**Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики.**

Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №3

предмет «Информатика» вариант 28

ФИО: Фам Мань Туан

Группа: P3112

Преподаватель: Павел Балакшин

Малышева Татьяна Алексеевна

**Санкт-Петербург 2020 г.**

**Цель задания:**

Овладеть знаниями о различных современных языках разметки документов и форматах данных, навыками обработки данных с помощью языка Python 3.x.

**Задание:**

1. Изучить форму Бэкуса-Наура.

2. Изучить особенности языков разметки/форматов JSON, YAML, XML, PROTOBUF.

3. Понять устройство страницы с расписанием для своей группы: http://www.ifmo.ru/ru/schedule/0/P3200/schedule.htm.

4. Исходя из структуры расписания конкретного дня сформировать файл с расписанием в формате, указанном в задании в качестве исходного.

5. Написать программу на языке Python, которая бы осуществляла парсинг и конвертацию исходного файла в новый. НЕЛЬЗЯ использовать готовые библиотеки, кроме re (регулярные выражения в Python)!

6. Номер варианта определить как остаток деления на 21 порядкового номера в списке группы в ISU.

7. Написать вывод по итогам выполнения лабораторной работы.

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

**Вариант: 28**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ варианта** | **Исходный формат** | **Результирующий формат** | **День недели** |
| 28 | JSON | XML | Пятница |

**Исходный формат:**

{

"urok1":

{

"day": "Пт",

"time": "15:20-16:50",

"lesson": "Дискретная Математика (Лек)",

"teacher": "Поляков Владимир Иванович",

"room": "466",

"location": "Кронверкский пр., д.49, лит.А",

"note": "Только 11.09.,25.09.",

"parity": false

}

}

**Код программы:**

import re

import time

start = time.perf\_counter()

class Urok:

def \_\_init\_\_(self):

self.day = None

self.time = None

self.room = None

self.lesson = None

self.teacher = None

self.location = None

self.parity = None

self.note = None

class Parser:

def excute(self, source):

uroki = [Urok(), Urok(), Urok(), Urok(), Urok(), Urok(), Urok(), Urok(), Urok(), Urok()]

lines = source.readlines()

count = 0

for line in lines:

line = line.replace("\t", "")

line = line.replace("\n", "")

if re.search(":", line):

if re.search(r"\"urok\d\":",line):

count += 1

else:

uroki[count-1] = self.parserline(uroki[count-1], line)

return uroki

def parserline(self, urok, line):

temp = str(line).split(": ")

key = re.search("\".\*\"",temp[0]).group(0).replace("\"","")

if re.search("\".\*\"",temp[1]):

value = re.search("\".\*\"",temp[1]).group(0).replace("\"","")

else:

value = temp[1].replace(",","")

if key == "day":

urok.day = value

elif key == "time":

urok.time = value

elif key == "room":

urok.room = value

elif key == "lesson":

urok.lesson = value

elif key == "teacher":

urok.teacher = value

elif key == "location":

urok.location = value

elif key == "parity":

urok.parity = value

else:

urok.note = value

return urok

def printxml(self,uroki):

xml = "<timetable>\n"

j=0

for i in range(len(uroki)):

j+=1

if uroki[i].day is not None:

xml += "\t<urok{}>\n".format(j)

dict = uroki[i].\_\_dict\_\_

for dict\_element in dict:

xml += "\t\t<{}>{}</{}>\n".format(dict\_element, dict[dict\_element], dict\_element)

xml += "\t</urok{}>\n".format(j)

xml += "</timetable>"

return xml

input = open(r"C:\Users\the7u\PycharmProjects\lab3\_info\timetable.json", "r", encoding="utf-8")

parser = Parser()

output = open(r"C:\Users\the7u\PycharmProjects\lab3\_info\timetable.xml", "w", encoding="utf-8")

output.write(parser.printxml(parser.excute(input)))

input.close()

output.close()

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

**Результат**

---0.0008559999999999957 seconds---

<timetable>

<urok1>

<day>Пт</day>

<time>15:20-16:50</time>

<room>466</room>

<lesson>Дискретная Математика (Лек)</lesson>

<teacher>Поляков Владимир Иванович</teacher>

<location>Кронверкский пр., д.49, лит.А</location>

<parity>false</parity>

<note>Только 11.09.,25.09.</note>

</urok1>

</timetable>

**Конвертируйте файлы с помощью доступных библиотек и сравните:**

Используемые библиотеки: <https://pypi.org/project/json2xml/>

Код программы:

from json2xml import json2xml

from json2xml.utils import readfromurl, readfromstring, readfromjson

import time

start = time.perf\_counter()

data = readfromstring('{"urok1":{"day":"Пт","time": "15:20-16:50", "lesson": "Дискретная Математика (Лек)","teacher":"Поляков Владимир Иванович","room": "466","location": "Кронверкский пр., д.49, лит.А","note": "Только 11.09.,25.09.","parity": false}}')

print(json2xml.Json2xml(data).to\_xml())

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

Результат

<?xml version="1.0" ?>

<all>

<urok1 type="dict">

<day type="str">Пт</day>

<time type="str">15:20-16:50</time>

<lesson type="str">Дискретная Математика (Лек)</lesson>

<teacher type="str">Поляков Владимир Иванович</teacher>

<room type="str">466</room>

<location type="str">Кронверкский пр., д.49, лит.А</location>

<note type="str">Только 11.09.,25.09.</note>

<parity type="bool">False</parity>

</urok1>

</all>

---0.0034719 seconds---

* Мой исходный код выполняется значительно быстрее, чем доступная библиотека, благодаря разметке файла JSON и спецификации входных данных. Но также по той же причине мой исходный код не может согласиться на файл JSON с неизвестными данными и случайным расположением.

**Вывод**

В этой лабораторной работе я познакомился с принципами программирования ООП в Python и принципами чтения и записи языков разметки XML, JSON и YAML.