

Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

**Исследование протоколов, форматов обмена информацией и языков
разметки документов**

Вариант 10

Выполнил:

Кулагин Вячеслав Дмитриевич, Р3109

Проверила:

Рудникова Тамара Владимировна

Санкт-Петербург

2023

Оглавление

Задание.....	3
Исходные данные и файл.....	5
Обязательное задание.....	7
Получаем XML:.....	8
Дополнительное задание №1.....	10
Итоговый XML.....	10
Дополнительное задание №2.....	12
Полученный XML.....	13
Дополнительное задание №3.....	15
Исходный файл.....	15
Исходный код.....	16
Полученный XML.....	18
Дополнительное задание №4.....	20
Основное задание.....	20
Дополнительное задание №1.....	21
Дополнительное задание №2.....	21
Дополнительное задание №3.....	22
Дополнительное задание №5.....	23
Дополнительное задание №5.....	24
Полученный CSV.....	26
Вывод.....	28
Список используемых источников.....	28

Задание

1. Определить номер варианта как остаток деления на 36 последних двух цифр своего идентификационного номера в ISU. В случае, если в данный день недели нет занятий, то увеличить номер варианта на восемь.
2. Изучить форму Бэкуса-Наура.
3. Изучить основные принципы организации формальных грамматик.
4. Изучить особенности языков разметки/форматов JSON, YAML, XML.
5. Понять устройство страницы с расписанием на примере расписания лектора
6. 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.
7. Обязательное задание (позволяет набрать до 45 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 путём простой замены метасимволов исходного формата на метасимволы результирующего формата.
8. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.
9. 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.
 - в) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
10. 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.
 - б) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
11. Дополнительное задание №3 (позволяет набрать +25 процентов от максимального числа баллов БаРС за данную лабораторную).

а) Переписать исходный код таким образом, чтобы для решения задачи использовались формальные грамматики. То есть ваш код должен уметь осуществлять парсинг и конвертацию любых данных, представленных в исходном формате, в данные, представленные в результирующем формате: как с готовыми библиотеками из дополнительного задания №1.

б) Проверку осуществить как минимум для расписания с двумя учебными днями по два занятия в каждом.

в) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

12. Дополнительное задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).

а) Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле.

б) Проанализировать полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

13. Дополнительное задание №5 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).

а) Переписать исходную программу, чтобы она осуществляла парсинг и конвертацию исходного файла в любой другой формат (кроме JSON, YAML, XML, HTML): PROTOBUF, TSV, CSV, WML и т.п.

б) Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте.

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

15. Написать отчёт о проделанной работе.

16. Подготовиться к устным вопросам на защите.

Исходные данные и файл

Для варианта 10 было предложено сделать парсинг и конвертацию из YAML в XML для дня недели — вторник. Был сделан исходный файл в YAML:

```
schedule:
  day_number: 2
  week_number: 11
  date: "2023-11-07"
  lessons:
    - id: lesson-206
      name: Математический анализ
      lesson_type: Лекции
      group: МАТ АН ПИиКТ 13
      time:
        start: 08:20
        end: 09:50
      teacher_name: Правдин Константин Владимирович
      place:
        campus: Кронверкский пр., д.49, лит.А
        room: Ауд.Orange Classroom (1229)
        distant: Очно – дистанционный
    - id: lesson-210
      name: Математический анализ
      lesson_type: Практические занятия
      group: МАТ АН ПИиКТ 13.3
      time:
        start: 10:00
        end: 11:30
      teacher_name: Блейхер Оксана Владимировна
      place:
        campus: Кронверкский пр., д.49, лит.А
        room: Ауд.2430
        distant: Очно – дистанционный
    - id: lesson-217
      name: Основы дискретной математики (базовый уровень)
      lesson_type: Лекции
      group: ДИСКР МАТ 2
      time:
        start: 11:40
        end: 13:10
      teacher_name: Поляков Владимир Иванович
      place:
        campus: Кронверкский пр., д.49, лит.А
        room: Ауд.2337
        distant: Очно – дистанционный
    - id: lesson-220
      name: Основы дискретной математики (базовый уровень)
      lesson_type: Практические занятия
```

group: ДИСКР МАТ 2.1
time:
 start: 13:30
 end: 15:00
teacher_name: Поляков Владимир Иванович
place:
 campus: Кронверкский пр., д.49, лит.А
 room: Ауд.2337
 distant: Очно – дистанционный

Обязательное задание

Задание заключается в простой замене метасимволов YAML в метасимволы XML. Для этого необходимо подсчитывать количество знаков табуляции, потому что по сути весь YAML именно на этом строится.

Листинг представлен на языке Python, данные берутся из файла data.yml и записываются в файл res_osn.xml:

```
f = open('data.yml')
st_out = ''
lines = [i.replace('\n', '') for i in f]
open_curves = []
tabs = 0
first_slash = False

for n in lines:
    if n == '---':
        st_out += '<?xml version="1.0" encoding="UTF-8"?>\n'
        st_out += '<schedule>\n'
    if ':' in n:
        tabs_prev = tabs
        tabs = n.count(' ')
        if (tabs_prev - tabs) > 0:
            st_out += '\t' * (len(open_curves))
            st_out += f'</{open_curves[-1]}>\n'
            open_curves.pop(-1)
        open_curves.append(n[:n.index(':')].replace(' ', ''))
        if (n.index(':') != 0) and (n[n.index(':') + 1:] != '') and (n[0] != '-'):
            st_out += '\t' * len(open_curves)
            st_out += f'<{open_curves[-1]}>'
            st_out += n[n.index(':') + 2:] + f'</{open_curves[-1]}>\n'
            open_curves.pop(-1)
        else:
            if n[0] == '-':
                if first_slash:
                    st_out += '\t' * (len(open_curves) - 1)
                    st_out += f'</{open_curves[0]}>\n'
                    st_out += '\t' * (len(open_curves) - 1)
                    st_out += f'<{open_curves[0]}>\n'
```

```

        st_out += '\t' * len(open_curves)
        st_out += f'<{open_curves[-1][1:]}>'
        st_out += n[n.index(':') + 2:] + f'</{open_curves[-1][1:]>\n"

        open_curves.pop(-1)
        first_slash = True
    else:
        st_out += '\t' * len(open_curves)
        st_out += f'<{open_curves[-1]}>\n'
st_out += '\t' * (len(open_curves))
st_out += f'</{open_curves[-1]}>\n'
open_curves.pop(-1)
st_out += '\t' * (len(open_curves))
st_out += f'</{open_curves[0]}>\n'
st_out += '</schedule>'

with open('res_osn.xml', 'w') as file:
    print(st_out, file=file)

```

Получаем XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<schedule>
    <day_number>2</day_number>
    <week_number>11</week_number>
    <date>"2023-11-07"</date>
    <lessons>
        <id>lesson-206</id>
        <name>Математический анализ</name>
        <lesson_type>Лекции</lesson_type>
        <group>МАТ АН ПИИКТ 13</group>
        <time>
            <start>08:20</start>
            <end>09:50</end>
        </time>
        <teacher_name>Правдин Константин Владимирович</teacher_name>
        <place>
            <campus>Кронверкский пр., д.49, лит.А</campus>
            <room>Ауд.Orange Classroom (1229)</room>
            <distant>Очно - дистанционный</distant>

```



```

        </place>
    </lessons>
    <lessons>
        <id>lesson-210</id>
        <name>Математический анализ</name>
        <lesson_type>Практические занятия</lesson_type>
        <group>МАТ АН ПИИКТ 13.3</group>
        <time>
            <start>10:00</start>
            <end>11:30</end>
        </time>
        <teacher_name>Блейхер Оксана Владимировна</teacher_name>
        <place>
            <campus>Кронверкский пр., д.49, лит.А</campus>
            <room>Ауд.2430</room>
            <distant>Очно - дистанционный</distant>
        </place>
    </lessons>
    <lessons>
        <id>lesson-217</id>
        <name>Основы дискретной математики (базовый уровень)</name>
        <lesson_type>Лекции</lesson_type>
        <group>ДИСКР МАТ 2</group>
        <time>
            <start>11:40</start>
            <end>13:10</end>
        </time>
        <teacher_name>Поляков Владимир Иванович</teacher_name>
        <place>
            <campus>Кронверкский пр., д.49, лит.А</campus>
            <room>Ауд.2337</room>
            <distant>Очно - дистанционный</distant>
        </place>
    </lessons>
    <lessons>
        <id>lesson-220</id>
        <name>Основы дискретной математики (базовый уровень)</name>
        <lesson_type>Практические занятия</lesson_type>
        <group>ДИСКР МАТ 2.1</group>
        <time>
            <start>13:30</start>

```

```

        <end>15:00</end>
    </time>
    <teacher_name>Поляков Владимир Иванович</teacher_name>
    <place>
        <campus>Кронверкский пр., д.49, лит.А</campus>
        <room>Ауд.2337</room>
        <distant>Очно - дистанционный</distant>
    </place>
</lessons>
</schedule>

```

Дополнительное задание №1

Использованы 2 библиотеки, одна из которых переводит исходный YAML в словарь, а другая переводит словарь в XML

Представлен листинг на Python, чтение из файла in.yml, запись происходит в файл res_dop1.xml, также используются библиотеки yaml, xmltodict

```

import yaml
from yaml import SafeLoader
import xmltodict

yaml_file = open('in1.yml')
data_dict = yaml.load(yaml_file, Loader=SafeLoader)
print(data_dict)
out_file = open("res_dop1.xml", "w")
xml_res = xmltodict.unparse(data_dict, output=out_file)

```

Итоговый XML

```

<?xml version="1.0" encoding="utf-8"?>
<schedule><day_number>2</day_number><week_number>11</week_number><date>2023-
11-07</date><lessons><id>lesson-206</id><name>Математический
анализ</name><lesson_type>Лекции</lesson_type><group>МАТ АН ПИИКТ
13</group><time><start>08:20</start><end>09:50</end></time><teacher_name>Прав
дин Константин Владимирович</teacher_name><place><campus>Кронверкский пр.,
д.49, лит.А</campus><room>Ауд.Orange Classroom (1229)</room><distant>Очно -
дистанционный</distant></place></lessons><lessons><id>lesson-210</
id><name>Математический анализ</name><lesson_type>Практические

```

занятия</lesson_type><group>МАТ АН ПИИКТ
13.3</group><time><start>600</start><end>690</end></time><teacher_name>Блейхер Оксана Владимировна</teacher_name><place><campus>Кронверкский пр., д.49, лит.А</campus><room>Ауд.2430</room><distant>Очно - дистанционный</distant></place></lessons><lessons><id>lesson-217</id><name>Основы дискретной математики (базовый уровень)</name><lesson_type>Лекции</lesson_type><group>ДИСКР МАТ 2</group><time><start>700</start><end>790</end></time><teacher_name>Поляков Владимир Иванович</teacher_name><place><campus>Кронверкский пр., д.49, лит.А</campus><room>Ауд.2337</room><distant>Очно - дистанционный</distant></place></lessons><lessons><id>lesson-220</id><name>Основы дискретной математики (базовый уровень)</name><lesson_type>Практические занятия</lesson_type><group>ДИСКР МАТ 2.1</group><time><start>810</start><end>900</end></time><teacher_name>Поляков Владимир Иванович</teacher_name><place><campus>Кронверкский пр., д.49, лит.А</campus><room>Ауд.2337</room><distant>Очно - дистанционный</distant></place></lessons></schedule>

Написание кода для парсинга и конвертации с использованием готовых библиотек намного проще, быстрее и понятнее, чем путём замены метасимволов. При этом в качестве промежуточного результата мы также получаем python-словарь, что удобно использовать для дальнейшего перевода любой другой язык разметки (не только XML). Кроме этого готовая библиотека более универсальна, а решение с помощью замены метасимволов больше заточено под конкретный файл и конкретные данные.

Дополнительное задание №2

При переписывании исходного кода полностью сохранять изначальную логику работы я не стал, так как с помощью регулярных выражений можно сделать решение более лаконичным, коротким и удобным. При этом исчезает необходимость проходить по всему коду с помощью циклов, так как возможно использовать метод `re.sub()`.

Листинг на Python, данные берутся из файла `data.yml` и записываются в файл `res_dop2.xml`, используется стандартная библиотека для работы с регулярными выражениями — `re`:

```
import re

f = open('data.yml')
lines = [i.replace('\n', '') for i in f]
st_out = '\n'.join(lines)

st_out = re.sub(r"---", '<?xml version="1.0" encoding="UTF-8"?>\n<schedule>\n', st_out)

st_out = re.sub(r"(\w+): (.+)", r'<\1>\2</\1>', st_out)

for i in re.findall(r"(\s*\S*\n(\s{4}.)+)", st_out, flags=re.MULTILINE):
    st = ''.join(i[:1])
    old_st = st
    match = re.findall(r"(\S*):$", st, flags=re.MULTILINE)
    st = st.replace(f"{match[0]}:", f'<{match[0]}>')
    st += f'</{match[0]}>\n'
    st_out = st_out.replace(old_st, st, 1)

open_lists = re.findall(r"(\S*):$\n-", st_out, flags=re.MULTILINE)
st_out = re.sub(r"^-", f"</{open_lists[0]}>\n<{open_lists[0]}>\n", st_out, flags=re.MULTILINE)
st_out = re.sub(rf"({open_lists[0]}):\n</{open_lists[0]}>", '', st_out, flags=re.MULTILINE)
st_out += f"</{open_lists[0]}>\n</schedule>"

with open('res_dop2.xml', 'w') as f:
    print(st_out, file=f)
```

Полученный XML

```
<?xml version="1.0" encoding="UTF-8"?>
<schedule>

<day_number>2</day_number>
<week_number>11</week_number>
<date>"2023-11-07"</date>

<lessons>
  <id>lesson-206</id>
  <name>Математический анализ</name>
  <lesson_type>Лекции</lesson_type>
  <group>МАТ АН ПИИКТ 13</group>
  <time>
    <start>08:20</start>
    <end>09:50</end></time>

  <teacher_name>Правдин Константин Владимирович</teacher_name>
  <place>
    <campus>Кронверкский пр., д.49, лит.А</campus>
    <room>Ауд.Orange Classroom (1229)</room>
    <distant>Очно - дистанционный</distant></place>

</lessons>
<lessons>
  <id>lesson-210</id>
  <name>Математический анализ</name>
  <lesson_type>Практические занятия</lesson_type>
  <group>МАТ АН ПИИКТ 13.3</group>
  <time>
    <start>10:00</start>
    <end>11:30</end></time>

  <teacher_name>Блейхер Оксана Владимировна</teacher_name>
  <place>
    <campus>Кронверкский пр., д.49, лит.А</campus>
    <room>Ауд.2430</room>
    <distant>Очно - дистанционный</distant></place>

</lessons>
```

```

<lessons>
  <id>lesson-217</id>
  <name>Основы дискретной математики (базовый уровень)</name>
  <lesson_type>Лекции</lesson_type>
  <group>ДИСКР МАТ 2</group>
  <time>
    <start>11:40</start>
    <end>13:10</end></time>

  <teacher_name>Поляков Владимир Иванович</teacher_name>
  <place>
    <campus>Кронверкский пр., д.49, лит.А</campus>
    <room>Ауд.2337</room>
    <distant>Очно - дистанционный</distant></place>

</lessons>
<lessons>
  <id>lesson-220</id>
  <name>Основы дискретной математики (базовый уровень)</name>
  <lesson_type>Практические занятия</lesson_type>
  <group>ДИСКР МАТ 2.1</group>
  <time>
    <start>13:30</start>
    <end>15:00</end></time>

  <teacher_name>Поляков Владимир Иванович</teacher_name>
  <place>
    <campus>Кронверкский пр., д.49, лит.А</campus>
    <room>Ауд.2337</room>
    <distant>Очно - дистанционный</distant></place>
</lessons>
</schedule>

```

При таком исполнении чтение XML файла кажется более сложным, нежели при исполнении путём простой замены метасимволов, однако всё же намного проще полученного в первом дополнительном задании XML-файла в одну строчку. С использованием регулярных выражений оказалось удобнее не сохранять знаки табуляции, при этом это не влияет на валидность результата, так как формал XML не чувствителен к пробелам и знакам табуляции.

Дополнительное задание №3

Для выполнения этого задания я перевожу исходный YAML-файл (in1.yml) в словарь Python, используя для этого отдельную функцию, которая при обнаружении в текущей строке больше знаков табуляции, чем в предыдущей вызывает себя рекурсивно. Далее, также рекурсивно, из полученного словаря создаётся XML файл с помощью другой функции, при этом я создаю его без знаков табуляции и пробелов, так как для XML они не нужны. Поскольку для этого задания необходимы другие входные данные (2 дня недели, по 2 пары в каждом), исходный файл был немного изменён.

Исходный файл

```
schedule:
  first:
    day_number: 2
    week_number: 11
    date: "2023-11-07"
    lessons:
      - id: lesson-206
        name: Математический анализ
        lesson_type: Лекции
        group: МАТ АН ПИИКТ 13
        time:
          start: 08:20
          end: 09:50
        teacher_name: Правдин Константин Владимирович
        place:
          campus: Кронверкский пр., д.49, лит.А
          room: Ауд.Orange Classroom (1229)
          distant: Очно – дистанционный
      - id: lesson-210
        name: Математический анализ
        lesson_type: Практические занятия
        group: МАТ АН ПИИКТ 13.3
        time:
          start: 10:00
          end: 11:30
        teacher_name: Блейхер Оксана Владимировна
        place:
          campus: Кронверкский пр., д.49, лит.А
          room: Ауд.2430
          distant: Очно – дистанционный
```

```

second:
  day_number2: 3
  week_number2: 10
  date2: "2023-11-12"
  lessons:
    - id: lesson-217
      name: Основы дискретной математики (базовый уровень)
      lesson_type: Лекции
      group: ДИСКР МАТ 2
      time:
        start: 11:40
        end: 13:10
      teacher_name: Поляков Владимир Иванович
      place:
        campus: Кронверкский пр., д.49, лит.А
        room: Ауд.2337
        distant: Очно - дистанционный
    - id: lesson-220
      name: Основы дискретной математики (базовый уровень)
      lesson_type: Практические занятия
      group: ДИСКР МАТ 2.1
      time:
        start: 13:30
        end: 15:00
      teacher_name: Поляков Владимир Иванович
      place:
        campus: Кронверкский пр., д.49, лит.А
        room: Ауд.2337
        distant: Очно - дистанционный

```

Исходный код

Далее представлен листинг на Python с двумя функциями, описанными выше. Данные загружаются из файла in1.yml и записываются в файл res_dop3.xml:

```

def add_part(a, i, prev_tabs, st, open_dash):
    while st[i].replace(' ', '')[0] == '-':
        if st[i - 1].split(':')[1].replace(' ', '').replace('\n', '') == '':
            open_dash.append(st[i - 1].split(':')[0].replace(' ', ''))
            a[open_dash[-1]] = []
        st[i] = st[i].replace('-', ' ')
        dop_part = add_part({}, i, prev_tabs, st, open_dash)
        a[open_dash[-1]].append(dop_part[0])
        i = dop_part[1]

```



```

        if i >= len(st):
            break
    if i >= len(st):
        return a, i
    cur_tabs = st[i].count(' ')
    if cur_tabs - prev_tabs > 0:
        while True:
            cur_st = st[i].split(':')
            if cur_st[1].replace(' ', '') == '\n':
                dop_part = add_part({}, i + 1, prev_tabs, st, open_dash)
                a[cur_st[0].replace(' ', '')] = dop_part[0]
                i = dop_part[1]
            else:
                if cur_st[1][0] == ' ':
                    a[cur_st[0].replace(' ', '')] =
':'.join(cur_st[1:]).replace('\n', '')[1:]
                else:
                    a[cur_st[0].replace(' ', '')] =
':'.join(cur_st[1:]).replace('\n', ' ')
                i += 1
            prev_tabs = cur_tabs
        if i >= len(st):
            return a, i
        cur_tabs = st[i].count(' ')
        if not cur_tabs - prev_tabs == 0:
            return a, i
    elif cur_tabs - prev_tabs == 0:
        if st[i + 1].split(':')[0].replace(' ', '')[0] != '-':
            cur_st = st[i].split(':')
            a[cur_st[0]] = add_part({}, i + 1, cur_tabs, st, open_dash)[0]
            return a, i
        else:
            return add_part({}, i + 1, 0, st, open_dash)

def make_xml(collection, parent=None):
    if parent is None:
        xml_str = '\n'
        xml_str += make_xml(collection, parent='')
        return xml_str
    else:

```

```

xml_str = ''
for key, value in collection.items():
    xml_str += f"<{key}>"
    if isinstance(value, dict):
        xml_str += make_xml(value)
    elif isinstance(value, list):
        for item in value:
            xml_str += make_xml(item)
    else:
        xml_str += str(value)
    xml_str += f"</{key}>"
return xml_str

f = open("in1.yml")
tabs_prev = 0
st = f.readlines()
open_dash = []
data = add_part({}, 0, 0, st, open_dash)[0]
st_res = make_xml(data)
st_res = '<?xml version="1.0" encoding="UTF-8"?>' + st_res

with open("res_dop3.xml", 'w') as out_file:
    print(st_res, file=out_file)

```

Полученный XML

```

<?xml version="1.0" encoding="UTF-8"?>
<schedule>
<first>
<day_number>2</day_number><week_number>11</week_number><date>"2023-11-07"</
date><lessons>
<lessons>
<id>lesson 206</id><name>Математический
анализ</name><lesson_type>Лекции</lesson_type><group>МАТ АН ПИИКТ
13</group><time>
<start>08:20</start><end>09:50</end></time><teacher_name>Правдин Константин
Владимирович</teacher_name><place>
<campus>Кронверкский пр., д.49, лит.А</campus><room>Ауд.Orange Classroom
(1229)</room><distant>Очно - дистанционный</distant></place>

```

```

<id>lesson 210</id><name>Математический
анализ</name><lesson_type>Практические занятия</lesson_type><group>МАТ АН
ПИИКТ 13.3</group><time>
<start>10:00</start><end>11:30</end></time><teacher_name>Блейхер Оксана
Владимировна</teacher_name><place>
<campus>Кронверкский пр., д.49,
лит.А</campus><room>Ауд.2430</room><distant>Очно -
дистанционный</distant></place></lessons><second>
<day_number2>3</day_number2><week_number2>10</week_number2><date2>"2023-11-
12"</date2></lessons>
</lessons>
<id>lesson 217</id><name>Основы дискретной математики (базовый
уровень)</name><lesson_type>Лекции</lesson_type><group>ДИСКР МАТ
2</group><time>
<start>11:40</start><end>13:10</end></time><teacher_name>Поляков Владимир
Иванович</teacher_name><place>
<campus>Кронверкский пр., д.49,
лит.А</campus><room>Ауд.2337</room><distant>Очно -
дистанционный</distant></place>
<id>lesson 220</id><name>Основы дискретной математики (базовый
уровень)</name><lesson_type>Практические занятия</lesson_type><group>ДИСКР
МАТ 2.1</group><time>
<start>13:30</start><end>15:00</end></time><teacher_name>Поляков Владимир
Иванович</teacher_name><place>
<campus>Кронверкский пр., д.49,
лит.А</campus><room>Ауд.2337</room><distant>Очно -
дистанционный</distant></place></lessons></lessons></second></lessons><second
>
<day_number2>3</day_number2><week_number2>10</week_number2><date2>"2023-11-
12"</date2></lessons>
<id>lesson 220</id><name>Основы дискретной математики (базовый
уровень)</name><lesson_type>Практические занятия</lesson_type><group>ДИСКР
МАТ 2.1</group><time>
<start>13:30</start><end>15:00</end></time><teacher_name>Поляков Владимир
Иванович</teacher_name><place>
<campus>Кронверкский пр., д.49,
лит.А</campus><room>Ауд.2337</room><distant>Очно -
дистанционный</distant></place></lessons></second></first></schedule>

```

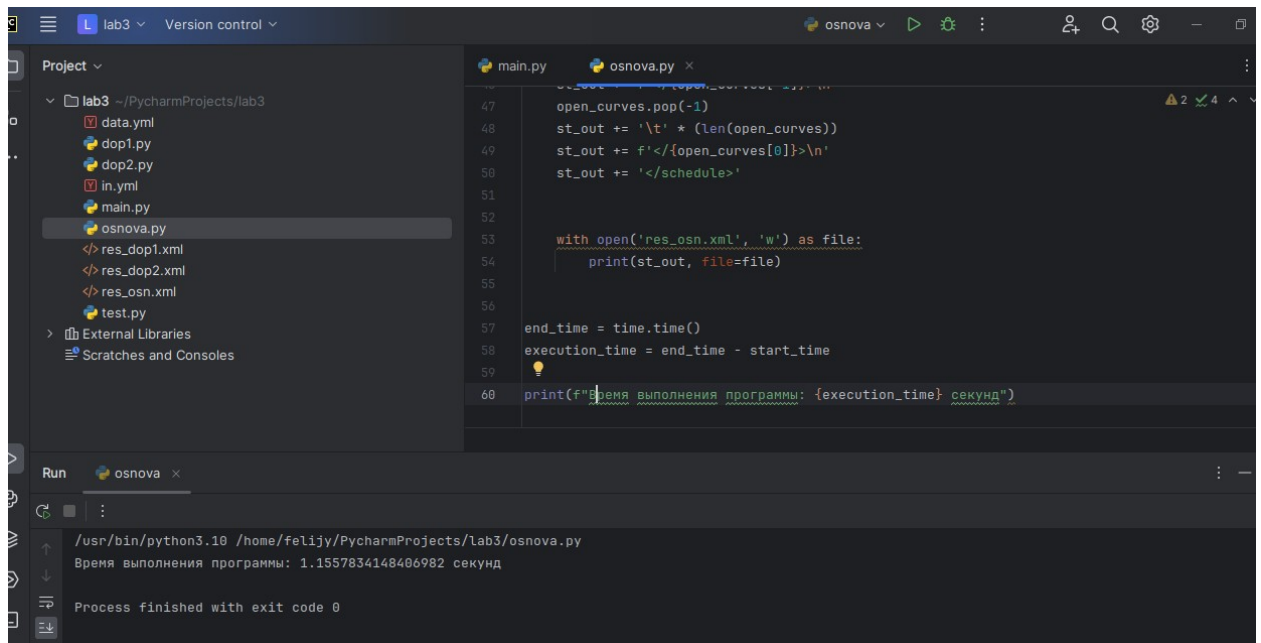
Данный способ парсинга и конвертации кажется одним из самых удачных и наиболее приближенным к решению в готовой библиотеке. Используется рекурсия и отдельные функции, что, на удивление, значительно уменьшает время работы программы. При этом данные, которые можно дать программе на вход, значительно шире, стало возможно использовать несколько дней недели по несколько пар в каждом. Также это решение выглядит наиболее удобным и понятным, нет неочевидных с первого взгляда циклов и условий.

Дополнительное задание №4

Для каждого задания (включая основное) будут приведены время выполнения программы 1000 раз. Программа выполнялась в цикле, для подсчёта использовалась встроенная библиотека `time`. В начале программы засекалось время `start_time`, в конце `stop_time`. Далее был реализован вывод результата, сколько времени потребовалось для выполнения 1000 раз программы. Также будут представлены скриншоты с выводом.

Основное задание

Время выполнения основного задания составило 1,55783 секунд. Результат представлен на Рисунке 1



```
47 open_curves.pop(-1)
48 st_out += '\t' * (len(open_curves))
49 st_out += f'</{open_curves[0]}>\n'
50 st_out += '</schedule>'
51
52
53 with open('res_osn.xml', 'w') as file:
54     print(st_out, file=file)
55
56
57 end_time = time.time()
58 execution_time = end_time - start_time
59
60 print(f"Время выполнения программы: {execution_time} секунд")
```

Run osnova x

/usr/bin/python3.10 /home/felijy/PycharmProjects/lab3/osnova.py
Время выполнения программы: 1.1557834148406982 секунд
Process finished with exit code 0

Рисунок 1: Время выполнения основного задания

Дополнительное задание №1

Время выполнения дополнительного задания №1 (про использование сторонних библиотек) составило 9,30449 секунд, что является самым большим и наиболее удивительным результатом. Результат представлен на Рисунке 2

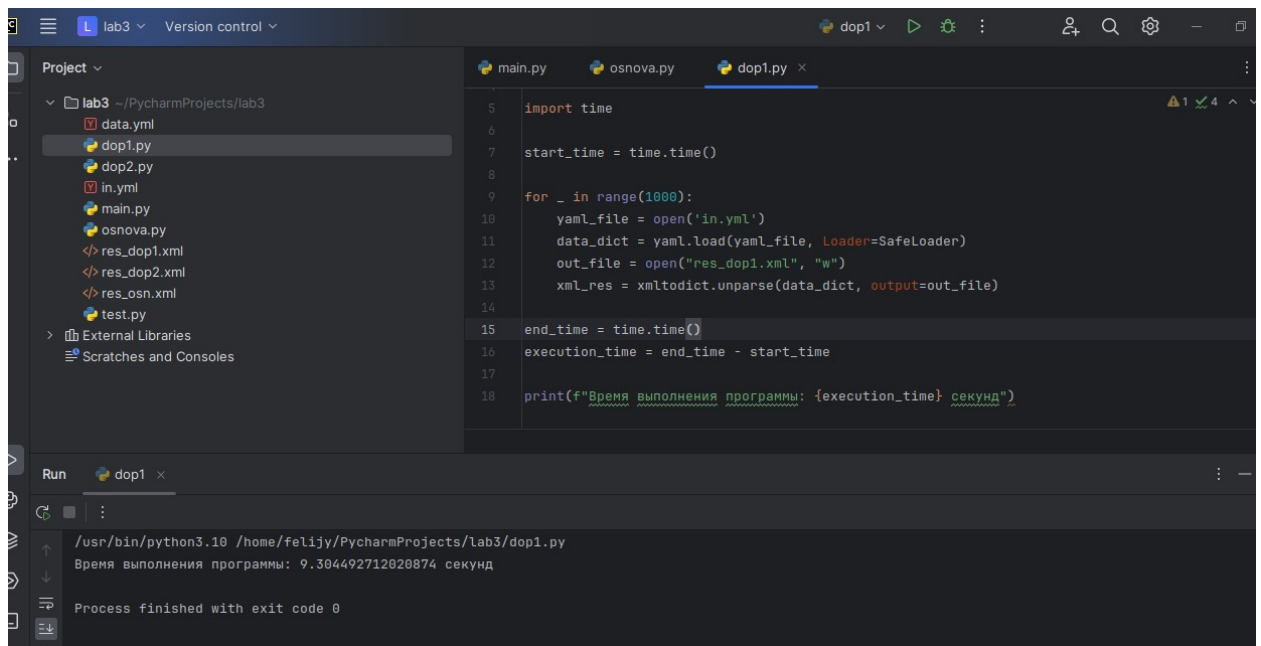


Рисунок 2: Время выполнения дополнительного задания №1

Дополнительное задание №2

Время выполнения дополнительного задания №2 (про использование регулярных выражений) составило 2,08834 секунд. Результат представлен на Рисунке 3

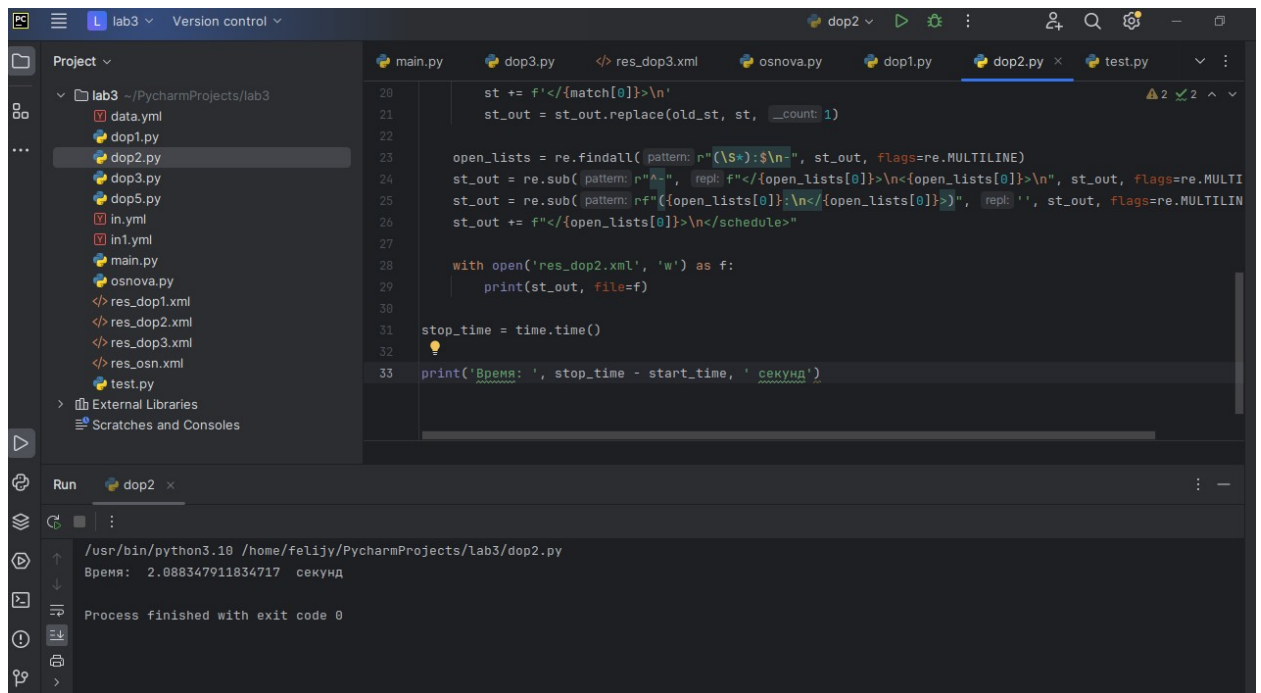


Рисунок 3: Время выполнения дополнительного задания №2

Дополнительное задание №3

Время выполнения дополнительного задания №3 (про использование словаря и дальнейшего преобразования из него в XML) составило 0,95446 секунд, что, к моему огромному удивлению, является лучшим результатом.

Рекурсионный самодельный метод решения оказался самым быстродейственным. Результат представлен на Рисунке 4

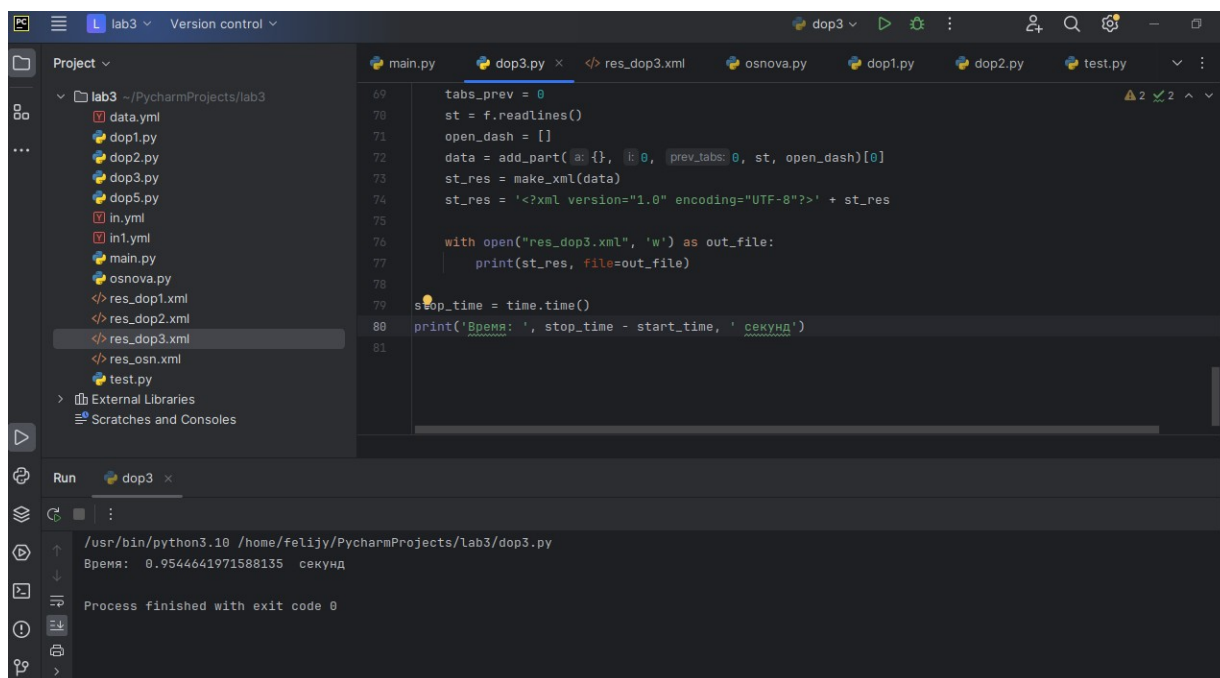


Рисунок 4: Время выполнения дополнительного задания №3

Дополнительное задание №5

Время выполнения дополнительного задания №5 (про конвертацию из YAML в CSV) составило 0,63528 секунд, однако это коневертация в файл совсем другого формата, поэтому я не учитывал его при нахождении лучшего способа решения задачи. Результат представлен на Рисунке 5

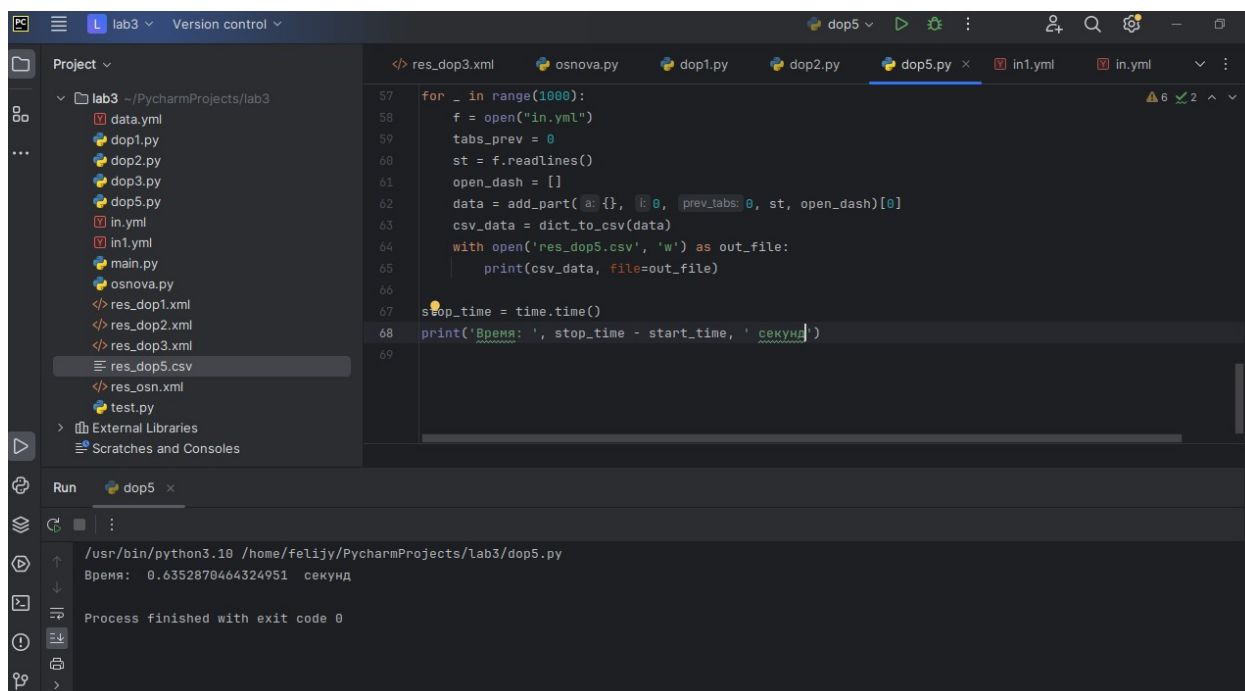


Рисунок 5: Время выполнения дополнительного задания №5

Дополнительное задание №5

Для решения этого задания было принято решение использовать наработки из дополнительного задания №3, так как оно оказалось наиболее эффективным. Поэтому функция формирования словаря была полностью скопирована оттуда, добавлена лишь функция преобразования его в формат CSV, листинг представлен на Python, данный загружаются из файла in.yml и записываются в файл res_dop5.csv:

```
def make_csv(data, parent_key='', csv_str=''):
    for key, value in data.items():
        if isinstance(value, dict):
            csv_str = make_csv(value, f'{parent_key}{key}.', csv_str)
        elif isinstance(value, list):
            for i, item in enumerate(value, start=1):
                csv_str = make_csv(item, f'{parent_key}{key}{i}.', csv_str)
        else:
            csv_str += f'{parent_key}{key},{value}\n'
    return csv_str

def add_part(a, i, prev_tabs, st, open_dash):
    while st[i].replace(' ', '')[0] == '-':
```



```

    if st[i - 1].split(':')[1].replace(' ', '').replace('\n', '') == '':
        open_dash.append(st[i - 1].split(':')[0].replace(' ', ''))
        a[open_dash[-1]] = []
    st[i] = st[i].replace('-', ' ')
    dop_part = add_part({}, i, prev_tabs, st, open_dash)
    a[open_dash[-1]].append(dop_part[0])
    i = dop_part[1]
    if i >= len(st):
        break
    if i >= len(st):
        return a, i
    cur_tabs = st[i].count(' ')
    if cur_tabs - prev_tabs > 0:
        while True:
            cur_st = st[i].split(' ')
            if cur_st[1].replace(' ', '').replace('\n', '') == '\n':
                dop_part = add_part({}, i + 1, prev_tabs, st, open_dash)
                a[cur_st[0].replace(' ', '')] = dop_part[0]
                i = dop_part[1]
            else:
                if cur_st[1][0] == ' ':
                    a[cur_st[0].replace(' ', '')] =
':'.join(cur_st[1:]).replace('\n', '')[1:]
                else:
                    a[cur_st[0].replace(' ', '')] =
':'.join(cur_st[1:]).replace('\n', ' ')
                i += 1
            prev_tabs = cur_tabs
            if i >= len(st):
                return a, i
            cur_tabs = st[i].count(' ')
            if not cur_tabs - prev_tabs == 0:
                return a, i
        elif cur_tabs - prev_tabs == 0:
            if st[i + 1].split(':')[0].replace(' ', '')[0] != '-':
                cur_st = st[i].split(' ')
                a[cur_st[0].replace(' ', '')] = add_part({}, i + 1, cur_tabs, st,
open_dash)[0]
                return a, i
            else:
                return add_part({}, i + 1, 0, st, open_dash)

```

```

f = open("in.yml")
tabs_prev = 0
st = f.readlines()
open_dash = []
data = add_part({}, 0, 0, st, open_dash)[0]
csv_data = make_csv(data)
with open('res_dop5.csv', 'w') as out_file:
    print(csv_data, file=out_file)

```

Полученный CSV

```

schedule.day_number,2
schedule.week_number,11
schedule.date,"2023-11-07"
schedule.lessons.lessons1.id,lesson 206
schedule.lessons.lessons1.name,Математический анализ
schedule.lessons.lessons1.lesson_type,Лекции
schedule.lessons.lessons1.group,МАТ АН ПИИКТ 13
schedule.lessons.lessons1.time.start,08:20
schedule.lessons.lessons1.time.end,09:50
schedule.lessons.lessons1.teacher_name,Правдин Константин
Владимирович
schedule.lessons.lessons1.place.campus,Кронверкский пр., д.49,
лит.А
schedule.lessons.lessons1.place.room,Ауд.Orange Classroom (1229)
schedule.lessons.lessons1.place.distant,Очно – дистанционный
schedule.lessons.lessons2.id,lesson 210
schedule.lessons.lessons2.name,Математический анализ
schedule.lessons.lessons2.lesson_type,Практические занятия
schedule.lessons.lessons2.group,МАТ АН ПИИКТ 13.3
schedule.lessons.lessons2.time.start,10:00
schedule.lessons.lessons2.time.end,11:30
schedule.lessons.lessons2.teacher_name,Блейхер Оксана
Владимировна
schedule.lessons.lessons2.place.campus,Кронверкский пр., д.49,
лит.А

```

```
schedule.lessons.lessons2.place.room,Ауд.2430
schedule.lessons.lessons2.place.distant,Очно - дистанционный
schedule.lessons.lessons3.id,lesson 217
schedule.lessons.lessons3.name,Основы дискретной математики
(базовый уровень)
schedule.lessons.lessons3.lesson_type,Лекции
schedule.lessons.lessons3.group,ДИСКР МАТ 2
schedule.lessons.lessons3.time.start,11:40
schedule.lessons.lessons3.time.end,13:10
schedule.lessons.lessons3.teacher_name,Поляков Владимир Иванович
schedule.lessons.lessons3.place.campus,Кронверкский пр., д.49,
лит.А
schedule.lessons.lessons3.place.room,Ауд.2337
schedule.lessons.lessons3.place.distant,Очно - дистанционный
schedule.lessons.lessons4.id,lesson 220
schedule.lessons.lessons4.name,Основы дискретной математики
(базовый уровень)
schedule.lessons.lessons4.lesson_type,Практические занятия
schedule.lessons.lessons4.group,ДИСКР МАТ 2.1
schedule.lessons.lessons4.time.start,13:30
schedule.lessons.lessons4.time.end,15:00
schedule.lessons.lessons4.teacher_name,Поляков Владимир Иванович
schedule.lessons.lessons4.place.campus,Кронверкский пр., д.49,
лит.А
schedule.lessons.lessons4.place.room,Ауд.2337
schedule.lessons.lessons4.place.distant,Очно - дистанционный
```

Формат CSV интересен своей структурой, необходимо для каждого элемента указывать как бы полный «путь» от самого начала структуры. Поскольку некоторые элементы (а именно lessons) у меня повторялись, а этого необходимо было избежать, пришлось использовать нумерацию, чтобы файл получился валидным.

Вывод

Был произведён парсинг и конвертация исходного файла формата YAML в XML и CSV. Наиболее быстрым оказался способ решения через самописную рекурсивную функцию, которая сначала создаёт словарь со всеми объектами, а затем делает из него файл XML, также рекурсивно. Наиболее же медленным оказалось решение с использованием сторонних библиотек, несмотря на то, что оно при этом наиболее простое, удобное и понятное. Также была выполнена конвертация в формат CSV.

Список используемых источников

1. <https://wtools.io/ru/validate-xml-online>
2. <https://wtools.io/ru/validate-yaml-online>
3. <https://wtools.io/ru/convert-yaml-to-xml>
4. <https://wtools.io/ru/convert-yaml-to-csv>
5. Лямин А.В., Череповская Е.Н. Объектно-ориентированное программирование. Компьютерный практикум. – СПб: Университет ИТМО, 2017. – 143 с. – Режим доступа: <https://books.ifmo.ru/file/pdf/2256.pdf>
6. <https://onlineyamltools.com/convert-yaml-to-xml>
7. <http://hilite.me>