

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

Мегафакультет компьютерных технологий и управления
Факультет программной инженерии и компьютерной техники

Отчёт
по лабораторной работе №4
«Исследование протоколов, форматов обмена информацией
и языков разметки документов»
по дисциплине «Информатика»

Вариант 10

Студентка: Богданова Мария Михайловна, группа Р3118
Преподаватель: Рыбаков Степан Дмитриевич

Санкт-Петербург
2022

Содержание

1	Задание	2
2	Ход работы	3
2.1	Обязательное задание	3
2.2	Дополнительное задание №1	6
2.3	Дополнительное задание №2	7
2.4	Дополнительное задание №3	9
3	Вывод	10
4	Список использованной литературы	11

1 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 языков разметки/форматов JSON, YAML, XML.
4. Понять устройство страницы с расписанием для своей группы:
<http://itmo.ru/ru/schedule/0/P3118/schedule.htm>
5. 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.
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. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML–файлов.
7. 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.
 - с) Сравнить полученные результаты и объяснить их сходство/различие.
8. 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.
9. Дополнительное задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
 - а) Используя свою исходную программу из обязательного задания, программу из дополнительного задания №1 и программу из дополнительного задания №2, сравнить стократное время выполнения парсинга + конвертации в цикле.
 - б) Проанализировать полученные результаты и объяснить их сходство/различие.
10. Дополнительное задание №4 (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.

2 Ход работы

2.1 Обязательное задание

```
timetable:
  day: Вторник
  lesson1:
    time: 8:20-9:50
    room: Акт. зал
    subject: Информатика (лек)
    teacher: Балакшин Павел Валерьевич
    location: ул.Ломоносова, д.9, лит. М
    format: Очно-дистанционный
    weeks: 3, 5, 7, 9, 11, 13, 15, 17
  lesson2:
    time: 10:00-11:30
    room: Акт. зал
    subject: Основы профессиональной деятельности (лек)
    teacher: Клименков С.В.
    location: ул.Ломоносова, д.9, лит. М
    format: Очно-дистанционный
    weeks: 3, 5, 7, 9, 11, 13, 15, 17
  lesson3:
    time: 11:40-13:10
    room: 1223
    subject: Программирование (лек)
    teacher: Письмак А.Е.
    location: ул.Ломоносова, д.9, лит. М
    format: Очно-дистанционный
    weeks: 3, 5, 7, 9, 11, 13, 15, 17
  lesson4:
    time: 8:20-9:50
    room: нет
    subject: Информатика (лек)
    teacher: Балакшин Павел Валерьевич
    location: дома
    format: Дистанционный
    weeks: 5
  lesson5:
    time: 10:00-11:30
    room: нет
    subject: Основы профессиональной деятельности (лек)
    teacher: Клименков С.В.
    location: дома
    format: Дистанционный
    weeks: 5
  lesson6:
    time: 11:40-13:10
    room: 1223
    subject: Программирование (лек)
    teacher: Письмак А.Е.
    location: дома
    format: Дистанционный
    weeks: 5
```

Код решения основного задания:

```
import time

obj = [str(i) for i in range(0,10)]
t_start = time.perf_counter()
yml_obj = 'day time place subject lesson teacher location format weeks'.split(' ')

def match_lesson(s):
    word = ''
    for j in range(len(s)):
        word = word + s[j]
        if word == 'lesson' and s[j+1] in obj:
            return True
    return False

class Timetable:
    def __init__(self):
        self.lessons = []
        self.day = ""

def parse_line(line):
    line = line.replace("  ", "").replace('\n', '')
    key, word = line.split(":", 1)
    return key, word

def parse_file(file):
    lines = file.readlines()
    schedule = Timetable()
    lesson = dict()
    timetable_start = False
    last_key = ""
    for line in lines:
        if timetable_start:
            if line.count("  ", 0, 2) == 0:
                break
            else:
                line = line.replace("\n", "").replace("  ", "").replace(": ", ":")
                if line.replace(' ', '').startswith('-'):
                    lesson[last_key].append(line.replace(' ', '').replace('-', ''))
                elif "day" in line:
                    key, word = parse_line(line)
                    schedule.day = word
                elif match_lesson(line) and len(lesson) != 0:
                    schedule.lessons.append(lesson)
                    lesson = dict()
                else:
                    key, word = parse_line(line)
                    if word == '':
                        word = []
                        last_key = key
                    if key in yml_obj:
```

```

        lesson[key] = word
    if "timetable" in line:
        timetable_start = True

    schedule.lessons.append(lesson)
    return schedule

def file_to_xml(file):
    schedule = parse_file(file)
    xml = "<timetable>\n"
    xml+=f"\t<day>{schedule.day}</day>\n"
    for i in range(len(schedule.lessons)):
        lesson = schedule.lessons[i]
        xml += f"\t<lesson{i+1}>\n"
        for p in lesson:
            if isinstance(lesson[p],list):
                for x in lesson[p]:
                    xml += f"\t\t<{p}>{x}</{p}>\n"
                continue
            xml += f"\t\t<{p}>{lesson[p]}</{p}>\n"

        xml += f"\t</lesson{i+1}>\n"
    xml += "</timetable>"
    return xml

def convert_file(input_file_name, output_file_name, show = False):
    input_file = open(input_file_name + ".yaml", 'r', encoding='utf-8')
    output_file = open(output_file_name + ".xml", 'w', encoding = "utf-8")
    xml = file_to_xml(input_file)
    output_file.write(xml)
    input_file.close()
    output_file.close()
    if show:
        print(xml)

convert_file("timetable","timetable",True)

start_time = 0
for n in range(10):
    t_start = time.perf_counter()
    convert_file("timetable","timetable")
print("time: ",time.perf_counter() - t_start)

```

Вывод программы - расписание в формате XML:

```
<timetable>
  <day>Вторник </day>
  <lesson1>
    <time>8:20-9:50 </time>
    <place>Акт. зал </room>
    <subject>Информатика (лек) </subject>
    <teacher>Балашкин Павел Валерьевич </teacher>
    <location>ул.Ломоносова, д.9, лит. М </location>
    <format>Очно-дистанционный </format>
    <weeks>3, 5, 7, 9, 11, 13, 15, 17 </weeks>
  </lesson1>
  <lesson2>
    <time>10:00-11:30 </time>
    <place>Акт. зал </place>
    <subject>Основы профессиональной деятельности (лек) </subject>
    <teacher>Клименков С.В. </teacher>
    <location>ул.Ломоносова, д.9, лит. М </location>
    <format>Очно-дистанционный </format>
    <weeks>3, 5, 7, 9, 11, 13, 15, 17 </weeks>
  </lesson2>
  <lesson3>
    <time>11:40-13:10 </time>
    <place>1223 </place>
    <subject>Программирование (лек) </subject>
    <teacher>Письмак А.Е. </teacher>
    <location>ул.Ломоносова, д.9, лит. М </location>
    <format>Очно-дистанционный </format>
    <weeks>3, 5, 7, 9, 11, 13, 15, 17 </weeks>
  </lesson3>
  <lesson4>
    <time>8:20-9:50 </time>
    <place>нет </place>
    <subject>Информатика (лек) </subject>
    <teacher>Балашкин Павел Валерьевич </teacher>
    <location>дома </location>
    <format>Дистанционный </format>
    <weeks>5 </weeks>
  </lesson4>
  <lesson5>
    <time>10:00-11:30 </time>
    <place>нет </place>
    <subject>Основы профессиональной деятельности (лек) </subject>
    <teacher>Клименков С.В. </teacher>
    <location>дома </location>
    <format>Дистанционный </format>
    <weeks>5 </weeks>
  </lesson5>
  <lesson6>
    <time>11:40-13:10 </time>
    <place>1223 </place>
    <subject>Программирование (лек) </subject>
    <teacher>Письмак А.Е. </teacher>
    <location>дома </location>
    <format>Дистанционный </format>
    <weeks>5 </weeks>
  </lesson6>
</timetable>
```

2.2 Дополнительное задание №1

Для выполнения этого задания я использовала библиотеки `safe_load` и `dict2xml`.

Код программы:

```
from dict2xml import dict2xml
from yaml import safe_load

def convert_file(input_file_name, output_file_name, show = False):
    dict = {}
    with open(f"{input_file_name}.yaml", "r", encoding='utf-8') as file:
        dict = safe_load(file)
    with open(f"{output_file_name}.xml", "w", encoding='utf-8') as file:
        file.write(dict2xml(dict))

if __name__ == "__main__":
    import time
    start_time = time.perf_counter()
    for n in range(10):
        convert_file("timetable", "timetable")
    print("time: ", time.perf_counter() - start_time)
```

Вывод данной программы соответствует выводу программы, реализованной в первом задании, единственное различие можно заметить в последней строке - изменилось время реализации кода (time: 0.0006699000005028211 -> time: 0.04540479998104274, см. рис 1.)

2.3 Дополнительное задание №2

При выполнении данного задания я использовала основной код, но некоторые функции в нем заменила регулярными выражениями:

```
import re

class Timetable:
    def __init__(self):
        self.lessons = []
        self.day = ""

def parse_file(file):
    schedule = Timetable()
    timetable_start = False
    key_value = re.compile(r"\s*(\w+):\s*(.*)")
    for line in file.readlines():
        line=line[:-1]
        if timetable_start:
            if re.fullmatch(r"(\s)+day:\s*(\w*)",line):
                word = key_value.search(line).group(2)
                schedule.day = word
            elif re.fullmatch(r"\s*lesson\d+:",line):
                schedule.lessons.append({})
            else:
                groups = key_value.search(line)
                schedule.lessons[-1][groups.group(1)] = groups.group(2)
        elif "timetable" in line:
            timetable_start = True
    return schedule

def file_to_xml(file):
    schedule = parse_file(file)
    xml = "<timetable>\n"
    xml+=f"\t<day>{schedule.day}</day>\n"
    for i in range(len(schedule.lessons)):
        lesson = schedule.lessons[i]
        xml += f"\t<lesson{i+1}>\n"
        for p in lesson:
            xml += f"\t\t<p>{lesson[p]}</p>\n"
        xml += f"\t</lesson{i+1}>\n"
    xml += "</timetable>"
    return xml

def convert_file(input_file_name, output_file_name):
    with open(f"{input_file_name}.yaml", "r", encoding='utf-8') as file:
        xml = file_to_xml(file)
    with open(f"{output_file_name}.xml", "w", encoding='utf-8') as file:
        file.write(xml)

if __name__ == "__main__":
    import time
```



```
t_start = time.perf_counter()
convert_file("timetable", "timetable")
print("time: ", time.perf_counter() - t_start)
```

Содержимое вывода данной программы соответствует содержимому выводов предыдущих программ за исключением времени (time: 0.001284400001168251)

```
===== RESTART: C:\Users\fergee\Desktop\lab4\inf_lab4_task1.py =====
Squeezed text (57 lines).
time: 0.0006614000012632459

===== RESTART: C:\Users\fergee\Desktop\lab4\inf_lab4_task2.py =====
time: 0.04356410002219491

===== RESTART: C:\Users\fergee\Desktop\lab4\lab4_task3.py =====
Squeezed text (51 lines).
time: 0.014723000000230968
```

Рис. 1: Время выполнения всех 3-х программ

2.4 Дополнительное задание №3

Как я уже упоминала выше, во всех решениях я использовала библиотеку time для того, чтобы засечь время выполнения программ и сравнить их между собой. Я написала цикл, чтобы получить среднее время выполнения.

Таблица со сравнением полученных данных:

Способ	Среднее время выполнения программы
• • Основное задание (без исп. сторонних библиотек и RE)	0.49 с.
Доп. задание №1 (С исп. библиотек safe_load и dict2xml)	4.21 с.
Доп. задание №2 (С исп. RE)	0.59 с.

Программа, написанная без использования сторонних библиотек и регулярных выражений, оказалась самой быстрой. Дольше всего работает программа, написанная при выполнении 1-го дополнительного задания, так как она производит рекурсивный парсинг YAML в отличие от двух других программ, в которых происходит отбор по конкретным заданным критериям. Программа с использованием регулярных выражений работает дольше первой, поскольку постоянно происходит проверка данных на соответствие заданному шаблону, однако время выполнения 1-ой и 3-ей программы мало отличается в силу небольшого объема кода. **Самый быстрый способ выделен в таблице серо-буро-малиновым (в подозрительную крапинку) цветом.**

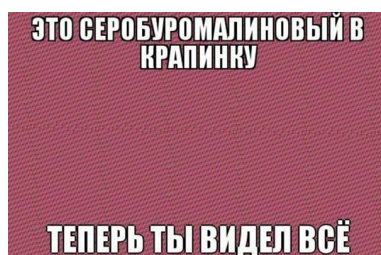


Рис. 2: светло-серо-буро-малиновый в крапинку цвет

3 Вывод

В ходе выполнения данной лабораторной работы я изучила форму Бэкуса-Наура, особенности языков разметки/форматов JSON, YAML, XML, научилась формировать файлы в этих форматах, писать парсер и конвертировать файлы из одного формата в другой.

L^AT_EX

4 Список использованной литературы

1. Лямин А.В., Череповская Е.Н. Объектно-ориентированное программирование. Компьютерный практикум. – СПб: Университет ИТМО, 2017. – 143 с. – Режим доступа: <https://books.ifmo.ru/> (дата обращения: 28.11.2022)
2. MSiter - учебник XML [электронный ресурс]. – Режим доступа: <https://msiter.ru/> / (дата обращения: 28.11.2022)
3. CoderLessons - учебник YAML [электронный ресурс]. – Режим доступа: <https://coderlessons.com/> (дата обращения: 28.11.2022)