Санкт-Петербургский Национальный Исследовательский Университет ИнформационныхТехнологий,Механики и ОптикиФакультетПрограммной Инженерии и КомпьютернойТехники



Вариант № 18

Лабораторная работа № 4

по дисциплине

’Информатика’

Выполнил:

Ортис Хосе

288867;

P3132;

Преподаватель:

Калинин Игорь

Владимирович

Санкт-Петербург 2019 г.

**Текст задания:**

Лабораторная работа No4. «Исследование языков разметки документов»

Порядок выполнения работы

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 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый.
6. Нельзя использовать готовые библиотеки, кроме re (регулярные выражения в Python) и библиотеки для загрузки XML-файлов.
7. Проверить, что все пункты задания выполнены и выполнены верно.
8. Написать отчёт о проделанной работе.
9. Подготовиться к устным вопросам на защите.

**Необязательное задание для получения оценки «4» и «5»** (позволяет набрать от 75 до 89 процентов от максимального числа баллов БаРС за данную лабораторную).

1. Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
2. Сравнить полученные результаты и объяснить их сходство/различие.

**Необязательное задание для получения оценки «5»** (позволяет набрать от 90 до 100 процентов от максимального числа баллов БаРС за данную лабораторную).

1. Используя свою программу и найденные готовые библиотеки, сравнить десятикратное время выполнения парсинга + конвертации в цикле.
2. Проанализировать полученные результаты и объяснить их сходство/различие.

Installation library

sudo pip install xlmplain

<https://pypi.org/project/xmlplain/>

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Calculating Time*  import time  start\_time = time.time()  main()  ...  print("%.2f" % (time.time() - start\_time))  *libConverter.py*  import xmlplain  # Read the YAML file  with open("input.yaml") as inf:  root = xmlplain.obj\_from\_yaml(inf)  # Output back XML  with open("output-lib.xml", "w") as outf:  xmlplain.xml\_from\_obj(root, outf, pretty=True) | Results of testing   |  |  | | --- | --- | | MyProgram [s] | Library [s] | | 0.00063324 | 0.00470591 | | 0.00055003 | 0.00614500 | | 0.00054145 | 0.00519085 | | 0.00055575 | 0.00655007 | | 0.00064850 | 0.00514770 | | 0.00061512 | 0.00513721 | | 0.00061488 | 0.00612378 | | 0.00067711 | 0.00477624 | | 0.00060797 | 0.00504160 | | 0.00066900 | 0.00489211 | |

**Программа:**

*main.py*

import re

inputFile = open("input.yaml", "r", encoding="utf-8")

inputArr = list()

closeTagsArr = list()

outputArr = list()

for line in inputFile:

numTabs = 0

for letter in line:

if letter != " ":

break

numTabs += 1

inputArr.append([numTabs // 4, line.lstrip().lower()])

inputFile.close()

def addTabs(numTabs):

return "\t" \* numTabs

for i, elem in enumerate(inputArr):

numTabs = elem[0]

strElem = elem[1]

# if is parent (only tag)

if re.match(".\*:\\n", strElem):

outputArr.append(addTabs(numTabs) + "<" + strElem[:-2] + ">\n")

closeTagsArr.insert(0, addTabs(numTabs) + "</" + strElem[:-2] + ">\n")

else:

strHelper = strElem[:-1].split(': ', maxsplit=1)

if strHelper[0][0] == "-":

strHelper[0] = re.sub(' ', '', strHelper[0].lstrip('-'))

outputArr.append(addTabs(numTabs) +

"<" + strHelper[0] + ">" +

strHelper[1].replace('"', '') +

"</" + strHelper[0] + ">\n")

# if the next element tabs are less than the actual, close the tag

if i + 1 < len(inputArr) - 1:

if inputArr[i][0] > inputArr[i + 1][0]:

outputArr.append(closeTagsArr[0])

del closeTagsArr[0]

# if reach finish add the rest of close tags to the main array

if i == len(inputArr) - 1:

outputArr.extend(closeTagsArr)

outputFile = open("output.xml", "w", encoding="utf-8")

outputFile.write('<?xml version="1.0" encoding="UTF-8"?>\n')

for line in outputArr:

outputFile.write(line)

outputFile.close()

*input.yaml*

Расписание:

Среда:

Пара1:

Время: "10:00-11:30"

Неделя: "все"

Ауд: "1122 АУД."

Корпус: "ул.Ломоносова, д.9, лит. М"

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

сложность:

- экзамен: "нет"

- уровень: "10/10"

Преподаватель: "Бойцев Антон Александрович"

Пара2:

Время: "11:40-13:10"

Неделя: "все"

Ауд: "3322 АУД."

Корпус: "ул.Ломоносова, д.9, лит. М"

Предмет: "ИНОСТРАННЫЙ ЯЗЫК (ПРАК)"

сложность:

- экзамен: "да"

- уровень: "7/10"

Преподаватель: "Наталья Кондрашова"

Пара3:

Время: "13:30-15:00"

Неделя: "все"

Ауд: "1513 АУД."

Корпус: "ул.Ломоносова, д.9, лит. М"

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

сложность:

- экзамен: "да"

- уровень: "9/10"

Преподаватель: "Уздин Валерий Моисеевич"

**Результат работы программы:**

<?xml version="1.0" encoding="UTF-8"?>

<расписание>

<среда>

<пара1>

<время>10:00-11:30</время>

<неделя>все</неделя>

<ауд>1122 ауд.</ауд>

<корпус>ул.ломоносова, д.9, лит. м</корпус>

<предмет>математика (базовый уровень) (лек)</предмет>

<сложность>

<экзамен>нет</экзамен>

<уровень>10/10</уровень>

</сложность>

<преподаватель>бойцев антон александрович</преподаватель>

</пара1>

<пара2>

<время>11:40-13:10</время>

<неделя>все</неделя>

<ауд>3322 ауд.</ауд>

<корпус>ул.ломоносова, д.9, лит. м</корпус>

<предмет>иностранный язык (прак)</предмет>

<сложность>

<экзамен>да</экзамен>

<уровень>7/10</уровень>

</сложность>

<преподаватель>наталья кондрашова</преподаватель>

</пара2>

<пара3>

<время>13:30-15:00</время>

<неделя>все</неделя>

<ауд>1513 ауд.</ауд>

<корпус>ул.ломоносова, д.9, лит. м</корпус>

<предмет>математика (базовый уровень) (прак)</предмет>

<сложность>

<экзамен>да</экзамен>

<уровень>9/10</уровень>

<преподаватель>уздин валерий моисеевич</преподаватель>

</сложность>

</пара3>

</среда>

</расписание>

**Выводы**:

*Personal:*  
Это было очень весело, не потратил много времени, и я узнал слишком много. Надеюсь, что следующая лаборатория будет такой же интересной, как и эта, python, похоже, имеет действительно большой потенциал.

It was super fun, didn’t spend much time and I learned too much. I hope the next labs would be as interesting as this one, python seems to have much potential.

*Testing (optional exercise):*

Моя программа работает примерно в 10 раз быстрее, чем библиотека, но потому что я проверяю гораздо меньше вещей для разбора. Моя программа является самой простой, и она может закончиться с ошибками или не анализировать весь возможный синтаксис, присутствующий во входных данных.

My program runs around 10 times faster than the library, but because I check much fewer things to parse. My program is the most basic one and it can run out with errors or don't parse all the possible syntax present in the input.