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

Факультет программной инженерии и компьютерной техники

Образовательная программа

«Системное и прикладное программное обеспечение»

Дисциплина «Информатика»

**Отчет**

**Лабораторная работа №4**

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

**Вариант №34**

Выполнил:

Ерохин Егор Геннадьевич

Группа: P3111

Проверил:

Доцент факультета ПИиКТ

Малышева Татьяна Алексеевна

Г. Санкт-Петербург, 2024 г.

Оглавление

[Задание 3](#_Toc182931777)

[Основные этапы вычисления 5](#_Toc182931778)

[Обязательное задание 5](#_Toc182931779)

[Дополнительное задание №1 6](#_Toc182931780)

[Дополнительное задание №2 7](#_Toc182931781)

[Дополнительное задание №3 7](#_Toc182931782)

[Дополнительное задание №4 8](#_Toc182931783)

[Дополнительное задание №5 9](#_Toc182931784)

[Заключение 10](#_Toc182931785)

[Список литературы 10](#_Toc182931786)

# **Задание**

1. Определить номер варианта как остаток деления на 36 последних двух цифр своего идентификационного номера в ISU: например, 125598 / 36 = 26. В случае, если в оба указнных дня недели нет занятий, то увеличить номер варианта на восемь. В случае, если занятий нет и в новом наборе дней, то продолжать увеличивать 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 хотя бы в одной из выбранных дней 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 или любом другом, 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).
10. Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
11. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.
12. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
13. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
14. Переписать исходный код, добавив в него использование регулярных выражений.
15. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
16. Дополнительное задание №3 (позволяет набрать +25 процентов от максимального числа баллов БаРС за данную лабораторную).
17. 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.
18. Проверку осуществить как минимум для расписания с двумя учебными днями по два занятия в каждом.
19. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
20. Дополнительное задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).
21. Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле.
22. Проанализировать полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
23. Дополнительное задание №5 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).
24. 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.
25. Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте.
26. Проверить, что все пункты задания выполнены и выполнены верно.
27. Написать отчёт о проделанной работе.
28. Подготовиться к устным вопросам на защите.

Таблица 1. Вариант 34

|  |  |  |  |
| --- | --- | --- | --- |
| **№ варианта** | **Исходный формат** | **Результирующий формат** | **Дни недели** |
| 34 | YAML | XML | Понедельник, суббота |

# Основные этапы вычисления

## Обязательное задание

Исходный YAML файл:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/data/Input.yaml>

Исходный код:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/main_task/yaml_to_xml.py>

Результат:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/main_task/Output.xml>

## Дополнительное задание №1

Исходный код:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/additional_task1/yaml_to_xml_libs.py>

Результат:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/additional_task1/Output.xml>

В коде используется библиотека yaml для чтения YAML файла и библиотека xml.etree.ElementTree для превращение python dict в XML файл. Результаты отличаются тем, что библиотека для элементов списка добавляет тэги <item>, а мой код добавляет тэг названия списка (Например <week>). Ещё библиотека xml.etree.ElementTree требует добавить общий тэг (Например <root>), куда поместит всё содержимое.

## Дополнительное задание №2

Исходный код:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/additional_task2/yaml_to_xml_regex.py>

Результат:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/additional_task2/Output.xml>

Результаты не отличаются, так регулярные выражения в коде просто заменяют некоторые проверки.

## Дополнительное задание №3

Исходный код:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/additional_task3/yaml_to_xml_pro.py>

Результат:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/additional_task3/Output.xml>

Результаты не отличаются, однако данный код способен обрабатывать более сложные YAML файлы.

## Дополнительное задание №4

Исходный код:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/additional_task4/times.py>

Результат:

Основное задание - 0.0555690000182949

Дополнительное задание №1 - 0.715122599969618

Дополнительное задание №2 - 0.10988559998804703

Дополнительное задание №3 - 0.030522799992468208

Быстрее всего работает код из задания №3, так как в нем используется более рациональный подход через словари. Далее идет код из основного задания. Следом на ним код из задания №2, он работает дольше, т.к. регулярные выражения требуют посимвольного сравнения. Дольше всего выполняется код из №1, т.к. библиотеки подразумевают большое число проверок, что замедляет работу.

## Дополнительное задание №5

Исходный код:

<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/additional_task5/yaml_to_csv.py>

Результат:  
<https://github.com/gorgi07/ITMO-SPPO/blob/main/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9B%D0%B0%D0%B1%D0%B0%204/additional_task5/Output.csv>

CSV – текстовый формат, предназначенный для хранения табличных данных.

# Заключение

В ходе выполнения лабораторной работы я познакомился с формальными грамматиками и новыми форматами файлов для хранения данных. Написал парсер переводы YAML файла в XML файл. Осознал, что проще воспользоваться онлайн конвертером.

# Список литературы

1. TechTarget – XML vs. YAML: Compare configuration file formats: <https://www.techtarget.com/searchdatacenter/tip/XML-vs-YAML-Compare-configuration-file-formats>
2. Презентация: Информатика\_2024\_2025\_Лекция\_4.pdf