

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной
техники

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине
«ИНФОРМАТИКА»

Вариант № 5

Выполнил:

Студент группы Р3113

Бободжонов К.Д

Преподаватели:

Балакшин Б. В , Рыбаков.С.Д

г.Санкт-Петербург, 2023

Оглавление

Оглавление	2
Задание	3
Обязательное задание	2
Дополнительное задание №1	8
Дополнительное задание №2	10
Дополнительное задание №3	10
Заключение	11
Источники	11

Задание

Порядок выполнения работы

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). а) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов. б) Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать. с) Сравнить полученные результаты и объяснить их сходство/различие.
4. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). а) Переписать исходный код, добавив в него использование регулярных выражений. 2 б) Сравнить полученные результаты и объяснить их сходство/различие.
5. Дополнительное задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). а) 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. б) Проанализировать полученные результаты и объяснить их сходство/различие.

Вариант (5)

Исходный формат	Результирующий формат	День недели
XML	YAML	Среда

Обязательное задание

```
from time import *

tm = time()
tab = 0
not_first = 0
yml = ""
content = ""
is_open = False

def settabs(tab):
    tabs = ""
    for i in range(tab):
        tabs += "\t"
    return tabs

def rep_tab(line):
    i = line.find("\t")
    while i != -1:
        line = line[:i] + "\\t" + line[i + 1:]
        i = line.find("\t")
    return line

def brackets(line):
    idx = line.find("\\t")
    if idx != -1:
        return "\"" + line + "\""
    else:
        return line

with open("input.xml", "r", encoding="utf-8") as file:
    for line in file:
        o_tag = ""
        c_tag = ""

        if not_first:
            idx_of_ot_open = line.find('<')
            idx_of_ot_close = line.find('>')

            idx_of_ct_open = line.rfind('</')
            idx_of_ct_close = line.rfind('>')

            if idx_of_ot_open != -1 and idx_of_ot_open != idx_of_ct_open:
                o_tag = line[idx_of_ot_open + 1 : idx_of_ot_close]
            if idx_of_ct_open != -1:
                c_tag = line[idx_of_ct_open + 2 : idx_of_ct_close]

            if o_tag and c_tag:
                yml += "\n" + settabs(tab) + o_tag + ': '
                content += line[idx_of_ot_close + 1 : idx_of_ct_open]
                yml += brackets(rep_tab(content))
                content = ""
                is_open = False

            elif o_tag:
                yml += "\n" + settabs(tab) + o_tag + ': '
                content += line[idx_of_ot_close + 1 : -1]
                tab += 1
                is_open = True

            elif c_tag:
                if is_open:
```

```

        content += "\\n" + line[ : idx_of_ct_open]
        yml += brackets(rep_tab(content))
        content = ""
        tab -= 1
        is_open = False

    else:
        content += "\\n" + rep_tab(line[:-1])

    not_first += 1

with open("4.yaml", "w", encoding="utf-8") as o:
    o.write(yml[1:])

print("Время работы реализации =", time() - tm, "мс")

```

input

```

1 <?xml version="1.0"?>
2 <timetable>
3   <lesson>
4     <time>13:30-15:00</time>
5     <audience>
6       АУД. 2117
7     </audience>
8     <address>
9       Кронвер/кский пр., д.49, лит.А</address>
10    <subject>ИНФОРМАТИКА</subject>
11    <subject_type>ЛАБ</subject_type>
12    <teacher>Рыбаков \Степан\ Дмитриевич</teacher>
13    <lesson_type>Очно - дистанционный</lesson_type>
14  </lesson>
15  <lesson>
16    <time>15:20-16:50</time>
17    <audience>АУД. 2117</audience>
18    <address>К ронв: еркский пр., д.49, лит.А</address>
19    <subject>ИНФОРМАТИКА</subject>
20    <subject_type>ЛАБ</subject_type>
21    <teacher>Рыбаков Степан Дмитриевич</teacher>
22    <lesson_type>Очно - дистанционный</lesson_type>
23  </lesson>
24 </timetable>

```

Output

```

1 timetable:
2   lesson:
3     time: 13:30-15:00
4     audience: "\n\n\t\t\tАУД. 2117\n\n\n\t\t"
5     address: "\t\t\n\t\t\tКронвер/кский пр., д.49, лит.А"
6     subject: ИНФОРМАТИКА
7     subject_type: ЛАБ
8     teacher: Рыбаков \Степан\ Дмитриевич
9     lesson_type: Очно - дистанционный
10  lesson:
11    time: 15:20-16:50
12    audience: АУД. 2117
13    address: К ронв: еркский пр., д.49, лит.А
14    subject: ИНФОРМАТИКА
15    subject_type: ЛАБ
16    teacher: Рыбаков Степан Дмитриевич
17    lesson_type: Очно - дистанционный

```

Дополнительное задание №1

```
import xmlplain
from time import *

tm = time()

with open("input.xml", "r", encoding="utf-8") as file:
    xml = xmlplain.xml_to_obj(file.read(), strip_space=True, fold_dict=True)

with open("lib.yaml", "w", encoding="utf-8") as yml:
    xmlplain.obj_to_yaml(xml, yml)

print("Время работы с библиотеками =", time() - tm, "мс")
```

input

```
1 <?xml version="1.0"?>
2 <timetable>
3   <lesson>
4     <time>13:30-15:00</time>
5     <audience>
6       АУД. 2117
7     </audience>
8     <address>
9       Кронвер/кский пр., д.49, лит.А</address>
10    <subject>ИНФОРМАТИКА</subject>
11    <subject_type>ЛАБ</subject_type>
12    <teacher>Рыбаков \Степан\ Дмитриевич</teacher>
13    <lesson_type>Очно - дистанционный</lesson_type>
14  </lesson>
15  <lesson>
16    <time>15:20-16:50</time>
17    <audience>АУД. 2117</audience>
18    <address>К ронв: еркский пр., д.49, лит.А</address>
19    <subject>ИНФОРМАТИКА</subject>
20    <subject_type>ЛАБ</subject_type>
21    <teacher>Рыбаков Степан Дмитриевич</teacher>
22    <lesson_type>Очно - дистанционный</lesson_type>
23  </lesson>
24 </timetable>
```

Output

```
1 timetable:
2 - lesson:
3   time: "\n\t\t\t13:30-15:00\n\t\t"
4   audience: "\n\t\t\t АУД. 2117"
5   address: "Кронвер/кский пр., д.49, лит.А\n\t\t"
6   subject: " \t\t\tИНФОРМАТИКА\n\t\t"
7   subject_type: ЛАБ
8   teacher: Рыбаков \Степан\ Дмитриевич
9   lesson_type: Очно - дистанционный
10 - lesson:
11   time: 15:20-16:50
12   audience: АУД. 2117
13   address: 'К ронв: еркский пр., д.49, лит.А'
14   subject: ИНФОРМАТИКА
15   subject_type: ЛАБ
16   teacher: Рыбаков Степан Дмитриевич
17   lesson_type: Очно - дистанционный
18
```

Дополнительное задание №2

```
from re import *
from time import *

tm = time()
tab = 0
not_first = 0
reg = ""
content = ""
is_open = False

def settabs(tab):
    tabs = ""
    for i in range(tab):
        tabs += "\t"
    return tabs

def rep_tab(line):
    return sub(r"\t", r"\\t", line)

def brackets(line):
    count = len(findall(r"\\t", line))
    if count:
        return "\"" + line + "\""
    else:
        return line

with open("input.xml", "r", encoding="utf-8") as file:
    for line in file:

        if not_first:
            open_close = findall(r"(<([\w\d]+)>.*</\2>)", line)
            open_ = findall(r"(<([\w\d]+>.*", line)
            close_ = findall(r"(.*)</([\w\d]+>", line)

            if len(open_close):
                reg += "\n" + settabs(tab) + findall(r"(<([\w\d]+)>.*",
open_close[0][0])[0] + ": "
                content += findall(r"(<([\w\d]+>(.*)</([\w\d]+>", open_close[0][0])[0]
                reg += brackets(rep_tab(content))
                content = ""
                is_open = False

            elif len(open_):
                reg += "\n" + settabs(tab) + findall(r"(<([\w\d]+>", open_[0])[0] + ": "
                content += findall(r"(<([\w\d]+>(.*)</([\w\d]+>", open_[0])[0]
                tab += 1
                is_open = True

            elif len(close_):
                if is_open:
                    content += "\\n" + findall(r"(.*)</([\w\d]+>", close_[0])[0]
                    reg += brackets(rep_tab(content))
                    content = ""
                    tab -= 1
                    is_open = False

                else:
                    content += "\\n" + rep_tab(line[:-1])

        not_first += 1
```



```

with open("reg.yaml", "w", encoding="utf-8") as o:
    o.write(reg[1:])

print("Время работы с регулярными выражениями =", time() - tm, "мс")

```

input

```

1  <?xml version="1.0"?>
2  <timetable>
3      <lesson>
4          <time>13:30-15:00</time>
5          <audience>
6              АУД. 2117
7          </audience>
8          <address>
9              Кронвер/кский пр., д.49, лит.А</address>
10         <subject>ИНФОРМАТИКА</subject>
11         <subject_type>ЛАБ</subject_type>
12         <teacher>Рыбаков \Степан\ Дмитриевич</teacher>
13         <lesson_type>Очно - дистанционный</lesson_type>
14     </lesson>
15     <lesson>
16         <time>15:20-16:50</time>
17         <audience>АУД. 2117</audience>
18         <address>К ронв: еркский пр., д.49, лит.А</address>
19         <subject>ИНФОРМАТИКА</subject>
20         <subject_type>ЛАБ</subject_type>
21         <teacher>Рыбаков Степан Дмитриевич</teacher>
22         <lesson_type>Очно - дистанционный</lesson_type>
23     </lesson>
24 </timetable>

```

Output

```

1  timetable:
2      lesson:
3          time:
4              audience:
5                  address:
6                      subject: \n\t\t\t13:30-15:00Кронвер/кский пр., д.49, лит.А \t\t\tИНФОРМАТИКА\n\t\t\t
7                      subject_type: ЛАБ
8                      teacher: Рыбаков \Степан\ Дмитриевич
9                      lesson_type: Очно - дистанционный
10         lesson:
11             time: 15:20-16:50
12             audience: АУД. 2117
13             address: К ронв: еркский пр., д.49, лит.А
14             subject: ИНФОРМАТИКА
15             subject_type: ЛАБ
16             teacher: Рыбаков Степан Дмитриевич
17             lesson_type: Очно - дистанционный

```

Дополнительное задание №3

```
with open("4.py", "r", encoding="utf-8") as c4:  
    code_c4 = c4.read()  
  
with open("lib.py", "r", encoding="utf-8") as lib:  
    code_lib = lib.read()  
  
with open("reg.py", "r", encoding="utf-8") as reg:  
    code_reg = reg.read()  
  
exec(code_c4)  
exec(code_lib)  
exec(code_reg)
```

Output

```
Время работы реализации = 0.006001710891723633 мс  
Время работы с библиотеками = 0.005957365036010742 мс  
Время работы с регулярными выражениями = 0.002785205841064453 мс
```

Заключение

Список литературы.

1. Лямин А.В., Череповская Е.Н. Объектно-ориентированное программирование. Компьютерный практикум. – СПб: Университет ИТМО, 2017. – 143 с..
2. <https://books.ifmo.ru/file/pdf/2256.pdf>