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

**Лабораторная работа**

**Исследование протоколов,**

**форматов обмена информацией и языков разметки**

**документов**

Вариант: 22

Выполнил:  
Васильев Артём Евгеньевич  
Группа: 3119

Преподаватель:  
Рыбаков Степан Дмитриевич

Преподаватель-практик

факультета ПИиКТ

Санкт-Петербург, 2023 г.

**Оглавление**

**[Задание 3](#_Toc8717)**

**[Основные этапы вычисления 5](#_Toc16597)**

[Код на Python: 5](#_Toc8947)

[Исходный файл: 12](#_Toc15843)

[Результат работы основной программы: 13](#_Toc29478)

[Результат работы программы 1 доп. задания: 17](#_Toc9045)

[Результат работы программы 2 доп. задания: 18](#_Toc15442)

[Результат работы программы 3 доп. задания: 21](#_Toc26735)

[Результат работы программы 5 доп. задания: 25](#_Toc25548)

**[Итоги выполнения заданий: 26](#_Toc972)**

[Итог дополнительного задания 1 26](#_Toc31220)

[Итог дополнительного задания 2 26](#_Toc12215)

[Итог дополнительного задания 3 26](#_Toc13740)

[Итог дополнительного задания 4 26](#_Toc7747)

[Итог дополнительного задания 5 26](#_Toc25741)

**[Заключение 27](#_Toc813)**

**[Источники 28](#_Toc15811)**

# 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:

https://itmo.ru/ru/schedule/3/125598/raspisanie\_zanyatiy.htm

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

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

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. Объяснение должно быть отражено в отчёте.

Дополнительное задание №2

a) Переписать исходный код, добавив в него использование регулярных выражений.

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

1. Дополнительное задание № 3

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.

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

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

1. Дополнительное задание № 4

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

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

1. Дополнительное задание №5

a) 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.

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

1. Проверить, что все пункты задания выполнены и выполнены верно.
2. Написать отчёт о проделанной работе.

16. Подготовиться к устным вопросам на защите.

# Основные этапы вычисления

## Код на Python:

# 22. YAML -> XML; четверг  
  
*'''основное задание'''*def first():  
 with open("in.yaml", "r", encoding="UTF-8") as yaml\_file:  
 r = yaml\_file.read()  
 list\_key = None  
 xml\_result = ['<?xml version="1.0" encoding="UTF-8"?>\n']  
 d = {}  
 c = False  
 r = r.split("\n")  
 for el in r:  
 el = el.replace("&", "&amp;").replace("<", "&lt;").replace(">", "&gt;").replace("'", "&apos;").replace('"', "&quot;")  
 if ":" in el:  
 for i in range(len((d.keys()))):  
 if d and el.split(":")[0].count(" ") <= max(d.keys()) and el.split(":")[0].strip() != d[max(d.keys())]:  
 if list\_key != d[max(d.keys())].strip():  
 xml\_result.append(f"{' ' \* max(d.keys())}</{d[max(d.keys())].strip()}>\n")  
 d.pop(max(d.keys()))  
 d[el.split(":")[0].count(" ")] = el.split(":")[0].strip()  
 if el.strip().split("-")[0] == "":  
 if d[list(d.keys())[-1]] != list\_key:  
 xml\_result[-1] = xml\_result[-1].strip().replace(f"<{d[list(d.keys())[-1]]}>", " ")  
 list\_key = d[list(d.keys())[-1]]  
 if not c:  
 xml\_result.append(f"{' ' \* (el.split('-')[0].count(' '))}<{list\_key}>\n")  
 c = True  
 xml\_result[-1] += f"{' ' \* (el.split('-')[0].count(' ') + 2)}<{d[list(d.keys())[-1]]}>{''.join(el.strip().split('-')[1::])}</{d[list(d.keys())[-1]]}>\n"  
 else:  
 if c:  
 c = False  
 xml\_result.append(f"{' ' \* (el.split('-')[0].count(' ') + 2)}</{list\_key}>\n")  
 if "[" in el and "]" in el and "," in el:  
 list\_key = el.split(":")[0].strip()  
 for i in el.split(':')[1].replace("[", "").replace("]", "").split(","):  
 if i != "":  
 xml\_result[-1] += f"{' ' \* (el.split(':')[0].count(' ') + 2)}<{el.split(':')[0].strip()}>{i.strip()}</{el.split(':')[0].strip()}>\n"  
 else:  
 if ":" not in el and el.strip().split(":")[0] != list\_key:  
 xml\_result.append(el + "\n")  
 else:  
 if el.split(':')[1] != "" and el.split(':')[1] != " |":  
 xml\_result.append(' ' \* el.split(":")[0].count(" ") + "<" + el.strip().replace(":", f">\n {' ' \* (el.split(':')[0].count(' ') + 2)}").replace(  
 " |", "") + "\n")  
 else:  
 xml\_result.append(  
 ' ' \* el.split(":")[0].count(" ") + "<" + el.strip().replace(":", f">\n").replace(" |", ""))  
  
 for i in range(len(d)):  
 xml\_result.append(f"{' ' \* max(d.keys())}</{d[max(d.keys())].strip()}>\n")  
 d.pop(max(d.keys()))  
 # print(\*xml\_result)  
  
 with open("out\_1.xml", "w") as xml\_file:  
 for i in xml\_result:  
 xml\_file.write(i)  
  
  
'''доп 1'''  
import yaml  
from yaml import SafeLoader  
import xmltodict  
  
  
def second():  
 with open("in.yaml", "r", encoding="UTF-8") as yaml\_file:  
 python\_dict = yaml.load(yaml\_file, Loader=SafeLoader)  
  
 with open("out\_2.xml", "w") as xml\_file:  
 xmltodict.unparse(python\_dict, output=xml\_file)  
  
  
'''доп 2'''  
import re  
  
  
def third():  
 with open("in.yaml", "r", encoding="UTF-8") as yaml\_file:  
 r = yaml\_file.read()  
 list\_key = None  
 xml\_result = '<?xml version="1.0" encoding="UTF-8"?>\n'  
 d = {}  
 c = False  
 r = r.split("\n")  
 for el in r:  
 el = el.replace("&", "&amp;").replace("<", "&lt;").replace(">", "&gt;").replace("'", "&apos;").replace('"', "&quot;")  
 try: # сюда попадает, если есть что-то после:  
 key, val = el.split(":")  
 if val == " |":  
 val = ""  
 except ValueError: # сюда попадает, когда попадается просто текст  
 n = re.findall(r'\s', el)  
 if el.strip().split("-")[0] == "":  
 c = True  
 #print(list\_key, d[list(d.keys())[-1]])  
 if d[list(d.keys())[-1]] != list\_key:  
 xml\_result = xml\_result.strip().replace(f"<{d[list(d.keys())[-1]]}>", " ")  
 list\_key = d[list(d.keys())[-1]]  
 xml\_result += f"\n{' ' \* (el.split('-')[0].count(' ') + 2)}<{d[list(d.keys())[-1]].strip()}>{''.join(el.strip().split('-')[1::])}</{d[list(d.keys())[-1]].strip()}>"  
 else:  
 xml\_result += f"{' ' \* len(n)}{el.strip()}\n"  
 continue  
  
 if c:  
 xml\_result += "\n"  
 c = False  
  
 for i in range(len((d.keys()))): # закрываем блок, если след элемент по кол-ву пробелов ближе к началу строки  
 if d and key.count(" ") <= max(d.keys()) and key != d[max(d.keys())]:  
 if list\_key != d[max(d.keys())].strip():  
 xml\_result += f"{' ' \* max(d.keys())}</{d[max(d.keys())].strip()}>\n"  
 d.pop(max(d.keys()))  
  
 d[key.count(  
 " ")] = key # добавляем в словарь, если строка удовлетворяет шаблону <key>:<val>, чтобы потом закрыть </>  
 n = re.findall(r'\s', key)  
 if "[" in val and "]" in val and "," in val:  
 s = ""  
 c = 0  
 list\_key = key.strip()  
 for i in re.sub(r"[\[\]]", "", val).split(","):  
 if i != "":  
 if c == 0:  
 s += f"{' ' \* (len(n) + 2)}<{key.strip()}>{i.strip()}</{key.strip()}>\n"  
 else:  
 s += f"{' ' \* (len(n) + 2)}<{key.strip()}>{i.strip()}</{key.strip()}>\n"  
 c += 1  
 val = s  
 if list\_key == key.strip():  
 xml\_result += s  
  
 elif val != "":  
 xml\_result += f"{' ' \* len(n)}<{key.strip()}>\n{' ' \* (len(n) + 2)}{val.strip()}\n" # выводим элемент  
 else:  
 xml\_result += f"{' ' \* len(n)}<{key.strip()}>\n" # выводим элемент  
  
 for i in range(len(d)):  
 xml\_result += f"{' ' \* max(d.keys())}</{d[max(d.keys())].strip()}>\n"  
 d.pop(max(d.keys()))  
 # print(xml\_result)  
  
 with open("out\_3.xml", "w") as xml\_file:  
 xml\_file.write(xml\_result)  
  
  
'''доп 3'''  
  
  
def fourth():  
 with open("in.yaml", "r", encoding="UTF-8") as yaml\_file:  
 r = yaml\_file.read()  
 list\_key = None  
 xml\_result = '<?xml version="1.0" encoding="UTF-8"?>\n'  
 d = {}  
 c = False  
 r = r.split("\n")  
 for el in r:  
 el = el.replace("&", "&amp;").replace("<", "&lt;").replace(">", "&gt;").replace("'", "&apos;").replace('"', "&quot;")  
 try: # сюда попадает, если есть что-то после :  
 key, val = el.split(":")  
 if val == " |":  
 val = ""  
 except ValueError: # сюда попадает, когда попадается просто текст  
 key = el.strip()  
  
 if el.strip().split("-")[0] == "":  
 c = True  
 #print(list\_key, d[list(d.keys())[-1]])  
 if d[list(d.keys())[-1]] != list\_key:  
 xml\_result = xml\_result.strip().replace(f"<{d[list(d.keys())[-1]]}>", " ")  
 list\_key = d[list(d.keys())[-1]]  
 xml\_result += f"\n{' ' \* (el.split('-')[0].count(' ') + 2)}<{d[list(d.keys())[-1]].strip()}>{''.join(el.strip().split('-')[1::])}</{d[list(d.keys())[-1]].strip()}>"  
 else:  
 xml\_result += f"{' ' \* el.count(' ')}{el.strip()}\n"  
 continue  
  
 if c:  
 c = False  
 xml\_result += "\n"  
  
 for i in range(len((d.keys()))): # закрываем блок, если след элемент по кол-ву пробелов ближе к началу строки  
 if d and key.count(" ") <= max(d.keys()) and key != d[max(d.keys())]:  
 if list\_key != d[max(d.keys())].strip():  
 xml\_result += f"{' ' \* max(d.keys())}</{d[max(d.keys())].strip()}>\n"  
 d.pop(max(d.keys()))  
  
 d[key.count(  
 " ")] = key # добавляем в словарь, если строка удовлетворяет шаблону <key>:<val>, чтобы потом закрыть </>  
  
 if "[" in val and "]" in val and "," in val:  
 s = ""  
 c = 0  
 list\_key = key.strip()  
 for i in val.replace("[", "").replace("]", "").split(","):  
 if i != "":  
 if c == 0:  
 s += f"<{key.strip()}>{i.strip()}</{key.strip()}>\n"  
 else:  
 s += f"{' ' \* (key.count(' ') + 2)}<{key.strip()}>{i.strip()}</{key.strip()}>\n"  
 c += 1  
 val = s  
 if list\_key == key.strip():  
 xml\_result += s  
 elif val != "":  
 xml\_result += f"{' ' \* key.count(' ')}<{key.strip()}>\n{' ' \* (key.count(' ') + 2)}{val.strip()}\n" # выводим элемент  
 else:  
 xml\_result += f"{' ' \* key.count(' ')}<{key.strip()}>\n" # выводим элемент  
  
 for i in range(len(d)):  
 xml\_result += f"{' ' \* max(d.keys())}</{d[max(d.keys())].strip()}>\n"  
 d.pop(max(d.keys()))  
 # print(xml\_result)  
  
 with open("out\_4.xml", "w") as xml\_file:  
 xml\_file.write(xml\_result)  
  
  
'''доп 4'''  
import time  
  
l = []  
for i in range(100):  
 t = time.perf\_counter()  
 first()  
 l.append(time.perf\_counter() - t)  
print("Время выполнения 1 программы:", sum(l) / len(l))  
  
l = []  
for i in range(100):  
 t = time.perf\_counter()  
 second()  
 l.append(time.perf\_counter() - t)  
print("Время выполнения 2 программы:", sum(l) / len(l))  
  
l = []  
for i in range(100):  
 t = time.perf\_counter()  
 third()  
 l.append(time.perf\_counter() - t)  
print("Время выполнения 3 программы:", sum(l) / len(l))  
  
l = []  
for i in range(50):  
 t = time.perf\_counter()  
 time.sleep(1)  
 fourth()  
 l.append(time.perf\_counter() - t)  
print("Время выполнения 4 программы:", sum(l) / len(l))  
  
'''доп 5'''  
  
with open("in.yaml", "r", encoding="UTF-8") as yaml\_file:  
 r = yaml\_file.read()  
  
csv\_result = {}  
r = r.split("\n")  
l\_key = ""  
for el in r:  
 try: # сюда попадает, если есть что-то после :  
 key, val = el.split(":")  
 val = val.strip()  
 if val != "" and val != "|":  
 if "[" in val and "]" in val and "," in val:  
 val = val.replace("[", "").replace("]", "")  
  
 if key.strip() == "hour" or key.strip() == "minutes":  
 key = f"{l\_key}\_{key.strip()}"  
 if "hour" in key and "minutes" in key:  
 key = key.replace("\_hour", "")  
  
 if key.strip() not in csv\_result.keys():  
 csv\_result[key.strip()] = [val]  
 else:  
 csv\_result[key.strip()].append(val)  
 l\_key = key.strip()  
 except ValueError: # сюда попадает, когда попадается просто текст  
 if l\_key in csv\_result.keys():  
 csv\_result[l\_key].append(el.strip())  
 else:  
 csv\_result[l\_key] = [el.strip()]  
  
res = [[], [], []]  
  
for i in csv\_result.keys():  
 res[0].append(f"{csv\_result[i][0]};")  
 res[1].append(f"{csv\_result[i][1]};")  
 res[2].append(f"{csv\_result[i][2]};")  
  
with open("out\_5.csv", "w") as csv\_file:  
 csv\_file.write(  
 f'''{str(list(csv\_result.keys())).replace("[", "").replace("]", "").replace("'", "").replace(", ", ";")}\n''')  
 for i in res:  
 csv\_file.write(f'''{str(i).replace("[", "").replace(";']", "").replace("', ", "").replace("'", "")}\n''')

## Исходный файл:

schedule:

lesson1:

time:

start:

hour: 10

minutes: 00

end:

hour: 13

minutes: 10

weeks: [2, 4, 6, 8, 10, 12, 14, 16]

place:

address: |

Кронверкский пр., д.49, лит.А

room: |

1320 АУД.

name: |

ОПД (Практика)

week: |

чётная неделя

teacher: |

Райла Мартин

kind: |

Очный

lesson2:

time:

start:

hour: 13

minutes: 30

end:

hour: 15

minutes: 00

weeks: [2, 4, 6, 8, 10, 12, 14, 16]

place:

address: |

Кронверкский пр., д.49, лит.А

room: |

1229 АУД.

name: |

Реформы и реформаторы в истории России (Лекция)

week: |

чётная неделя

teacher: |

Вычеров Дмитрий Александрович

kind: |

Очный

lesson3:

time:

start:

hour: 15

minutes: 20

end:

hour: 16

minutes: 50

weeks: [2, 4, 6, 8, 10, 12, 14, 16]

place:

address: |

Кронверкский пр., д.49, лит.А

room: |

2412 АУД.

name: |

Реформы и реформаторы в истории России (Практика)

week: |

чётная неделя

teacher: |

Вычеров Дмитрий Александрович

kind: |

Очный

## Результат работы основной программы:

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

<schedule>

<lesson1>

<time>

<start>

<hour>

10

</hour>

<minutes>

00

</minutes>

</start>

<end>

<hour>

13

</hour>

<minutes>

10

</minutes>

</end>

</time>

<weeks>2</weeks>

<weeks>4</weeks>

<weeks>6</weeks>

<weeks>8</weeks>

<weeks>10</weeks>

<weeks>12</weeks>

<weeks>14</weeks>

<weeks>16</weeks>

<place>

<address>

Кронверкский пр., д.49, лит.А

</address>

<room>

1320 АУД.

</room>

</place>

<name>

ОПД (Практика)

</name>

<week>

чётная неделя

</week>

<teacher>

Райла Мартин

</teacher>

<kind>

Очный

</kind>

</lesson1>

<lesson2>

<time>

<start>

<hour>

13

</hour>

<minutes>

30

</minutes>

</start>

<end>

<hour>

15

</hour>

<minutes>

00

</minutes>

</end>

</time>

<weeks>2</weeks>

<weeks>4</weeks>

<weeks>6</weeks>

<weeks>8</weeks>

<weeks>10</weeks>

<weeks>12</weeks>

<weeks>14</weeks>

<weeks>16</weeks>

<place>

<address>

Кронверкский пр., д.49, лит.А

</address>

<room>

1229 АУД.

</room>

</place>

<name>

Реформы и реформаторы в истории России (Лекция)

</name>

<week>

чётная неделя

</week>

<teacher>

Вычеров Дмитрий Александрович

</teacher>

<kind>

Очный

</kind>

</lesson2>

<lesson3>

<time>

<start>

<hour>

15

</hour>

<minutes>

20

</minutes>

</start>

<end>

<hour>

16

</hour>

<minutes>

50

</minutes>

</end>

</time>

<weeks>2</weeks>

<weeks>4</weeks>

<weeks>6</weeks>

<weeks>8</weeks>

<weeks>10</weeks>

<weeks>12</weeks>

<weeks>14</weeks>

<weeks>16</weeks>

<place>

<address>

Кронверкский пр., д.49, лит.А

</address>

<room>

2412 АУД.

</room>

</place>

<name>

Реформы и реформаторы в истории России (Практика)

</name>

<week>

чётная неделя

</week>

<teacher>

Вычеров Дмитрий Александрович

</teacher>

<kind>

Очный

</kind>

</lesson3>

</schedule>

## Результат работы программы 1 доп. задания:

<?xml version="1.0" encoding="utf-8"?>

<schedule><lesson1><time><start><hour>10</hour><minutes>0</minutes></start><end><hour>13</hour><minutes>10</minutes></end></time><weeks>2</weeks><weeks>4</weeks><weeks>6</weeks><weeks>8</weeks><weeks>10</weeks><weeks>12</weeks><weeks>14</weeks><weeks>16</weeks><place><address>Кронверкский пр., д.49, лит.А

</address><room>1320 АУД.

</room></place><name>ОПД (Практика)

</name><week>чётная неделя

</week><teacher>Райла Мартин

</teacher><kind>Очный

</kind></lesson1><lesson2><time><start><hour>13</hour><minutes>30</minutes></start><end><hour>15</hour><minutes>0</minutes></end></time><weeks>2</weeks><weeks>4</weeks><weeks>6</weeks><weeks>8</weeks><weeks>10</weeks><weeks>12</weeks><weeks>14</weeks><weeks>16</weeks><place><address>Кронверкский пр., д.49, лит.А

</address><room>1229 АУД.

</room></place><name>Реформы и реформаторы в истории России (Лекция)

</name><week>чётная неделя

</week><teacher>Вычеров Дмитрий Александрович

</teacher><kind>Очный

</kind></lesson2><lesson3><time><start><hour>15</hour><minutes>20</minutes></start><end><hour>16</hour><minutes>50</minutes></end></time><weeks>2</weeks><weeks>4</weeks><weeks>6</weeks><weeks>8</weeks><weeks>10</weeks><weeks>12</weeks><weeks>14</weeks><weeks>16</weeks><place><address>Кронверкский пр., д.49, лит.А

</address><room>2412 АУД.

</room></place><name>Реформы и реформаторы в истории России (Практика)

</name><week>чётная неделя

</week><teacher>Вычеров Дмитрий Александрович

</teacher><kind>Очный</kind></lesson3></schedule>

## Результат работы программы 2 доп. задания:

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

<schedule>

<lesson1>

<time>

<start>

<hour>

10

</hour>

<minutes>

00

</minutes>

</start>

<end>

<hour>

13

</hour>

<minutes>

10

</minutes>

</end>

</time>

<weeks>2</weeks>

<weeks>4</weeks>

<weeks>6</weeks>

<weeks>8</weeks>

<weeks>10</weeks>

<weeks>12</weeks>

<weeks>14</weeks>

<weeks>16</weeks>

<place>

<address>

Кронверкский пр., д.49, лит.А

</address>

<room>

1320 АУД.

</room>

</place>

<name>

ОПД (Практика)

</name>

<week>

чётная неделя

</week>

<teacher>

Райла Мартин

</teacher>

<kind>

Очный

</kind>

</lesson1>

<lesson2>

<time>

<start>

<hour>

13

</hour>

<minutes>

30

</minutes>

</start>

<end>

<hour>

15

</hour>

<minutes>

00

</minutes>

</end>

</time>

<weeks>2</weeks>

<weeks>4</weeks>

<weeks>6</weeks>

<weeks>8</weeks>

<weeks>10</weeks>

<weeks>12</weeks>

<weeks>14</weeks>

<weeks>16</weeks>

<place>

<address>

Кронверкский пр., д.49, лит.А

</address>

<room>

1229 АУД.

</room>

</place>

<name>

Реформы и реформаторы в истории России (Лекция)

</name>

<week>

чётная неделя

</week>

<teacher>

Вычеров Дмитрий Александрович

</teacher>

<kind>

Очный

</kind>

</lesson2>

<lesson3>

<time>

<start>

<hour>

15

</hour>

<minutes>

20

</minutes>

</start>

<end>

<hour>

16

</hour>

<minutes>

50

</minutes>

</end>

</time>

<weeks>2</weeks>

<weeks>4</weeks>

<weeks>6</weeks>

<weeks>8</weeks>

<weeks>10</weeks>

<weeks>12</weeks>

<weeks>14</weeks>

<weeks>16</weeks>

<place>

<address>

Кронверкский пр., д.49, лит.А

</address>

<room>

2412 АУД.

</room>

</place>

<name>

Реформы и реформаторы в истории России (Практика)

</name>

<week>

чётная неделя

</week>

<teacher>

Вычеров Дмитрий Александрович

</teacher>

<kind>

Очный

</kind>

</lesson3>

</schedule>

## Результат работы программы 3 доп. задания:

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

<schedule>

<lesson1>

<time>

<start>

<hour>

10

</hour>

<minutes>

00

</minutes>

</start>

<end>

<hour>

13

</hour>

<minutes>

10

</minutes>

</end>

</time>

<weeks>2</weeks>

<weeks>4</weeks>

<weeks>6</weeks>

<weeks>8</weeks>

<weeks>10</weeks>

<weeks>12</weeks>

<weeks>14</weeks>

<weeks>16</weeks>

<place>

<address>

Кронверкский пр., д.49, лит.А

</address>

<room>

1320 АУД.

</room>

</place>

<name>

ОПД (Практика)

</name>

<week>

чётная неделя

</week>

<teacher>

Райла Мартин

</teacher>

<kind>

Очный

</kind>

</lesson1>

<lesson2>

<time>

<start>

<hour>

13

</hour>

<minutes>

30

</minutes>

</start>

<end>

<hour>

15

</hour>

<minutes>

00

</minutes>

</end>

</time>

<weeks>2</weeks>

<weeks>4</weeks>

<weeks>6</weeks>

<weeks>8</weeks>

<weeks>10</weeks>

<weeks>12</weeks>

<weeks>14</weeks>

<weeks>16</weeks>

<place>

<address>

Кронверкский пр., д.49, лит.А

</address>

<room>

1229 АУД.

</room>

</place>

<name>

Реформы и реформаторы в истории России (Лекция)

</name>

<week>

чётная неделя

</week>

<teacher>

Вычеров Дмитрий Александрович

</teacher>

<kind>

Очный

</kind>

</lesson2>

<lesson3>

<time>

<start>

<hour>

15

</hour>

<minutes>

20

</minutes>

</start>

<end>

<hour>

16

</hour>

<minutes>

50

</minutes>

</end>

</time>

<weeks>2</weeks>

<weeks>4</weeks>

<weeks>6</weeks>

<weeks>8</weeks>

<weeks>10</weeks>

<weeks>12</weeks>

<weeks>14</weeks>

<weeks>16</weeks>

<place>

<address>

Кронверкский пр., д.49, лит.А

</address>

<room>

2412 АУД.

</room>

</place>

<name>

Реформы и реформаторы в истории России (Практика)

</name>

<week>

чётная неделя

</week>

<teacher>

Вычеров Дмитрий Александрович

</teacher>

<kind>

Очный

</kind>

</lesson3>

</schedule>

## Результат работы программы 5 доп. задания:

start\_hour;start\_minutes;end\_hour;end\_minutes;weeks;address;room;name;week;teacher;kind

10;00;13;10;2, 4, 6, 8, 10, 12, 14, 16;Кронверкский пр., д.49, лит.А;1320 АУД.;ОПД (Практика);чётная неделя;Райла Мартин;Очный

13;30;15;00;2, 4, 6, 8, 10, 12, 14, 16;Кронверкский пр., д.49, лит.А;1229 АУД.;Реформы и реформаторы в истории России (Лекция);чётная неделя;Вычеров Дмитрий Александрович;Очный

15;20;16;50;2, 4, 6, 8, 10, 12, 14, 16;Кронверкский пр., д.49, лит.А;2412 АУД.;Реформы и реформаторы в истории России (Практика);чётная неделя;Вычеров Дмитрий Александрович;Очный

# Итоги выполнения заданий:

## Итог дополнительного задания 1

В получившемся файле отсутствуют символы табуляции. В остальном он идентичен файлу, получившемуся в основном задании. Библиотека производит преобразования, аналогичные исходному коду.

## Итог дополнительного задания 2

Получившийся файл идентичен файлу, получившемуся в основном задании, потому что кроме использования регулярных выражений, ничего не изменилось.

## Итог дополнительного задания 3

Получившийся файл идентичен файлу, получившемуся в основном задании, потому что реализация не критично различается

## Итог дополнительного задания 4

|  |  |
| --- | --- |
| Время выполнения программ | |
| Основное задание | 0.0007641434669494629 |
| Доп. задание 1 | 0.003906986713409424 |
| Доп. задание 2 | 0.0006803202629089355 |
| Доп. задание 3 | 0.0006174373626708984 |

## Итог дополнительного задания 5

Исходный XML файл был преобразован в файл с расширением CSV. Особенности формата CSV в том, что данный формат предназначен для представления табличных данных. Столбцы разделены между собой разделителем (любой символ, чаще всего “;”).

# Заключение

В ходе выполнения данной лабораторной работы, я познакомился с YAML, XML и CSV разметкой. Узнал особенности каждой из них и закрепил написание программ, преобразующих файлы одного типа в файлы других типов. Также закрепил работу с регулярными выражениями и научился использовать библиотеки для преобразования YAML файлов в XML.

# Источники

1. Чтение и запись файлов XML с помощью Pandas - <https://rukovodstvo.net/posts/id_659/>
2. Python YAML: How to Load, Read, and Write YAML - <https://python.land/data-processing/python-yaml>
3. Регулярные выражения в Python от простого к сложному - <https://habr.com/ru/articles/349860/>