"'], ['"5"'], ['"6"']]  
for v in csvdatakey[::-1]:  
 if v not in csvdatakey2:  
 csvdatakey2.append(v)  
csvdatakey2 = [csvdatakey2[0]] + csvdatakey2[::-1][:3] + csvdatakey2[1:-3][::-1]  
for i in range(6):  
 k = 0  
 remove = []  
 for j in range(8):  
 try:  
 csvdatavalue2[i].append(csvdatavalue[j])  
 except:  
 break  
 k += 1  
 if k > 3: remove.append(csvdatavalue[j])  
 for g in range(5):  
 try: csvdatavalue.remove(remove[g])  
 except: break  
for v in csvdatakey2:  
 csvdata += v + ';'  
csvdata += '\n'  
for v in csvdatavalue2:  
 for i in v:  
 csvdata += i + ';'  
 csvdata += '\n'  
  
print(csvdata)

Csv файл – текстовый файл, каждая строка в котором является строкой таблицы, а столбцы отделены друг от друга специальными символами(разделителями, например, “;”). Csv файлы представляются в виде таблицы.

Изображение выглядит как текст

Автоматически созданное описание

Заключение

В ходу лабораторной работы я познакомился с некоторыми языками разметки, узнал о библитоеках Python, работающих с ними, разобрался с устройством HTML файлов, применил регулярные выражения для парсинга и конвертации файлов.

# Список литературы

[В Интернете] // Википедия. - 4 апрель 2016 г.. - https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0\_%D0%91%D1%8D%D0%BA%D1%83%D1%81%D0%B0\_%E2%80%94\_%D0%9D%D0%B0%D1%83%D1%80%D0%B0.

**saluev** Пишем изящный парсер на Питоне [В Интернете]. - 5 сентябрь 2016 г.. - https://habr.com/ru/post/309242/.

**Орлов С. А. Цилькер Б. Я.** Арифметические основы вычислительных машин [Книга]. - Санкт-Петербург : [б.н.], 2011. - Т. 2.