

Санкт-Петербургский Национальный Исследовательский
Университет Информационных технологий, Механики и Оптики
Факультет Программной Инженерии и Компьютерных Технологий



Вариант №15
Лабораторная работа №4
По дисциплине информатика

Выполнил:

Студент группы: Р3111

Кривоносов Егор Дмитриевич

Преподаватель:

Балакшин П.В.

Санкт-Петербург

2019 г.

Краткое описание задачи:

Написать парсер с YAML в JSON с помощью Python.

1. Определить номер варианта как остаток деления на 35 порядкового номера в списке группы в ISU. В случае, если в данный день недели нет занятий, то увеличить номер варианта на восемь.
2. Изучить форму Бэкуса-Наура.
3. Изучить особенности языков разметки/форматов JSON, YAML, XML, PROTOBUF.
4. Понять устройство страницы с расписанием для своей группы:
<http://www.ifmo.ru/ru/schedule/0/P3200/schedule.htm>
5. Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного.
6. Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый.
7. Нельзя использовать готовые библиотеки, кроме re (регулярные выражения в Python) и библиотеки для загрузки XML-файлов.
8. Необязательное задание для получения оценки «4» и «5» (позволяет набрать от 75 до 89 процентов от максимального числа баллов BaPC за данную лабораторную).
 - а) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
 - б) Сравнить полученные результаты и объяснить их сходство/различие.
9. Необязательное задание для получения оценки «5» (позволяет набрать от 90 до 100 процентов от максимального числа баллов BaPC за данную лабораторную).
 - а) Используя свою программу и найденные готовые

библиотеки, сравнить десятикратное время выполнения парсинга + конвертации в цикле.

b) Проанализировать полученные результаты и объяснить их сходство/различие.

10. Проверить, что все пункты задания выполнены и выполнены верно.

Решение:

Файл start.yml :

Среда:

Расписание:

Пара1:

Время: "10:00-11:30"

Ауд: "2219 АУД."

Корпус: "Ломоносова, д.9, лит. А"

Предмет: "МАТЕМАТИКА (БАЗОВЫЙ УРОВЕНЬ) (ЛЕК)"

Преподаватель: "Холодова Светлана Евгеньевна"

Пара2:

Время: "11:40-13:10"

Ауд: " "

Корпус: "Ломоносова, д.9, лит. Е"

Предмет: "ИНОСТРАННЫЙ ЯЗЫК (ПРАК)"

Преподаватель: " "

Пара3:

Время: "13:30-15:00"

Ауд: "1121 АУД."

Корпус: "Ломоносова, д.9, лит. М"

Предмет: "МАТЕМАТИКА (БАЗОВЫЙ УРОВЕНЬ) (ПРАК)"

Преподаватель: "Мелихова Алина Семеновна"

Файл end.json (изначально он пустой):

Код:

```
import sys
```

```
def gg():
```

```
    return print("Не, не, не...\nЯ так работать не буду, иди сначала файл заполни  
правильно, а потом запускай!")
```

```
def vremia():
```

```
    return  
(time.time()+time.time()+time.time()+time.time()+time.time()+time.time()+time.time()  
+time.time()+time.time()+time.time())/10
```

```
inputfile = open(r'C:\Users\Egor\Desktop\start.yaml', 'r')
```

```
outputfile = open(r'C:\Users\Egor\Desktop\end.json', 'w')
```

```
newline = inputfile.readline()
```

```
data = list()
```

```
lines = 0
```

```
closethink = list()
```

```
helpstring = list()
```

```
while newline:
```

```
    data.append(newline)
```

```
    lines += 1
```

```
    newline = inputfile.readline()
```

```
inputfile.close()
```

```
start_k = len(data[0]) - len(data[0].lstrip())
```

```
outputfile.write("{\n")
```

```
start_time = vremia()
```

```

for i in range(0, lines-1):
    end_k = len(data[i + 1]) - len(data[i + 1].lstrip())
    if data[i] in
["Понедельник:\n", "Вторник:\n", "Среда:\n", "Четверг:\n", "Пятница:\n", "Суббота:\n"
]:
    if data[0] == data[i]:
        helpstring = data[i].lstrip().split(':', maxsplit = 1)
        outputfile.write('"' + helpstring[0] + "':" + helpstring[1].lstrip())
    else:
        helpstring = data[i].lstrip().split(':', maxsplit = 1)
        outputfile.write(' } \n } \n' + helpstring[0] + "':" + helpstring[1].lstrip())
elif data[i] in [" Расписание:\n"]:
    helpstring = data[i].lstrip().split(':', maxsplit = 1)
    try:
        outputfile.write('\t"' + helpstring[0] + "':" + helpstring[1].lstrip())
    except IndexError:
        gg()
        sys.exit()
elif data[i] in [" Папа1: \n", " Папа2:\n", " Папа3:\n", " Папа4:\n", " Папа5:\n", "
Папа6:\n"]:
    helpstring = data[i].lstrip().split(':', maxsplit = 1)
    try:
        outputfile.write("\t\t'" + helpstring[0] + "':" + helpstring[1].lstrip())
    except IndexError:
        gg()
        sys.exit()
else:
    helpstring = data[i].lstrip().split(':', maxsplit = 1)
    gg = helpstring[1].split("\n")
    try:

```

```
        outputfile.write("\t\t\t" + helpstring[0] + ": ' + gg[0].lstrip()+",\n")
except IndexError:
    gg()
    sys.exit()
if end_k < start_k:
    outputfile.write(" \t\t\t", "\n')
```

Результат, файл end.json:

```
{
  "Среда":
    {
      "Расписание":
        {
          "Пара1":
            {
              "Время": "10:00-11:30" ,
              "Ауд": "2219 АУД.",
              "Корпус": "Ломоносова, д.9, лит. А",
              "Предмет": "МАТЕМАТИКА (БАЗОВЫЙ УРОВЕНЬ) (ЛЕК)",
              "Преподаватель": "Холодова Светлана Евгеньевна",
            },
          "Пара2":
            {
              "Время": "11:40-13:10",
              "Ауд": " ",
              "Корпус": "Ломоносова, д.9, лит. Е",
              "Предмет": "ИНОСТРАННЫЙ ЯЗЫК (ПРАК)",
              "Преподаватель": " ",
            },
          "Пара3":
            {
              "Время": "13:30-15:00",
              "Ауд": "1121 АУД.",
              "Корпус": "Ломоносова, д.9, лит. М",
              "Предмет": "МАТЕМАТИКА (БАЗОВЫЙ УРОВЕНЬ) (ПРАК)",
            },
        }
    }
}
```


Вывод:

В ходе проделанной работы я научился преобразовывать YAML в JSON с помощью Python. Ещё раз убедился, что python более удобен для начинающих программистов.

Так же я сравнил библиотеку ruyaml и свой парсер, на время выполнения данной задачи. Оказалось, что ruyaml выполнил свою работу за 0.003014087677001953 секунд, а моя программа сделала это за 0.005534014962769000 секунд. Это может быть связано с тем, что запуск программ производился на разной мощности ПК и из-за этого немного различны результаты, так же я запустил свой код на helios , там моя программы сделала преобразование всего за 0.003014087677001953 секунд . Ещё заметил различия после обработки библиотекой ruyaml, что она не воспринимает русские слова/символы и у неё по-другому устанавливаются { и } (но это не существенно, только для восприятия глаза).