Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

**Факультет программной инженерии и компьютерной техники**

**Информатика**

Лабораторная работа №4

«Исследование протоколов, форматов обмена информацией и языков разметки документов

Выполнил: Герасимов Артём Кириллович

Группа: P3108

Вариант: 6

Преподаватель: Малышева Татьяна Алексеевна

Санкт-Петербург

2021

Оглавление

[Задание 2](#_Toc88830745)

[Ход работы 3](#_Toc88830746)

[Исходный файл в формате YAML 3](#_Toc88830747)

[Обязательное задание 4](#_Toc88830748)

[Дополнительное задание №1 6](#_Toc88830749)

[Дополнительное задание №2 7](#_Toc88830750)

[Дополнительное задание №3 8](#_Toc88830751)

[Дополнительное задание №4 9](#_Toc88830752)

[Вывод 12](#_Toc88830753)

[Список литературы 12](#_Toc88830754)

Задание

* **Обязательное задание** (позволяет набрать до 65 процентов от максимального числа баллов БаРС за данную лабораторную):
* Написать программу на языке Python 3, которая бы осуществляла парсинг и конвертацию исходного файла в новый.
* Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.
* **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.
* Сравнить полученные результаты и объяснить их сходство/различие.
* **Дополнительное задание №2** (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную):
* Переписать исходный код, добавив в него использование регулярных выражений.
* Сравнить полученные результаты и объяснить их сходство/различие.
* **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 десятикратное время выполнения парсинга + конвертации в цикле.
* Проанализировать полученные результаты и объяснить их сходство/различие.
* **Дополнительное задание №4** (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную):
* 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.
* Проанализировать полученные результаты, объяснить особенности использованного формата.

Ход работы

Исходный файл в формате YAML

timetable:  
 tuesday:  
 lesson1:  
 subject: "Информатика (ЛАБ)"  
 teacher: "Малышева Татьяна Алексеевна"  
 time: "10:00-11:30"  
 classroom: "306"  
 location: "Кронверкский пр., д.49, лит.А"  
 week: "Чётная неделя"  
 format: "Очно-дистанционный"  
 lesson2:  
 subject: "Информатика (ЛАБ)"  
 teacher: "Малышева Татьяна Алексеевна"  
 time: "11:40-13:10"  
 classroom: "306"  
 location: "Кронверкский пр., д.49, лит.А"  
 week: "Чётная неделя"  
 format: "Очно-дистанционный"  
 lesson3:  
 subject: "Дискретная Математика (ЛЕК)"  
 teacher: "Поляков Владимир Иванович"  
 time: "13:30-15:00"  
 classroom: "285"  
 location: "Кронверкский пр., д.49, лит.А"  
 week: "Чётная неделя"  
 format: "Очно-дистанционный"

Обязательное задание

import time  
# модуль для работы со временем  
  
startTime = time.perf\_counter()  
# начало отсчёта времени  
  
with open('timetable.yml', encoding='utf-8') as inputFile:  
 yamlFile = inputFile.readlines()  
  
subjects = []; teachers = []; times = []; classrooms = []; locations = []; weeks = []; formats = []  
# списки, где будут лежать элементы расписания (названия предметов в subjects, адреса в locations и т.д.)  
  
for line in yamlFile:  
 line = line.replace(' ', '').replace('\n', '').replace(' ', '', 1).replace('"', '')  
 # замена всех отступов и переносов строк, а также кавычек (если есть)  
 key, value = line.split(':', 1)  
 if key == 'subject':  
 subjects.append(value)  
 if key == 'teacher':  
 teachers.append(value)  
 if key == 'time':  
 times.append(value)  
 if key == 'classroom':  
 classrooms.append(value)  
 if key == 'location':  
 locations.append(value)  
 if key == 'week':  
 weeks.append(value)  
 if key == 'format':  
 formats.append(value)  
 # заполнение списков (элементов расписания)  
  
lessons = []  
lessonCount = len(subjects)  
for i in range(lessonCount):  
 i += 1  
 # прибавляем 1, чтобы не было lesson0  
 lessons.append(f'lesson{i}')  
# счёт кол-ва пар  
  
xmlFile = "<timetable>\n\t<tuesday>\n\t\t"  
for n in range(lessonCount):  
 xmlFile += f'<{lessons[n]}>\n\t\t\t'  
 xmlFile += f'<subject>{subjects[n]}</subject>\n\t\t\t'  
 xmlFile += f'<teacher>{teachers[n]}</teacher>\n\t\t\t'  
 xmlFile += f'<time>{times[n]}</time>\n\t\t\t'  
 xmlFile += f'<classroom>{classrooms[n]}</classroom>\n\t\t\t'  
 xmlFile += f'<location>{locations[n]}</location>\n\t\t\t'  
 xmlFile += f'<week>{weeks[n]}</week>\n\t\t\t'  
 xmlFile += f'<format>{formats[n]}</format>\n\t\t'  
 # xml += f'</{lessons[n]}>\n\t\t'  
 if n == lessonCount - 1:  
 xmlFile += f'</{lessons[n]}>\n\t'  
 # если это последняя итерация цикла, то сделать 1 отступ, если нет, то 2  
 else:  
 xmlFile += f'</{lessons[n]}>\n\t\t'  
xmlFile += '</tuesday>\n'  
xmlFile += '</timetable>'  
  
with open('timetable.xml', 'w', encoding='utf-8') as outputFile:  
 outputFile.write(xmlFile)  
  
endTime = time.perf\_counter()  
# конец отсчёта времени  
finalTime = round(endTime - startTime, 8)  
print(f'Время выполнения программы {finalTime} секунд.')

**Результат работы:**

<timetable>  
 <tuesday>  
 <lesson1>  
 <subject>Информатика (ЛАБ)</subject>  
 <teacher>Малышева Татьяна Алексеевна</teacher>  
 <time>10:00-11:30</time>  
 <classroom>306</classroom>  
 <location>Кронверкский пр., д.49, лит.А</location>  
 <week>Чётная неделя</week>  
 <format>Очно-дистанционный</format>  
 </lesson1>  
 <lesson2>  
 <subject>Информатика (ЛАБ)</subject>  
 <teacher>Малышева Татьяна Алексеевна</teacher>  
 <time>11:40-13:10</time>  
 <classroom>306</classroom>  
 <location>Кронверкский пр., д.49, лит.А</location>  
 <week>Чётная неделя</week>  
 <format>Очно-дистанционный</format>  
 </lesson2>  
 <lesson3>  
 <subject>Дискретная Математика (ЛЕК)</subject>  
 <teacher>Поляков Владимир Иванович</teacher>  
 <time>13:30-15:00</time>  
 <classroom>285</classroom>  
 <location>Кронверкский пр., д.49, лит.А</location>  
 <week>Чётная неделя</week>  
 <format>Очно-дистанционный</format>  
 </lesson3>  
 </tuesday>  
</timetable>

Дополнительное задание №1

import time  
# модуль для работы со временем  
import yaml  
# модуль, который позволяет открыть файл YAML как словарь Питона  
import dict2xml  
# модуль который конвертирует словарь Питона в XML  
  
startTime = time.perf\_counter()  
# начало отсчёта времени  
  
with open('timetable.yml', encoding='utf-8') as inputFile:  
 yamlFile = yaml.safe\_load(inputFile)  
  
xmlFile = dict2xml.dict2xml(yamlFile, indent='\t')  
  
with open('timetable.xml', 'w', encoding='utf-8') as outputFile:  
 outputFile.write(xmlFile)  
  
endTime = time.perf\_counter()  
# конец отсчёта времени  
finalTime = round(endTime - startTime, 8)  
print(f'Время выполнения программы {finalTime} секунд.')

**Результат работы:**

<timetable>  
 <tuesday>  
 <lesson1>  
 <classroom>306</classroom>  
 <format>Очно-дистанционный</format>  
 <location>Кронверкский пр., д.49, лит.А</location>  
 <subject>Информатика (ЛАБ)</subject>  
 <teacher>Малышева Татьяна Алексеевна</teacher>  
 <time>10:00-11:30</time>  
 <week>Чётная неделя</week>  
 </lesson1>  
 <lesson2>  
 <classroom>306</classroom>  
 <format>Очно-дистанционный</format>  
 <location>Кронверкский пр., д.49, лит.А</location>  
 <subject>Информатика (ЛАБ)</subject>  
 <teacher>Малышева Татьяна Алексеевна</teacher>  
 <time>11:40-13:10</time>  
 <week>Чётная неделя</week>  
 </lesson2>  
 <lesson3>  
 <classroom>285</classroom>  
 <format>Очно-дистанционный</format>  
 <location>Кронверкский пр., д.49, лит.А</location>  
 <subject>Дискретная Математика (ЛЕК)</subject>  
 <teacher>Поляков Владимир Иванович</teacher>  
 <time>13:30-15:00</time>  
 <week>Чётная неделя</week>  
 </lesson3>  
 </tuesday>  
</timetable>

Единственное различие – это то, что при использовании библиотеки dict2xml, значения сортируются по алфавиту, т.к. это было задумано автором

Дополнительное задание №2

import time  
# модуль для работы со временем  
import re  
# модуль для работы с регулярными выражениями  
  
startTime = time.perf\_counter()  
# начало отсчёта времени  
  
with open('timetable.yml', encoding='utf-8') as inputFile:  
 yamlFile = inputFile.readlines()  
  
subjects = []; teachers = []; times = []; classrooms = []; locations = []; weeks = []; formats = []  
# списки, где будут лежать элементы расписания (названия предметов в subjects, адреса в locations и т.д.)  
lessons = []  
# список, где лежит кол-во пар  
  
for line in yamlFile:  
 line = line.replace(' ', '').replace('\n', '').replace(' ', '', 1).replace('"', '')  
 if re.search(r'subject', line):  
 line = re.sub(r'\w+:', '', line)  
 subjects.append(line)  
 if re.search(r'teacher', line):  
 line = re.sub(r'\w+:', '', line)  
 teachers.append(line)  
 if re.search(r'time:', line):  
 line = re.sub(r'\w+:', '', line, 1)  
 times.append(line)  
 if re.search(r'classroom', line):  
 line = re.sub(r'\w+:', '', line)  
 classrooms.append(line)  
 if re.search(r'location', line):  
 line = re.sub(r'\w+:', '', line)  
 locations.append(line)  
 if re.search(r'week', line):  
 line = re.sub(r'\w+:', '', line)  
 weeks.append(line)  
 if re.search(r'format', line):  
 line = re.sub(r'\w+:', '', line)  
 formats.append(line)  
 if re.search(r'lesson\d', line):  
 line = re.sub(r':', '', line)  
 lessons.append(line)  
  
xmlFile = '<timetable>\n\t<tuesday>\n\t\t'  
for n in range(len(lessons)):  
 xmlFile += f'<{lessons[n]}>\n\t\t\t'  
 xmlFile += f'<subject>{subjects[n]}</subject>\n\t\t\t'  
 xmlFile += f'<teacher>{teachers[n]}</teacher>\n\t\t\t'  
 xmlFile += f'<time>{times[n]}</time>\n\t\t\t'  
 xmlFile += f'<classroom>{classrooms[n]}</classroom>\n\t\t\t'  
 xmlFile += f'<location>{locations[n]}</location>\n\t\t\t'  
 xmlFile += f'<week>{weeks[n]}</week>\n\t\t\t'  
 xmlFile += f'<format>{formats[n]}</format>\n\t\t'  
 # xml += f'</{lessons[n]}>\n\t\t'  
 if n == len(lessons) - 1:  
 xmlFile += f'</{lessons[n]}>\n\t'  
 # если это последняя итерация цикла, то сделать 1 отступ, если нет, то 2  
 else:  
 xmlFile += f'</{lessons[n]}>\n\t\t'  
xmlFile += '</tuesday>\n'  
xmlFile += '</timetable>'  
  
with open('timetable.xml', 'w', encoding='utf-8') as outputFile:  
 outputFile.write(xmlFile)  
  
endTime = time.perf\_counter()  
# конец отсчёта времени  
finalTime = round(endTime - startTime, 8)  
print(f'Время выполнения программы {finalTime} секунд.')

**Результат работы:**

<timetable>  
 <tuesday>  
 <lesson1>  
 <subject>Информатика (ЛАБ)</subject>  
 <teacher>Малышева Татьяна Алексеевна</teacher>  
 <time>10:00-11:30</time>  
 <classroom>306</classroom>  
 <location>Кронверкский пр., д.49, лит.А</location>  
 <week>Чётная неделя</week>  
 <format>Очно-дистанционный</format>  
 </lesson1>  
 <lesson2>  
 <subject>Информатика (ЛАБ)</subject>  
 <teacher>Малышева Татьяна Алексеевна</teacher>  
 <time>11:40-13:10</time>  
 <classroom>306</classroom>  
 <location>Кронверкский пр., д.49, лит.А</location>  
 <week>Чётная неделя</week>  
 <format>Очно-дистанционный</format>  
 </lesson2>  
 <lesson3>  
 <subject>Дискретная Математика (ЛЕК)</subject>  
 <teacher>Поляков Владимир Иванович</teacher>  
 <time>13:30-15:00</time>  
 <classroom>285</classroom>  
 <location>Кронверкский пр., д.49, лит.А</location>  
 <week>Чётная неделя</week>  
 <format>Очно-дистанционный</format>  
 </lesson3>  
 </tuesday>  
</timetable>

Дополнительное задание №3

Дольше всего работает программа, где используются готовые библиотеки, так как она производит полноценный рекурсивный парсинг YAML, в то время как две остальные программы выбирают только нужные свойства, и парсинг производится в один проход.

Программа, где используются регулярные выражения работает дольше исходной программы, так как регулярные выражения более затратны из-за проверки исходной строки на соответствие шаблону.

Так как размер файла невелик, то разница несущественная.

Дополнительное задание №4

with open('timetable.yml', encoding='utf-8') as inputFile:  
 yamlFile = inputFile.readlines()  
  
subjects = []; teachers = []; times = []; classrooms = []; locations = []; weeks = []; formats = []  
# списки, где будут лежать элементы расписания (названия предметов в subjects, адреса в locations и т.д.)  
  
for line in yamlFile:  
 line = line.replace(' ', '').replace('\n', '').replace(' ', '', 1).replace('"', '')  
 # замена всех отступов и переносов строк, а также кавычек (если есть)  
 key, value = line.split(':', 1)  
 if key == 'subject':  
 subjects.append(value)  
 if key == 'teacher':  
 teachers.append(value)  
 if key == 'time':  
 times.append(value)  
 if key == 'classroom':  
 classrooms.append(value)  
 if key == 'location':  
 locations.append(value)  
 if key == 'week':  
 weeks.append(value)  
 if key == 'format':  
 formats.append(value)  
 # заполнение списков (элементов расписания)  
  
lessons = []  
lessonCount = len(subjects)  
for i in range(lessonCount):  
 i += 1  
 # прибавляем 1, чтобы не было lesson0  
 lessons.append(f'lesson{i}')  
# счёт кол-ва пар  
  
csvFile = ''  
for n in range(lessonCount):  
 csvFile += f'timetable.tuesday.{lessons[n]}.subject;'  
 csvFile += f'timetable.tuesday.{lessons[n]}.teacher;'  
 csvFile += f'timetable.tuesday.{lessons[n]}.time;'  
 csvFile += f'timetable.tuesday.{lessons[n]}.classroom;'  
 csvFile += f'timetable.tuesday.{lessons[n]}.location;'  
 csvFile += f'timetable.tuesday.{lessons[n]}.week;'  
 csvFile += f'timetable.tuesday.{lessons[n]}.format;'  
 if n == lessonCount - 1:  
 csvFile = csvFile[:-1]  
 csvFile += '\n'  
 # если это последняя итерация цикла, то удалить полсдений символ (;) и сделать перенос строки  
# составление 1-ой строчки  
  
for n in range(lessonCount):  
 csvFile += f'{subjects[n]};'  
 csvFile += f'{teachers[n]};'  
 csvFile += f'{times[n]};'  
 csvFile += f'{classrooms[n]};'  
 csvFile += f'{locations[n]};'  
 csvFile += f'{weeks[n]};'  
 csvFile += f'{formats[n]};'  
 if n == lessonCount - 1:  
 csvFile = csvFile[:-1]  
# составление 2-ой строчки  
  
with open('timetable.csv', 'w', encoding='utf-8') as outputFile:  
 outputFile.write(csvFile)

**Результат работы:**

timetable.tuesday.lesson1.subject;timetable.tuesday.lesson1.teacher;timetable.tuesday.lesson1.time;timetable.tuesday.lesson1.classroom;timetable.tuesday.lesson1.location;timetable.tuesday.lesson1.week;timetable.tuesday.lesson1.format;timetable.tuesday.lesson2.subject;timetable.tuesday.lesson2.teacher;timetable.tuesday.lesson2.time;timetable.tuesday.lesson2.classroom;timetable.tuesday.lesson2.location;timetable.tuesday.lesson2.week;timetable.tuesday.lesson2.format;timetable.tuesday.lesson3.subject;timetable.tuesday.lesson3.teacher;timetable.tuesday.lesson3.time;timetable.tuesday.lesson3.classroom;timetable.tuesday.lesson3.location;timetable.tuesday.lesson3.week;timetable.tuesday.lesson3.format  
Информатика (ЛАБ);Малышева Татьяна Алексеевна;10:00-11:30;306;Кронверкский пр., д.49, лит.А;Чётная неделя;Очно-дистанционный;Информатика (ЛАБ);Малышева Татьяна Алексеевна;11:40-13:10;306;Кронверкский пр., д.49, лит.А;Чётная неделя;Очно-дистанционный;Дискретная Математика (ЛЕК);Поляков Владимир Иванович;13:30-15:00;285;Кронверкский пр., д.49, лит.А;Чётная неделя;Очно-дистанционный

Файл неудобен для чтения, но его можно открыть в Microsoft Excel:





**Рисунок 1 - создатель URI, URL, HTTP и HTML Тим Бернерс-Ли**

Вывод

Во время выполнения лабораторной работы я узнал о языках разметки XML и YAML, научился с ними работать и переводить один формат в другой с помощью средств языка Python. Научился работать с библиотекой time для измерения времени, и некоторыми библиотеками для автоматического парсинга и конвертации.

Список литературы

1. Балакшин П.В., Соснин В.В., Калинин И.В., Малышева Т.А., Раков С.В., Рущенко Н.Г., Дергачев А.М. Информатика: лабораторные работы и тесты. – СПб: Университет ИТМО, 2019. – 56 с. - Режим доступа: <https://books.ifmo.ru/file/pdf/2464.pdf>.
2. 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>.