Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 4**

по дисциплине

‘ИНФОРМАТИКА’

Вариант №21

*Выполнил:*

Студент группы P3109

Mallev Sabur N.

*Преподаватель:*

Ильина Аглая

Геннадьевна



Санкт-Петербург, 2021

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. Дополнительное задание задание No1 (позволяет набрать +10 процентов   
от максимального числа баллов БаРС за данную лабораторную).   
a) Найти готовые библиотеки, осуществляющие аналогичный   
парсинг и конвертацию файлов.   
b) Переписать исходный код, применив найденные библиотеки.   
Регулярные выражения также нельзя использовать.   
c) Сравнить полученные результаты и объяснить их   
сходство/различие.   
3. Дополнительное задание задание No2 (позволяет набрать +10 процентов   
от максимального числа баллов БаРС за данную лабораторную).   
a) Переписать исходный код, добавив в него использование   
регулярных выражений.   
b) Сравнить полученные результаты и объяснить их   
сходство/различие.   
4. Дополнительное задание задание No3 (позволяет набрать +10 процентов   
от максимального числа баллов БаРС за данную лабораторную).

a) Используя свою исходную программу из обязательного   
задания, программу из дополнительного задания No1 и   
программу из дополнительного задания No2, сравнить   
десятикратное время выполнения парсинга + конвертации в   
цикле.   
b) Проанализировать полученные результаты и объяснить их   
сходство/различие.   
5. Дополнительное задание задание No4 (позволяет набрать +5 процентов   
от максимального числа баллов БаРС за данную лабораторную.   
a) 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.   
b) Проанализировать полученные результаты, объяснить   
осоебнности использованного формата.

Ход работы

Изначальный файл

{

"weekday": "Чт",

"lessons": [

{

"time": "08:20-09:50",

"place": "ул.Ломоносова, д.9, лит. М",

"room": "1512 ауд.",

"name": "Математика(Прак)",

"groups": "15",

"format": "Очно - дистанционный ",

"teacher": "Холодова Светлана Евгеньевна"

},

{

"time": "11:40-13:10",

"place": "ул.Ломоносова, д.9, лит. М",

"room": "1329 ауд.",

"name": "Математика(Прак)",

"groups": "2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17",

"format": "Очно - дистанционный ",

"teacher": "Холодова Светлана Евгеньевна"

},

{

"time": "13:30-15:00",

"place": "ул.Ломоносова, д.9, лит. М",

"room": "1222 ауд.",

"name": "Математика(Лек)",

"groups": "2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17",

"format": "Очно - дистанционный ",

"teacher": "Холодова Светлана Евгеньевна"

}

]

}

\*Добавлены переносы строки для простоты чтения.

Задание 1

#var 21

#JSON to YAML

def newLine(result, depth, isList = False):

if not isList:

result += "\n" + " " \* depth

else:

result += "\n" + " " \* (depth - 1) + "- "

return result

def convert():

f = open("schedule.json", encoding='unicode\_escape')

text = f.read()

f.close()

writing = False

element = True

inList = False

depth = -1

result = ""

for i in text:

if writing:

if i == '"':

writing = not writing

if not element:

result = newLine(result, depth)

else:

result += ": "

element = not element

continue

result += i

else:

if i == '"':

writing = True

elif i == "{":

depth += 1

element = True

if inList:

result = newLine(result, depth, True)

elif i == "}":

depth -= 1

elif (i == "["):

inList = True

elif i == "]":

inList = False

f = open("schedule.yml", mode="w")

f.write(result)

f.close()

if \_\_name\_\_ == '\_\_main\_\_':

convert()

Результат

weekday: Чт

lessons:

- time: 08:20-09:50

place: ул.Ломоносова, д.9, лит. М

room: 1512 ауд.

name: Математика(Прак)

groups: 15

format: Очно - дистанционный

teacher: Холодова Светлана Евгеньевна

- time: 11:40-13:10

place: ул.Ломоносова, д.9, лит. М

room: 1329 ауд.

name: Математика(Прак)

groups: 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17

format: Очно - дистанционный

teacher: Холодова Светлана Евгеньевна

- time: 13:30-15:00

place: ул.Ломоносова, д.9, лит. М

room: 1222 ауд.

name: Математика(Лек)

groups: 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17

format: Очно - дистанционный

teacher: Холодова Светлана Евгеньевна

Задание 2

Для интерпретирования JSON файлов используем встроенную библиотеку json, а для создания YAML файла используем библиотеку [PyYAML](https://pyyaml.org/wiki/PyYAMLDocumentation).

#var 21

#JSON to YAML

import yaml

import json

def convert():

f = open("schedule.json", encoding='unicode\_escape')

text = json.loads(f.read())

f.close()

text = yaml.dump(text)

f = open("librarySchedule.yml", mode="w")

f.write(text)

f.close()

if \_\_name\_\_ == '\_\_main\_\_':

convert()

Результат

lessons:

- format: "\u041E\u0447\u043D\u043E - \u0434\u0438\u0441\u0442\u0430\u043D\u0446\u0438\

\u043E\u043D\u043D\u044B\u0439 "

groups: '15'

name: "\u041C\u0430\u0442\u0435\u043C\u0430\u0442\u0438\u043A\u0430(\u041F\u0440\

\u0430\u043A)"

place: "\u0443\u043B.\u041B\u043E\u043C\u043E\u043D\u043E\u0441\u043E\u0432\u0430\

, \u0434.9, \u043B\u0438\u0442. \u041C"

room: "1512 \u0430\u0443\u0434."

teacher: "\u0425\u043E\u043B\u043E\u0434\u043E\u0432\u0430 \u0421\u0432\u0435\u0442\

\u043B\u0430\u043D\u0430 \u0415\u0432\u0433\u0435\u043D\u044C\u0435\u0432\u043D\

\u0430"

time: 08:20-09:50

- format: "\u041E\u0447\u043D\u043E - \u0434\u0438\u0441\u0442\u0430\u043D\u0446\u0438\

\u043E\u043D\u043D\u044B\u0439 "

groups: 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17

name: "\u041C\u0430\u0442\u0435\u043C\u0430\u0442\u0438\u043A\u0430(\u041F\u0440\

\u0430\u043A)"

place: "\u0443\u043B.\u041B\u043E\u043C\u043E\u043D\u043E\u0441\u043E\u0432\u0430\

, \u0434.9, \u043B\u0438\u0442. \u041C"

room: "1329 \u0430\u0443\u0434."

teacher: "\u0425\u043E\u043B\u043E\u0434\u043E\u0432\u0430 \u0421\u0432\u0435\u0442\

\u043B\u0430\u043D\u0430 \u0415\u0432\u0433\u0435\u043D\u044C\u0435\u0432\u043D\

\u0430"

time: 11:40-13:10

- format: "\u041E\u0447\u043D\u043E - \u0434\u0438\u0441\u0442\u0430\u043D\u0446\u0438\

\u043E\u043D\u043D\u044B\u0439 "

groups: 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17

name: "\u041C\u0430\u0442\u0435\u043C\u0430\u0442\u0438\u043A\u0430(\u041B\u0435\

\u043A)"

place: "\u0443\u043B.\u041B\u043E\u043C\u043E\u043D\u043E\u0441\u043E\u0432\u0430\

, \u0434.9, \u043B\u0438\u0442. \u041C"

room: "1222 \u0430\u0443\u0434."

teacher: "\u0425\u043E\u043B\u043E\u0434\u043E\u0432\u0430 \u0421\u0432\u0435\u0442\

\u043B\u0430\u043D\u0430 \u0415\u0432\u0433\u0435\u043D\u044C\u0435\u0432\u043D\

\u0430"

time: 13:30-15:00

weekday: "\u0427\u0442"

Отличия

* Все символы кириллицы заменены на Unicode escape characters, так как кириллица не является частью ASCII
* Порядок многих элементов изменен
* Некоторые строки закрыты кавычками

Задание 4

import time

import lab3task0

import lab3task1

print("Now testing my interpreter")

start\_time = time.time()

for i in range(10):

lab3task0.convert()

print("--- %s seconds ---" % (time.time() - start\_time))

print("Now testing some other library")

start\_time = time.time()

for i in range(10):

lab3task1.convert()

print("--- %s seconds ---" % (time.time() - start\_time))

Результат

Время выполнения моего кода: 0.0049860477447509766 сек.

Время выполнения кода с библиотеками: 0.026281118392944336 сек.

Мой код работает быстрее, на что есть несколько причин:

* Нет промежуточного шага, когда JSON сериализуется.
* Библиотеки предназначены для работы с многими различными случаями, в то время как мой код скорее всего не будет работать ни на чем, кроме моего одного конкретного кейса.

Задание 5

Выбранный формат: TSV (Tab separated values)

#var 21

#JSON to TSV

def convert():

f = open("schedule.json", encoding='unicode\_escape')

text = f.read()

f.close()

writing = False

element = True

result = {}

realResult = ""

keys = []

longest = 0

elementName = ""

value = ""

for i in text:

if writing:

if i == '"':

writing = not writing

if not element:

result[elementName].append(value)

if len(result[elementName]) > longest:

longest = len(result[elementName])

elementName = ""

else:

if elementName not in result:

result[elementName] = []

keys.append(elementName)

value = ""

element = not element

continue

if not element:

value += i

else:

elementName += i

else:

if i == '"':

writing = True

elif i == "{":

element = True

elementName = ""

realResult = "\t".join(keys) + "\n"

for i in range(longest):

for key in keys:

if (i < len(result[key])):

realResult += result[key][i]

realResult += "\t"

realResult += "\n"

f = open("schedule.tsv", mode="w")

f.write(realResult)

f.close()

if \_\_name\_\_ == '\_\_main\_\_':

convert()

Результат



Иллюстрация 1: Результат задание 5 загруженный в Google Docs Таблицы

Особенности

TSV преимущество используется для таблиц и в нем тяжело задавать массивы и объекты. Я поместил все ключи в первую строку, а их значения в столбик под ними.

Выводы

YAML очень простой и понятный формат для понимания человеком, но не компьютером. А форматы для таблиц лучше оставить таблицам, а не пытаться запарсить в них JSON.

Список литературы

YAML – Wikipedia [https HYPERLINK "https://en.wikipedia.org/wiki/YAML":// HYPERLINK "https://en.wikipedia.org/wiki/YAML"en HYPERLINK "https://en.wikipedia.org/wiki/YAML". HYPERLINK "https://en.wikipedia.org/wiki/YAML"wikipedia HYPERLINK "https://en.wikipedia.org/wiki/YAML". HYPERLINK "https://en.wikipedia.org/wiki/YAML"org HYPERLINK "https://en.wikipedia.org/wiki/YAML"/ HYPERLINK "https://en.wikipedia.org/wiki/YAML"wiki HYPERLINK "https://en.wikipedia.org/wiki/YAML"/ HYPERLINK "https://en.wikipedia.org/wiki/YAML"YAML](https://en.wikipedia.org/wiki/YAML)

What is a YAML file? – Initial Commit <https://initialcommit.com/blog/what-is-a-yaml-file>