МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

ЛАБОРАТОРНАЯ РАБОТА № 4

“Информатика‘

Номер ISU:336209

Вариант №-5

Выполнил:

Студент группы P3133

Окечукву Александр Чуквуэмека

Преподаватель:

Малышева Т. А.

Санкт-Петербург,

2021

Table of Contents

[Задание 3](#_Toc93027294)

[решение 4](#_Toc93027295)

[Рекомендации 9](#_Toc93027296)

[Вывод 9](#_Toc93027297)

# Задание

A close-up of a document

Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence

# решение

**КОД**

import re  
from timeit import default\_timer  
import xmltodict  
import yaml  
  
def n\_tabdef(i):  
 n\_tab = 0  
 for t in i:   
 if t == ' ' or t == '-':  
 n\_tab += 1  
 else:  
 break  
 return n\_tab  
  
start = default\_timer()   
with open("text.yaml", mode="r", encoding="utf-8") as infile:  
 infilelist = open("text.yaml", mode="r", encoding="utf-8").read().split('\n')  
 outfile = ''   
 closes = {}   
 count = 0  
 for i in infile.readlines():   
 if count < len(infilelist):  
 count += 1  
 i = i.rstrip('\n')   
 if i:   
 n\_tab = n\_tabdef(i)  
 if n\_tab // 2 in closes and n\_tab != n\_tabdef(infilelist[count]):

outfile += '\t' \* (n\_tab // 2) + '</' + closes[n\_tab // 2] + '>' + '\n'   
 closes[n\_tab] = ''  
 if i[-1] == ':':   
 teg = i[n\_tab:len(i) - 1]   
 if teg[:2] == '- ':   
 teg = teg[2:]  
 outfile += '\t' \* (n\_tab // 2) + '<' + teg + '>' + '\n'   
 closes[n\_tab // 2] = teg   
 else:   
 teg = i[n\_tab:i.index(':')]  
 if teg[:2] == '- ':  
 teg = teg[2:]  
 outfile += '\t' \* (n\_tab // 2) + '<' + teg + '>' + i[i.index(':') + 2:] + '</' + teg + '>' + '\n'  
 n\_tab = 0  
 for i in list(closes)[::-1]:  
 if closes[i]:  
 outfile += '\t' \* i + '</' + closes[i] + '>' + '\n'   
 end1 = default\_timer() - start   
 with open("text.xml", mode='w', encoding='utf-8') as fileout:  
 fileout.write(outfile)   
print(end1)

Text

Description automatically generated 

Text

Description automatically generated

**Дополнительные задачи**

import re  
from timeit import default\_timer  
import xmltodict  
import yaml  
  
start = default\_timer()   
  
def n\_tabdef(i):  
 n\_tab = 0  
 for t in i:   
 if t == ' ' or t == '-':  
 n\_tab += 1  
 else:  
 break  
 return n\_tab  
  
with open("text.yaml", mode="r", encoding="utf-8") as infile:  
 yam = yaml.safe\_load(infile)  
 with open("Tuelib.xml", mode='w', encoding='utf-8') as fileout:  
 fileout.write(xmltodict.unparse(yam, pretty=True))   
 end2 = default\_timer()  
  
with open("text.yaml", mode="r", encoding="utf-8") as file:  
 infilelist = open("text.yaml", mode="r", encoding="utf-8").read().split('\n')  
 n\_tab = 0   
 closes = {}  
 outfile = ''  
 count = 0  
 for i in file.readlines():  
 if count < len(infilelist):  
 count += 1  
 if re.fullmatch(r'.+\s', i, re.DOTALL):   
 n\_tab = n\_tabdef(i)  
 if n\_tab // 2 in closes:  
 outfile += '\t' \* (n\_tab // 2) + '</' + closes[n\_tab // 2] + '>' + '\n'  
 closes[n\_tab] = ''  
 if re.fullmatch(r'[^:]+:\s', i):   
 teg = i[n\_tab:len(i) - 2]  
 if re.fullmatch(r'- .+', teg, re.DOTALL):   
 teg = teg[2:]  
 outfile += '\t' \* (n\_tab // 2) + '<' + teg + '>' + '\n'  
 closes[n\_tab // 2] = teg  
 else:  
 teg = i[n\_tab:i.index(':')]  
 if re.fullmatch(r'- .+', teg, re.DOTALL):  
 teg = teg[2:]  
 outfile += '\t' \* (n\_tab // 2) + '<' + teg + '>' + i[i.index(':') + 2:].rstrip('\n') + '</' + \  
 teg + '>' + '\n'  
 n\_tab = 0  
 for i in list(closes)[::-1]:  
 if closes[i]:  
 outfile += '\t' \* i + '</' + closes[i] + '>' + '\n'  
 end3 = default\_timer() - end2  
 with open("text.xml", mode='w', encoding='utf-8') as fileout:  
 fileout.write(outfile)  
  
print(end2)  
print(end3)  
  
with open("text.yaml", mode="r", encoding="utf-8") as infile:  
 yam = yaml.safe\_load(infile)  
 with open("Tue\_csv.xml", mode='w', encoding='utf-8') as fileout:  
 for item, doc in yam.items():  
 fileout.write(item)  
 fileout.write('\n')  
 fileout.write('"' + str(doc) + '"')

Задание 4.

Сравним работу трех программ по скорости с помощью

from timeit import default\_timer

0.00045450000000002433  
0.1535877  
0.0009060000000000179

### 

# 

# **Рекомендации**

1. <https://docs.python.org/3/library/timeit.html>
2. <https://ru.wikipedia.org/wiki/YAML>
3. <https://stackoverflow.com/questions/50846431/converting-a-yaml-file-to-python-xml-object/50846530>

# **Вывод**

Работая над этой задачей, я изучал различные языки разметки, их свойства, работал со строками в Python и регулярными выражениями (Regx).