Факультет программной инженерии и компьютерной техники

Информатика

Лабораторная работа №4

Исследование языков разметки для документов.

Вариант 3

Преподаватель: Рудникова Тамара Владимировна

Выполнила: Берелехис Светлана Михайловна

Р3112

Санкт-Петербург

2021

Оглавление

[Основное задание: 2](#_Toc86686955)

[Выполнение работы: 3](#_Toc86686956)

[Дополнительное задание 1. 6](#_Toc86686957)

[Дополнительное задание 2. 7](#_Toc86686958)

[Дополнительное задание 3. 10](#_Toc86686959)

[Дополнительное задание 4. 13](#_Toc86686960)

[Вывод 16](#_Toc86686961)

# Основное задание:

1. Определить номер варианта как остаток деления на 36 порядкового номера в списке группы в ISU. В случае, если в данный день недели нет занятий, то увеличить номер варианта на восемь.

2. Изучить форму Бэкуса-Наура.

3. Изучить особенности протоколов и форматов обмена информацией между системами: JSON, YAML, XML.

4. Понять устройство страницы с расписанием для своей группы: <https://itmo.ru/ru/schedule/0/P3110/schedule.htm>

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

6. 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.

7. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.

8. Дополнительное задание задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). a) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов. b) Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать. c) Сравнить полученные результаты и объяснить их сходство/различие.

9. Дополнительное задание задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). a) Переписать исходный код, добавив в него использование регулярных выражений. b) Сравнить полученные результаты и объяснить их сходство/различие.

10.Дополнительное задание задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). a) Используя свою исходную программу из обязательного задания, программу из дополнительного задания №1 и программу из дополнительного задания №2, сравнить десятикратное время выполнения парсинга + конвертации в цикле. b) Проанализировать полученные результаты и объяснить их сходство/различие.

11.Дополнительное задание задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную. a) Переписать исходную, чтобы она осуществляла парсинг и конвертацию исходного файла в любой другой формат (кроме JSON, YAML, XML, HTML): PROTOBUF, TSV, CSV, WML и т.п. b) Проанализировать полученные результаты, объяснить осоебнности использованного формата.

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

13.Написать отчёт о проделанной работе.

14.Подготовиться к устным вопросам на защите.

# Выполнение работы:

Составим БНФ:

root ::= json – общее множество

json ::= array | object - что может быть в json

value ::= string | number | json - возможные значения

array ::= '[' [!']' item (!']' ',' item) \*] ']' - описание массива

object ::= '{' [!'}' prop (!'}' ',' prop) \*] '}' – описание json-объекта

prop ::= name ':' value

name ::= id | string

Создадим исходный файл разметки на языке JSON:

{

"Понедельник":{

"Расписание":{

"Пара1":{

"Время":"08:20-10:50",

"Неделя":"нечетная неделя",

"Ауд":"369А АУД.",

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

"Предмет":"ДИСКРЕТНАЯ МАТЕМАТИКА(ПРАК)",

"Преподаватель":"Поляков Владимир Иванович"

},

"Пара2":{

"Время":"10:00-11:30",

"Неделя":"нечетная неделя",

"Ауд":"369А АУД.",

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

"Предмет":"ДИСКРЕТНАЯ МАТЕМАТИКА(ПРАК)",

"Преподаватель":"Поляков Владимир Иванович"

},

"Пара3(чет)":{

"Время":"11:40-13:10",

"Неделя":"четная неделя",

"Ауд":"306 АУД.",

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

"Предмет":"ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛАБ)",

"Преподаватель":"Ткешелашвили Нино Мерабиевна"

},

"Пара4(чет)":{

"Время":"13:30-15:00",

"Неделя":"четная неделя",

"Ауд":"306 АУД.",

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

"Предмет":"ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛАБ)",

"Преподаватель":"Ткешелашвили Нино Мерабиевна"

}

}

}

}

Запустим программу:

lass JSONelement:  
 inp = []  
 flag = 0  
 def \_\_int\_\_(self):  
 self.name = ""  
  
 def \_\_init\_\_(self, name):  
 self.intStr = ""  
 self.name = name  
  
 def addJson(self, Json):  
 self.flag = 1  
 self.inp = Json  
  
 def addStr(self, str):  
 self.str = str  
 self.flag = 0  
  
 def addNum(self, i):  
 self.i = i  
 self.flag = 2  
  
 def toYMI(self, count):  
 res = ""  
 if self.flag == 0:  
 res += " " \* count + self.name + ": "  
 res += self.str  
 elif self.flag == 1:  
 res += " " \* count + self.name + ":\n"  
 for j in range(len(self.inp)):  
 if j == 0:  
 res += self.inp[j].toYMI(count + 1)  
 else:  
 res += "\n"  
 res += self.inp[j].toYMI(count + 1)  
 else:  
 res += " " \* count + self.name + ": "  
 res += self.i  
 return res  
  
def pareseJSON(str):  
 mas = []  
 i = 0  
 isInQ = False  
 while i < len(str):  
 if str[i] == "\"":  
 if isInQ:  
 isInQ = False  
 else:  
 isInQ = True  
 if not isInQ and str[i] == ":" and i > 0 and str[i - 1] == "\"":  
 i += 1  
 h = i - 2  
 t = 0  
 name = ""  
 while h >= 0 and t < 2:  
 if str[h] == "\"":  
 t += 1  
 else:  
 name = str[h] + name  
 h -= 1  
 nJ = JSONelement(name)  
 if str[i] == "{":  
 otk = 0  
 j = i + 1  
 otk += 1  
 i += 1  
 while i < len(str) and otk > 0:  
 if str[i] == "\"":  
 if isInQ:  
 isInQ = False  
 else:  
 isInQ = True  
 if not isInQ and str[i] == "{":  
 otk += 1  
 if not isInQ and str[i] == "}":  
 otk -= 1  
 i += 1  
 nJ.addJson(pareseJSON(str[j: i - 1]))  
  
 else:  
 j = i + 1  
 isItInMas = False  
 while i < len(str) and (str[i] != "}" or isInQ) and (str[i] != "," or isInQ or isItInMas):  
 if str[i] == "\"":  
 if isInQ:  
 isInQ = False  
 else:  
 isInQ = True  
 if not isInQ and str[i] == "[":  
 isItInMas = True  
 if not isInQ and str[i] == "]":  
 break  
 i += 1  
 if isItInMas:  
 i += 2  
 k = i - 1  
 while str[k] == " " or str[k] == "\n":  
 k -= 1  
 e = str[j: k]  
 nJ.addStr(e)  
 mas.append(nJ)  
 i += 1  
 return mas  
  
  
inputfile = open('input.json', 'r', encoding="utf-8")  
file = inputfile.read()  
mas = pareseJSON(file)  
for i in mas:  
 print(i.toYMI(0))

Получим:

Понедельник:

Расписание:

Пара1:

Время: 08:20-10:50

Неделя: нечетная неделя

Ауд: 369А АУД.

Корпус: Кронверкский пр., д.49, лит.А

Предмет: ДИСКРЕТНАЯ МАТЕМАТИКА(ПРАК)

Преподаватель: Поляков Владимир Иванович

Пара2:

Время: 10:00-11:30

Неделя: нечетная неделя

Ауд: 369А АУД.

Корпус: Кронверкский пр., д.49, лит.А

Предмет: ДИСКРЕТНАЯ МАТЕМАТИКА(ПРАК)

Преподаватель: Поляков Владимир Иванович

Пара3(чет):

Время: 11:40-13:10

Неделя: четная неделя

Ауд: 306 АУД.

Корпус: Кронверкский пр., д.49, лит.А

Предмет: ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛАБ)

Преподаватель: Ткешелашвили Нино Мерабиевна

Пара4(чет):

Время: 13:30-15:00

Неделя: четная неделя

Ауд: 306 АУД.

Корпус: Кронверкский пр., д.49, лит.А

Предмет: ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛАБ)

Преподаватель: Ткешелашвили Нино Мерабиевна

# Дополнительное задание 1.

Запустим программу:

import yaml, json  
  
inputfile = open('input.json', 'r', encoding="utf-8")  
yaml.dump(json.load(inputfile), open('output.yml', 'wb'), encoding='utf8', allow\_unicode=True)  
inputfile.close()

В выходном файле:

Понедельник:  
 Расписание:  
 Пара1:  
 Ауд: 369А АУД.  
 Время: 08:20-10:50  
 Корпус: Кронверкский пр., д.49, лит.А  
 Неделя: нечетная неделя  
 Предмет: ДИСКРЕТНАЯ МАТЕМАТИКА(ПРАК)  
 Преподаватель: Поляков Владимир Иванович  
 Пара2:  
 Ауд: 369А АУД.  
 Время: 10:00-11:30  
 Корпус: Кронверкский пр., д.49, лит.А  
 Неделя: нечетная неделя  
 Предмет: ДИСКРЕТНАЯ МАТЕМАТИКА(ПРАК)  
 Преподаватель: Поляков Владимир Иванович  
 Пара3(чет):  
 Ауд: 306 АУД.  
 Время: 11:40-13:10  
 Корпус: Кронверкский пр., д.49, лит.А  
 Неделя: четная неделя  
 Предмет: ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛАБ)  
 Преподаватель: Ткешелашвили Нино Мерабиевна  
 Пара4(чет):  
 Ауд: 306 АУД.  
 Время: 13:30-15:00  
 Корпус: Кронверкский пр., д.49, лит.А  
 Неделя: четная неделя  
 Предмет: ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛАБ)  
 Преподаватель: Ткешелашвили Нино Мерабиевна

Отличий нет.

# Дополнительное задание 2.

Запустим программу:

import re  
class JSONelement:  
 inp = []  
 flag = 0  
  
 def \_\_int\_\_(self):  
 self.name = ""  
  
 def \_\_init\_\_(self, name):  
 self.intStr = ""  
 self.name = name  
  
 def addJson(self, Json):  
 self.flag = 1  
 self.inp = Json  
  
 def addStr(self, str):  
 self.str = str  
 self.flag = 0  
  
 def toYMI(self, count):  
 res = ""  
 if self.flag == 0:  
 res += " " \* count + self.name + ": "  
 res += self.str  
 else:  
 res += " " \* count + self.name + ":\n"  
 for j in range(len(self.inp)):  
 if j == 0:  
 res += self.inp[j].toYMI(count + 1)  
 else:  
 res += "\n"  
 res += self.inp[j].toYMI(count + 1)  
 return res  
  
def pareseJSON(str):  
 mas = []  
 if "{" in str or "}" is str:  
 i = 0  
 while i < len(str):  
 if str[i] == "{" and i > 0 and str[i - 1] == ":":  
 otk = 0  
 j = i + 1  
 otk += 1  
 i += 1  
 h = i - 3  
 t = 0  
 name = ""  
 while h >= 0 and t < 2:  
 if str[h] == "\"":  
 t += 1  
 else:  
 name = str[h] + name  
 h -= 1  
 while i < len(str) and otk > 0:  
 if str[i] == "{":  
 otk += 1  
 if str[i] == "}":  
 otk -= 1  
 i += 1  
 nJ = JSONelement(name)  
 nJ.addJson(pareseJSON(str[j: i - 1]))  
 mas.append(nJ)  
 i += 1  
 return mas  
 else:  
 mass = parseSimpleJSON(str)  
 for m in mass:  
 mas.append(m)  
 return mas  
  
  
def parseSimpleJSON(s):  
 res = re.findall(r'"(\S+)":"([,\sА-Яа-я\d\.\-\:\)\(]+)"', s)  
 mas = []  
 for i in res:  
 name = i[0]  
 value = i[1]  
 je = JSONelement(name)  
 je.addStr(value)  
 mas.append(je)  
 return mas  
  
  
inputfile = open('input.json', 'r', encoding="utf-8")  
file = inputfile.read()  
mas = pareseJSON(file)  
for i in mas:  
 print(i.toYMI(0))

Вывод:

Понедельник:

Расписание:

Пара1:

Время: 08:20-10:50

Неделя: нечетная неделя

Ауд: 369А АУД.

Корпус: Кронверкский пр., д.49, лит.А

Предмет: ДИСКРЕТНАЯ МАТЕМАТИКА(ПРАК)

Преподаватель: Поляков Владимир Иванович

Пара2:

Время: 10:00-11:30

Неделя: нечетная неделя

Ауд: 369А АУД.

Корпус: Кронверкский пр., д.49, лит.А

Предмет: ДИСКРЕТНАЯ МАТЕМАТИКА(ПРАК)

Преподаватель: Поляков Владимир Иванович

Пара3(чет):

Время: 11:40-13:10

Неделя: четная неделя

Ауд: 306 АУД.

Корпус: Кронверкский пр., д.49, лит.А

Предмет: ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛАБ)

Преподаватель: Ткешелашвили Нино Мерабиевна

Пара4(чет):

Время: 13:30-15:00

Неделя: четная неделя

Ауд: 306 АУД.

Корпус: Кронверкский пр., д.49, лит.А

Предмет: ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛАБ)

Преподаватель: Ткешелашвили Нино Мерабиевна

# Дополнительное задание 3.

Запустим программу:

import time  
import yaml, json  
import re  
  
class JSONelement:  
 inp = []  
 flag = 0  
 def \_\_int\_\_(self):  
 self.name = ""  
  
 def \_\_init\_\_(self, name):  
 self.intStr = ""  
 self.name = name  
  
 def addJson(self, Json):  
 self.flag = 1  
 self.inp = Json  
  
 def addStr(self, str):  
 self.str = str  
 self.flag = 0  
  
 def addNum(self, i):  
 self.i = i  
 self.flag = 2  
  
 def toYMI(self, count):  
 res = ""  
 if self.flag == 0:  
 res += " " \* count + self.name + ": "  
 res += self.str  
 elif self.flag == 1:  
 res += " " \* count + self.name + ":\n"  
 for j in range(len(self.inp)):  
 if j == 0:  
 res += self.inp[j].toYMI(count + 1)  
 else:  
 res += "\n"  
 res += self.inp[j].toYMI(count + 1)  
 else:  
 res += " " \* count + self.name + ": "  
 res += self.i  
 return res  
  
def pareseJSON(str):  
 mas = []  
 i = 0  
 isInQ = False  
 while i < len(str):  
 if str[i] == "\"":  
 if isInQ:  
 isInQ = False  
 else:  
 isInQ = True  
 if not isInQ and str[i] == ":" and i > 0 and str[i - 1] == "\"":  
 i += 1  
 h = i - 2  
 t = 0  
 name = ""  
 while h >= 0 and t < 2:  
 if str[h] == "\"":  
 t += 1  
 else:  
 name = str[h] + name  
 h -= 1  
 nJ = JSONelement(name)  
 if str[i] == "{":  
 otk = 0  
 j = i + 1  
 otk += 1  
 i += 1  
 while i < len(str) and otk > 0:  
 if str[i] == "\"":  
 if isInQ:  
 isInQ = False  
 else:  
 isInQ = True  
 if not isInQ and str[i] == "{":  
 otk += 1  
 if not isInQ and str[i] == "}":  
 otk -= 1  
 i += 1  
 nJ.addJson(pareseJSON(str[j: i - 1]))  
  
 else:  
 j = i + 1  
 isItInMas = False  
 while i < len(str) and (str[i] != "}" or isInQ) and (str[i] != "," or isInQ or isItInMas):  
 if str[i] == "\"":  
 if isInQ:  
 isInQ = False  
 else:  
 isInQ = True  
 if not isInQ and str[i] == "[":  
 isItInMas = True  
 if not isInQ and str[i] == "]":  
 break  
 i += 1  
 if isItInMas:  
 i += 2  
 k = i - 1  
 while str[k] == " " or str[k] == "\n":  
 k -= 1  
 e = str[j: k]  
 nJ.addStr(e)  
 mas.append(nJ)  
 i += 1  
 return mas  
  
def pareseJSONThird(str):  
 mas = []  
 if "{" in str or "}" is str:  
 i = 0  
 while i < len(str):  
 if str[i] == "{" and i > 0 and str[i - 1] == ":":  
 otk = 0  
 j = i + 1  
 otk += 1  
 i += 1  
 h = i - 3  
 t = 0  
 name = ""  
 while h >= 0 and t < 2:  
 if str[h] == "\"":  
 t += 1  
 else:  
 name = str[h] + name  
 h -= 1  
 while i < len(str) and otk > 0:  
 if str[i] == "{":  
 otk += 1  
 if str[i] == "}":  
 otk -= 1  
 i += 1  
 nJ = JSONelement(name)  
 nJ.addJson(pareseJSONThird(str[j: i - 1]))  
 mas.append(nJ)  
 i += 1  
 return mas  
 else:  
 mass = parseSimpleJSONThird(str)  
 for m in mass:  
 mas.append(m)  
 return mas  
  
  
def parseSimpleJSONThird(s):  
 res = re.findall(r'"(\S+)":"([,\sА-Яа-я\d\.\-\:\)\(]+)"', s)  
 mas = []  
 for i in res:  
 name = i[0]  
 value = i[1]  
 je = JSONelement(name)  
 je.addStr(value)  
 mas.append(je)  
 return mas  
  
startTime = time.time()  
for i in range(10):  
 inputfile = open('input.json', 'r', encoding="utf-8")  
 file = inputfile.read()  
 mas = pareseJSON(file)  
 for j in mas:  
 print(j.toYMI(0))  
 inputfile.close()  
secondTime = time.time()  
for i in range(10):  
 inputfile = open('input.json', 'r', encoding="utf-8")  
 output = open('output.yml', 'wb')  
 yaml.dump(json.load(inputfile), output, encoding='utf8', allow\_unicode=True)  
 inputfile.close()  
 output.close()  
thirdTime = time.time()  
for i in range(10):  
 inputfile = open('input.json', 'r', encoding="utf-8")  
 file = inputfile.read()  
 mas = pareseJSONThird(file)  
 for j in mas:  
 print(j.toYMI(0))  
 inputfile.close()  
fourthTime = time.time()  
print("--- %s первый способ ---" % (secondTime - startTime))  
print("--- %s второй способ ---" % (thirdTime - secondTime))  
print("--- %s третий способ ---" % (fourthTime - thirdTime))

Значимый вывод программы:

--- 0.04709339141845703 первый способ ---

--- 0.030994892120361328 второй способ ---

--- 0.04686880111694336 третий способ ---

Вывод: с библиотеками получается быстрее.

# Дополнительное задание 4.

Запустим программу:

class JSONelement:  
 inp = []  
 flag = 0  
 def \_\_int\_\_(self):  
 self.name = ""  
  
 def \_\_init\_\_(self, name):  
 self.intStr = ""  
 self.name = name  
  
 def addJson(self, Json):  
 self.flag = 1  
 self.inp = Json  
  
 def addStr(self, str):  
 self.str = str  
 self.flag = 0  
  
 def addNum(self, i):  
 self.i = i  
 self.flag = 2  
  
 def toWLM(self, count):  
 res = ""  
 if self.flag == 0:  
 res += " " \* count + "<p id = \"" + self.name + "\">"  
 res += self.str + "</p>"  
 elif self.flag == 1:  
 res += " " \* count + "<card id=\"" + self.name + "\">"  
 res += "\n"  
 for Json in self.inp:  
 res += Json.toWLM(count + 1)  
 res += "</card>\n"  
 else:  
 res += self.i  
 if self.name != "":  
 res += "\n"  
 return res  
  
 def toYMI(self, count):  
 res = ""  
 if self.flag == 0:  
 res += " " \* count + self.name + ": "  
 res += self.str  
 elif self.flag == 1:  
 res += " " \* count + self.name + ":\n"  
 for j in range(len(self.inp)):  
 if j == 0:  
 res += self.inp[j].toYMI(count + 1)  
 else:  
 res += "\n"  
 res += self.inp[j].toYMI(count + 1)  
 else:  
 res += " " \* count + self.name + ": "  
 res += self.i  
 return res  
  
def pareseJSON(str):  
 mas = []  
 i = 0  
 isInQ = False  
 while i < len(str):  
 if str[i] == "\"":  
 if isInQ:  
 isInQ = False  
 else:  
 isInQ = True  
 if not isInQ and str[i] == ":" and i > 0 and str[i - 1] == "\"":  
 i += 1  
 h = i - 2  
 t = 0  
 name = ""  
 while h >= 0 and t < 2:  
 if str[h] == "\"":  
 t += 1  
 else:  
 name = str[h] + name  
 h -= 1  
 nJ = JSONelement(name)  
 if str[i] == "{":  
 otk = 0  
 j = i + 1  
 otk += 1  
 i += 1  
 while i < len(str) and otk > 0:  
 if str[i] == "\"":  
 if isInQ:  
 isInQ = False  
 else:  
 isInQ = True  
 if not isInQ and str[i] == "{":  
 otk += 1  
 if not isInQ and str[i] == "}":  
 otk -= 1  
 i += 1  
 nJ.addJson(pareseJSON(str[j: i - 1]))  
  
 else:  
 j = i + 1  
 isItInMas = False  
 while i < len(str) and (str[i] != "}" or isInQ) and (str[i] != "," or isInQ or isItInMas):  
 if str[i] == "\"":  
 if isInQ:  
 isInQ = False  
 else:  
 isInQ = True  
 if not isInQ and str[i] == "[":  
 isItInMas = True  
 if not isInQ and str[i] == "]":  
 break  
 i += 1  
 if isItInMas:  
 i += 2  
 k = i - 1  
 while str[k] == " " or str[k] == "\n":  
 k -= 1  
 e = str[j: k]  
 nJ.addStr(e)  
 mas.append(nJ)  
 i += 1  
 return mas  
  
  
inputfile = open('input.json', 'r', encoding="utf-8")  
file = inputfile.read()  
mas = pareseJSON(file)  
for i in mas:  
 print(i.toWLM(0))

Вывод программы:

<card id="Понедельник">

<card id="Расписание">

<card id="Пара1">

<p id = "Время">08:20-10:50</p>

<p id = "Неделя">нечетная неделя</p>

<p id = "Ауд">369А АУД.</p>

<p id = "Корпус">Кронверкский пр., д.49, лит.А</p>

<p id = "Предмет">ДИСКРЕТНАЯ МАТЕМАТИКА(ПРАК)</p>

<p id = "Преподаватель">Поляков Владимир Иванович</p>

</card>

<card id="Пара2">

<p id = "Время">10:00-11:30</p>

<p id = "Неделя">нечетная неделя</p>

<p id = "Ауд">369А АУД.</p>

<p id = "Корпус">Кронверкский пр., д.49, лит.А</p>

<p id = "Предмет">ДИСКРЕТНАЯ МАТЕМАТИКА(ПРАК)</p>

<p id = "Преподаватель">Поляков Владимир Иванович</p>

</card>

<card id="Пара3(чет)">

<p id = "Время">11:40-13:10</p>

<p id = "Неделя">четная неделя</p>

<p id = "Ауд">306 АУД.</p>

<p id = "Корпус">Кронверкский пр., д.49, лит.А</p>

<p id = "Предмет">ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛАБ)</p>

<p id = "Преподаватель">Ткешелашвили Нино Мерабиевна</p>

</card>

<card id="Пара4(чет)">

<p id = "Время">13:30-15:00</p>

<p id = "Неделя">четная неделя</p>

<p id = "Ауд">306 АУД.</p>

<p id = "Корпус">Кронверкский пр., д.49, лит.А</p>

<p id = "Предмет">ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ(ЛАБ)</p>

<p id = "Преподаватель">Ткешелашвили Нино Мерабиевна</p>

</card>

</card>

</card>

# Вывод

При выполнение лабораторной работы я познакомилась с форматами json, yaml и wml и познакомилась с библиотеками python для конвертации файлов из json в yaml