Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №1**

**«Исследование протоколов, форматов обмена информацией**

**и языков разметки документов»**

**Вариант: 11**

Выполнил: студент 1 курса

Садовой Григорий Владимирович

Группа: Р3107

Принял:

Белозубов Александр Владимирович

Отчёт принят «\_\_»\_\_\_\_\_2022 г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

г. Санкт-Петербург, 2022

**Оглавление**

[Задание 3](#_Toc124434336)

[Основное задание 4](#_Toc124434337)

[Дополнительное задание №1 6](#_Toc124434338)

[Заключение 7](#_Toc124434339)

# Задание

1. Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного. При этом необходимо, чтобы в выбранном дне было не менее двух занятий (можно использовать своё персональное). В случае, если в данный день недели нет таких занятий, то увеличить номер варианта ещё на восемь.
2. 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.
3. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.
4. Дополнительное задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.
5. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). Переписать исходный код, добавив в него использование регулярных выражений.
6. Дополнительное задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). 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.

# Основное задание

def add(): # добавление информации из временных хранилищ в нормальные  
 global flags, text, flag, tmp  
 if flag != 0:  
 flags.append(flag)  
 text.append(tmp)  
 tmp = ""  
  
  
def open\_file():  
 global file  
 file = open("xml.xml", "r", encoding="UTF-8").readlines()  
  
  
def chtenie():  
 global flags, text, flag, tmp, slash  
 for s in file:  
 if s == file[0]:  
 continue  
 for i in range(len(s)):  
 if (slash == True): #экранирование слеша  
 slash = False  
 continue  
 elif (s[i] == "\\"):  
 slash = True  
 elif (s[i] == "<"):  
 if (s[i+1] == "/"):  
 add()  
 flag = 2  
 ignore = True  
 else:  
 flag, tmp = 1, ""  
 elif (s[i] == ">"):  
 add()  
 if (flag == 1):  
 flag = 3  
 else:  
 flag = 0  
 else:  
 tmp += s[i]  
  
  
def printTabs(tabs):  
 res = ""  
 for j in range(tabs): res += " "  
 return res  
  
  
def writeInFile(output):  
 fout = open("bebra.txt", "w", encoding="utf-8")  
 for s in output:  
 if (s == output[0]): s = s.replace("\n", "") #надо убрать пустоту в начале файла  
 fout.write(s)  
 fout.close()  
  
  
def convert():  
 #for i in range(len(flags)):  
 #print(flags[i], text[i])  
 tabs = 0  
 res = ""  
 output = []  
  
 for i in range(len(flags)):  
 if (flags[i] == 1):  
 if (res != ""):  
 output.append(res)  
 res = ""  
 res += ("\n" + printTabs(tabs) + text[i] + ":")  
 tabs += 1  
 elif (flags[i] == 3):  
 res += (" " + text[i])  
 else:  
  
 tabs -= 1  
 writeInFile(output)  
  
  
  
####################  
  
  
flags =[]  
flag = 0  
  
  
  
text = [] #Массив полученного текста  
tmp = "" #временный массив для сортировки  
ignore = False #экранирование слеша  
  
  
open\_file()  
chtenie()  
convert()

# Дополнительное задание №1

import yaml  
import xmltodict as xmltodict  
  
  
newfile = xmltodict.parse(open("xml.xml", "r", encoding="utf8").read()) #преобразуем в словарь  
yaml.dump(newfile, open("bebra.yaml", "w"), allow\_unicode=True)

# Заключение

В ходе выполнения данной лабораторной работы я узнал о ряде языков графической разметки, изучил их синтаксис, и попробовал написать парсер. Также мне удалось поработать со сторонними библиотеками.