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

Факультет программной инженерии и компьютерной техники

Лабораторная работа по информатике №4

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант №18

Выполнил:  
Студент группы P3106  
Мельник Фёдор Александрович

Проверил:

Балакшин П.В.,

Кандидат технических наук, доцент ФПиКТ

Санкт-Петербург, 2024

# Оглавление

[Оглавление 2](#_Toc182007934)

[Задание 3](#_Toc182007935)

[Решение 5](#_Toc182007936)

[Задание 1 5](#_Toc182007937)

[Исходник JSON 5](#_Toc182007938)

[Задание 2 6](#_Toc182007939)

[Результат 6](#_Toc182007940)

[Задание 3 7](#_Toc182007941)

[Результат 7](#_Toc182007942)

[Сравнение результатов задания 2 и задания 3 7](#_Toc182007943)

[Задание 4 8](#_Toc182007944)

[Результат 8](#_Toc182007945)

[Сравнение результатов задания 2 и задания 4 8](#_Toc182007946)

[Задание 5 9](#_Toc182007947)

[Исходник JSON 9](#_Toc182007948)

[Код 9](#_Toc182007949)

[Результат 9](#_Toc182007950)

[Сравнение результатов задания 2 и задания 5 9](#_Toc182007951)

[Задание 6 10](#_Toc182007952)

[Результат 10](#_Toc182007953)

[Объяснение результата 10](#_Toc182007954)

[Задание 7 11](#_Toc182007955)

[Код 11](#_Toc182007956)

[Результат 11](#_Toc182007957)

[Описание формата 11](#_Toc182007958)

[Заключение 12](#_Toc182007959)

[Список источников 13](#_Toc182007960)

# Задание

1. 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.

2. 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.

Нельзя использовать готовые библиотеки, в том числе регулярные

выражения в Python и библиотеки для загрузки XML-файлов.

3. Дополнительное задание №1 (позволяет набрать +10 процентов от

максимального числа баллов БаРС за данную лабораторную).

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.

4. Дополнительное задание №2 (позволяет набрать +10 процентов от

максимального числа баллов БаРС за данную лабораторную).

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.

5. 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.

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.

6. Дополнительное задание №4 (позволяет набрать +5 процентов от

максимального числа баллов БаРС за данную лабораторную).

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.

7. Дополнительное задание №5 (позволяет набрать +5 процентов от

максимального числа баллов БаРС за данную лабораторную).

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.

# Решение

## Задание 1

### Исходник JSON

<https://github.com/ldpst/itmo/blob/main/sem-1_inf/labs/lab4/tasks/data/in.json>

## Задание 2

Репозиторий с кодом: <https://github.com/ldpst/itmo/tree/main/sem-1_inf/labs/lab4/tasks/main_task>

### Результат

https://github.com/ldpst/itmo/blob/main/sem-1\_inf/labs/lab4/tasks/data/out.xml

## Задание 3

Репозиторий с кодом: <https://github.com/ldpst/itmo/tree/main/sem-1_inf/labs/lab4/tasks/additional1>

### Результат

<https://github.com/ldpst/itmo/blob/main/sem-1_inf/labs/lab4/tasks/data/out-add1.xml>

### Сравнение результатов [задания 2](#_Результат) и [задания 3](#_Результат_1)

Результаты отличаются форматом вывода элементов массива. Если в случае задания 2 тэги элемента массива назывались <(list\_name)\_element>, то в задании 3 теги элемента массива называются <item>. Это происходит из-за того, что во втором задании мы сами выбрали название тэгов, а в третьем библиотека сама поставила тегу название по умолчанию.

## Задание 4

Репозиторий с кодом: <https://github.com/ldpst/itmo/tree/main/sem-1_inf/labs/lab4/tasks/additional2>

### Результат

<https://github.com/ldpst/itmo/blob/main/sem-1_inf/labs/lab4/tasks/data/out-add2.xml>

### Сравнение результатов [задания 2](#_Результат_2) и [задания 4](#_Результат_3)

Результаты полностью совпадают, так как изменен лишь принцип парсинга JSON файла, а конвертация не изменялась.

## Задание 5

### Исходник JSON

Json файл, имеющий больший объем данных

<https://github.com/ldpst/itmo/blob/main/sem-1_inf/labs/lab4/tasks/data/in-add3.json>

### Код

Репозиторий с кодом: <https://github.com/ldpst/itmo/tree/main/sem-1_inf/labs/lab4/tasks/additional3>

### Результат

<https://github.com/ldpst/itmo/blob/main/sem-1_inf/labs/lab4/tasks/data/out-add3.xml>

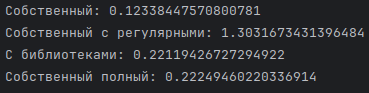
### Сравнение результатов [задания 2](#_Результат_4) и [задания 5](#_Результат_5)

Входной и выходной файл для задания 5 содержат в себе больше данных, чем файлы задания 2. Парсер задания 5 в отличии от парсера задания 2 может обратотать пары ключ-значение, знакение которых равно null или имеет тип bool

## Задание 6

Код: <https://github.com/ldpst/itmo/blob/main/sem-1_inf/labs/lab4/tasks/additional4/main.py>

### Результат



### Объяснение результата

Первый код выполняется быстрее всего, так как он обладает меньшими возможностями, чем коды 3 и 4 и запущен на тесте меньшего размера. Второй код обрабатывается дольше чем первый (хоть у них одинаковые возможности), так как каждое регулярное выражение медленно обрабатывает всю строку на поиск всех “мэтчей”. Третий код и запущен на меньшем объеме данных, поэтому он выполняется быстрее кода 4. Четвертый код выполняется дольше всех, так как обладает самыми большими возможностями (=3) и запущен на самом большом наборе данных.

## Задание 7

Выбранный формат: TSV

### Код

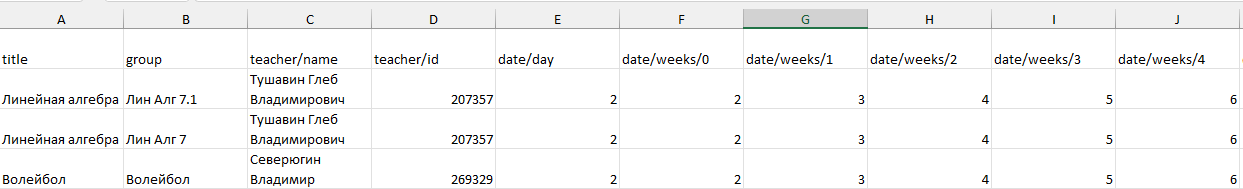
Репозиторий с кодом: <https://github.com/ldpst/itmo/tree/main/sem-1_inf/labs/lab4/tasks/additional5>

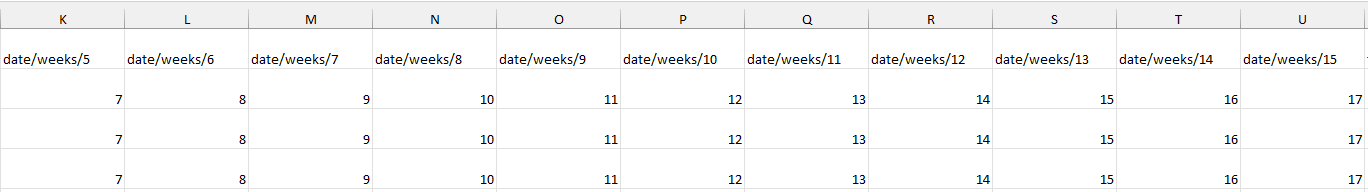
### Результат

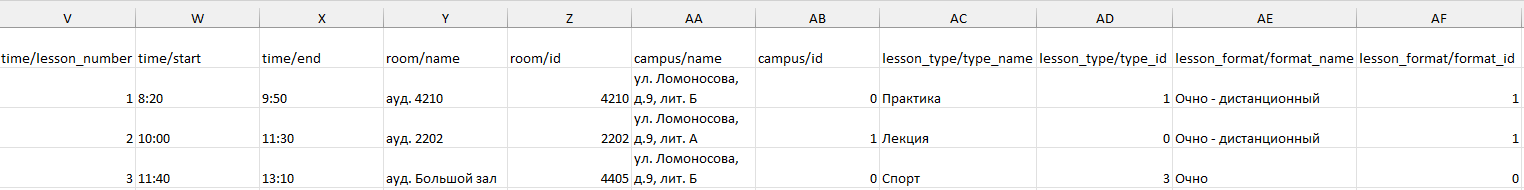
<https://github.com/ldpst/itmo/blob/main/sem-1_inf/labs/lab4/tasks/data/out-add5.tsv>

В виде таблицы:

Таблица разделена на несколько изображений в связи с размером: [рис. 1](#Закладка5), [рис. 2](#Закладка6) и [рис. 3](#Закладка7)

*Рисунок 1 - Таблица. Часть 1*

*Рисунок 2 - Таблица. Часть 2*

*Рисунок 3 - Таблица. Часть 3*

### Описание формата

TSV - табличный формат, разделителем которого служит символ табуляция

# Заключение

В процессе выполнения лабораторной работы я познакомился с такими форматами, как JSON, XML, YAML, CSV, TSV. Я написал собственный парсер из JSON файла и конвертер в XML и CSV. Я изучил библиотеки, с помощью которых можно с легкостью обрабатывать данные файлы и углубился в познании регулярных выражений, используя их на практике. Также я получил опыт работы с хронометражом и анализа полученных данных.

# Список источников

1. 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>.
2. Пишем изящный парсер на Питоне [Электронный ресурс] Хабр : [сайт]. — URL: <https://habr.com/ru/articles/309242> (дата обращения: 28.10.2024).