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»

по теме

“Исследование протоколов, форматов обмена информацией и языков разметки документов”

Вариант № 24

**Выполнил:**

студент группы Р3117

Агей Михаил Александрович  
  
**Преподаватель:**

Машина Екатерина Алексеевна

Санкт-Петербург

2022

## Оглавление

[**Оглавление**](#_l11qsy41eg5b) **2**

[**Задания**](#_16q2pwcoe4hi) **3**

[**Основные этапы выполнения**](#_myi3jk4115qm) **4**

[Основное задание](#_1vzhjqgz1j9) 4

[Дополнительное задание 1](#_277610enxk65) 5

[Дополнительное задание 2](#_clcru3b1mse6) 6

[Дополнительное задание 3](#_yl7kc2xkqow5) 8

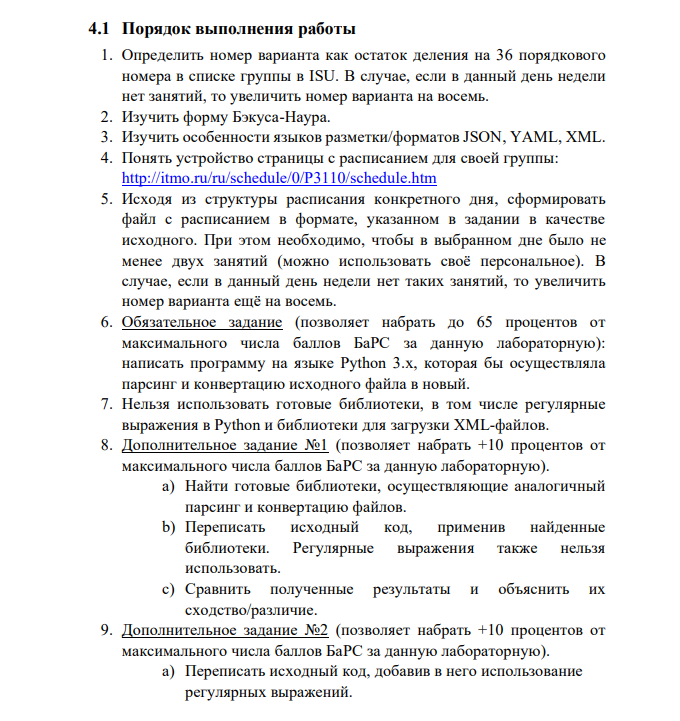
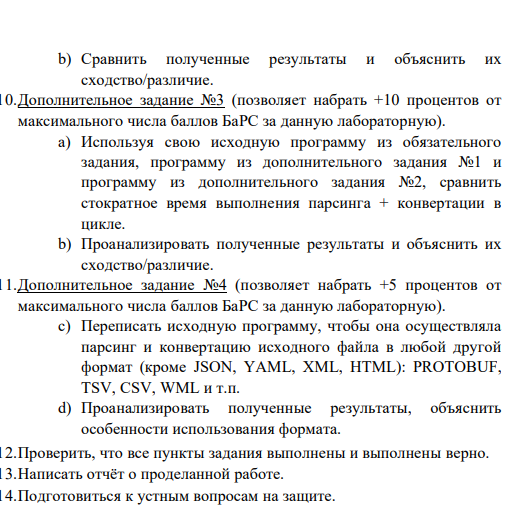
[Дополнительное задание 4](#_hijd7lx5lvef) 8

[**Вывод**](#_gwpytctht36r) **10**

[**Список использованных источников**](#_vnqqdyplpuyq) **10**

[**Приложение**](#_1azy31yytj04) **11**

## Задания



В моем варианте необходимо было произвести конвертацию из JSON в XML (см [Приложение](#_1azy31yytj04)). За основу наполнения JSON-файла взято мое расписание по вторникам нечетной недели.

## Основные этапы выполнения

*JSON-файл (schedule.json)*

*{*

*"first": {*

*"day": "Вт",*

*"time": "08:20-09:50",*

*"room": "ул.Ломоносова, д.9, лит. М",*

*"lesson": "Информатика",*

*"lesson-format": "Очно-дистанционный"*

*},*

*"second": {*

*"day": "Вт",*

*"time": "10:00-11:30",*

*"room": "ул.Ломоносова, д.9, лит. М",*

*"lesson": "ОПД",*

*"lesson-format": "Очно-дистанционный"*

*},*

*"third": {*

*"day": "Вт",*

*"time": "11:40-13:10",*

*"room": "ул.Ломоносова, д.9, лит. М",*

*"lesson": "Программирование",*

*"lesson-format": "Дистанционный"*

*}*

*}*

### Основное задание

*Код (lab4\_0.py)*

*# coding: utf-8*

import json

def zero():

with open("schedule.json", "r", encoding="utf-8") as ex:

s = json.load(ex)

with open("lab4\_0.xml", "w", encoding="utf-8") as im:

im.write('<?xml version="1.0" ?>')

c = ['<root>']

for number in s:

a = "<" + number + ">"

c.append(a)

for param in s[number]:

a = "<" + param + ">" + s[number][param] + "</" + param + ">"

c.append(a)

a = "</" + number + ">"

c.append(a)

c.append('</root>')

for string in c:

im.write("\n")

im.write(string)

*Результат работы программы (lab4\_0.xml)*

*<?xml version="1.0" ?>*

*<root>*

*<first>*

*<day>Вт</day>*

*<time>08:20-09:50</time>*

*<room>ул.Ломоносова, д.9, лит. М</room>*

*<lesson>Информатика</lesson>*

*<lesson-format>Очно-дистанционный</lesson-format>*

*</first>*

*<second>*

*<day>Вт</day>*

*<time>10:00-11:30</time>*

*<room>ул.Ломоносова, д.9, лит. М</room>*

*<lesson>ОПД</lesson>*

*<lesson-format>Очно-дистанционный</lesson-format>*

*</second>*

*<third>*

*<day>Вт</day>*

*<time>11:40-13:10</time>*

*<room>ул.Ломоносова, д.9, лит. М</room>*

*<lesson>Программирование</lesson>*

*<lesson-format>Дистанционный</lesson-format>*

*</third>*

*</root>*

### Дополнительное задание 1

*Код*

*# coding: utf-8*

*import json as JS*

*import xml.etree.ElementTree as ET*

*def first():*

*with open("schedule.json", "r", encoding="utf-8") as json\_file:*

*data = JS.load(json\_file)*

*root = ET.Element("root")*

*for number in data:*

*a = ET.SubElement(root, number)*

*for param in data[number]:*

*ET.SubElement(a, param).text = data[number][param]*

*tree = ET.ElementTree(root)*

*ET.indent(tree, ' ')*

*tree.write("lab4\_1.xml", encoding="utf-8", xml\_declaration=True)*

*Результат работы программы*

*<?xml version='1.0' encoding='utf-8'?>*

*<root>*

*<first>*

*<day>Вт</day>*

*<time>08:20-09:50</time>*

*<room>ул.Ломоносова, д.9, лит. М</room>*

*<lesson>Информатика</lesson>*

*<lesson-format>Очно-дистанционный</lesson-format>*

*</first>*

*<second>*

*<day>Вт</day>*

*<time>10:00-11:30</time>*

*<room>ул.Ломоносова, д.9, лит. М</room>*

*<lesson>ОПД</lesson>*

*<lesson-format>Очно-дистанционный</lesson-format>*

*</second>*

*<third>*

*<day>Вт</day>*

*<time>11:40-13:10</time>*

*<room>ул.Ломоносова, д.9, лит. М</room>*

*<lesson>Программирование</lesson>*

*<lesson-format>Дистанционный</lesson-format>*

*</third>*

*</root>*

### Дополнительное задание 2

*Код*

*# coding: utf-8*

*import json*

*def second():*

*with open("schedule.json", "r", encoding="utf-8") as ex:*

*s = json.load(ex)*

*with open("lab4\_2.xml", "w", encoding="utf-8") as im:*

*im.write('<?xml version="1.0" ?>')*

*c = ['<root>']*

*for number in s:*

*c.append(f'<{number}>')*

*for param in s[number]:*

*c.append(f'<{param}>{s[number][param]}</{param}>')*

*c.append(f'</{number}>')*

*c.append('</root>')*

*for string in c:*

*im.write("\n")*

*im.write(string)*

*Результат работы программы*

*<?xml version="1.0" ?>*

*<root>*

*<first>*

*<day>Вт</day>*

*<time>08:20-09:50</time>*

*<room>ул.Ломоносова, д.9, лит. М</room>*

*<lesson>Информатика</lesson>*

*<lesson-format>Очно-дистанционный</lesson-format>*

*</first>*

*<second>*

*<day>Вт</day>*

*<time>10:00-11:30</time>*

*<room>ул.Ломоносова, д.9, лит. М</room>*

*<lesson>ОПД</lesson>*

*<lesson-format>Очно-дистанционный</lesson-format>*

*</second>*

*<third>*

*<day>Вт</day>*

*<time>11:40-13:10</time>*

*<room>ул.Ломоносова, д.9, лит. М</room>*

*<lesson>Программирование</lesson>*

*<lesson-format>Дистанционный</lesson-format>*

*</third>*

*</root>*

### Дополнительное задание 3

*Код*

*import time*

*from lab4\_0 import zero*

*from lab4\_1 import first*

*from lab4\_2 import second*

*start0 = time.time()*

*for \_ in range(100):*

*zero()*

*end0 = time.time()*

*start1 = time.time()*

*for \_ in range(100):*

*first()*

*end1 = time.time()*

*start2 = time.time()*

*for \_ in range(100):*

*second()*

*end2 = time.time()*

*print("Нулевая программа выполняется 100 раз за", (end0 - start0) \* 1000, "миллисекунд")*

*print("Первая программа выполняется 100 раз за", (end1 - start1) \* 1000, "миллисекунд")*

*print("Вторая программа выполняется 100 раз за", (end2 - start2) \* 1000, "миллисекунд")*

*Результат работы программы*

Нулевая программа выполняется 100 раз за 58.03680419921875 миллисекунд

Первая программа выполняется 100 раз за 63.00067901611328 миллисекунд

Вторая программа выполняется 100 раз за 57.886600494384766 миллисекунд

### Дополнительное задание 4

*Код*

*# coding: utf-8*

*import json*

*import csv*

*with open('schedule.json', "r", encoding="utf-8") as json\_file:*

*data = json.load(json\_file)*

*print(data)*

*data\_file = open('lab4\_4.csv', 'w', encoding="utf-8")*

*csv\_writer = csv.writer(data\_file)*

*count = 0*

*for number in data.keys():*

*print(number)*

*if count == 0:*

*header = data[number].keys()*

*csv\_writer.writerow(header)*

*count += 1*

*csv\_writer.writerow(data[number].values())*

*data\_file.close()*

*Результат работы программы*

*day, time, room, lesson, lesson-format*

*Вт,08:20-09:50, "ул.Ломоносова, д.9, лит.М", Информатика, Очно-дистанционный*

*Вт,10:00-11:30, "ул.Ломоносова, д.9, лит. М", ОПД, Очно-дистанционный*

*Вт,11:40-13:10,"ул.Ломоносова, д.9, лит. М", Программирование, Дистанционный*

**Анализ**

1. Результаты работы программ из основного и двух первых дополнительных заданий не отличаются. Единственная деталь: при использовании специальных библиотек в итоговом файле проставляется табуляция (2 пробела). Получается красивее и удобнее, но на логику файла и работу с ним это не влияет.
2. Заметим, что программы, написанные без спец возможностей и только с использованием регулярных выражений, работают примерно одинаково. Программа, в которой используются возможности библиотек, работает заметно (если сравнивать целые количества миллисекунд) дольше, потому что работа регулярных выражений почти не замедляет программу, написанную с использованием лишь стандартных возможностей, в то время как подключение библиотек и конвертация с их помощью явно замедляет программу. На отрезке даже 100 запусков разница может показаться незначительной, но если кратно увеличивать количество запусков, отличие будет заметным.
3. В дополнительном задании 4 была произведена конвертация из JSON в CSV. Последний формат является строковым отображением данных в базах данных, где значения одной строки разделяются запятыми. Поэтому первая строка в итоговом файле - названия столбцов(ключей), а в остальных строках последовательно располагаются значения этих ключей для каждого занятия в расписании. Конвертация в 1 файл получилась правильной, благодаря тому что каждый “объект” в JSON файле имеет одинаковый с остальными набор ключей.

## Вывод

В процессе выполнения лабораторной работы 4 по информатике я познакомился со многими форматами разметки. На примере формата JSON научился производить парсинг и конвертацию из 1 формата в другой, используя стандартные возможности языка Python, регулярные выражения и готовые библиотеки. А также проанализировал время работы программ, которые имеют одинаковый результат, но написаны при помощи разных возможностей разработки.

## Список использованных источников

<https://www.geeksforgeeks.org/>

<https://stackoverflow.com/>

[xml.etree.ElementTree — The ElementTree XML API — Python 3.10.8 documentation](https://docs.python.org/3/library/xml.etree.elementtree.html#xml.etree.ElementTree.indent)

## Приложение

