Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.01 - Информатика и вычислительная техника, Компьютерные системы и технологии*

*Дисциплина «Информатика»*

**Лабораторная работа №4**

**Исследование протоколов,**

**форматов обмена информацией и языков разметки**

**документов**

**Вариант №0**

Выполнила:

Денисова Алёна Александровна

Группа: Р3131

Преподаватель:

Авксентьева Елена Юрьевна

Санкт-Петербург 2023

Оглавление

[Задание 3](#_Toc150737927)

[Обязательное задание 4](#_Toc150737928)

[Дополнительное задание №1 9](#_Toc150737929)

[Дополнительное задание №2 10](#_Toc150737930)

[Дополнительное задание №3 13](#_Toc150737931)

[Дополнительное задание №4 17](#_Toc150737932)

[Дополнительное задание №5 18](#_Toc150737933)

[Ответы на вопросы 21](#_Toc150737934)

[Заключение 23](#_Toc150737935)

[Список использованных источников 23](#_Toc150737936)

# Задание

1. Обязательное задание (позволяет набрать до 45 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 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.
2. 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.

1. 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.

1. 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.

1. 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.

1. 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.

# Обязательное задание

Вариант: JSON -> XML

Исходный файл JSON:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1. Исходный файл JSON. Часть 1

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2. Исходный файл JSON. Часть 2

Полученный файл XML:

Изображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описание

Рисунок 3. Полученный файл XML

Код программы:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 4. Код программы обязательного задания. Часть 1

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 5. Код программы обязательного задания. Часть 2

# Дополнительное задание №1

Код программы:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 6. Код программы дополнительного задания №1

Полученный файл:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 7. Полученный файл в дополнительном задании №1. Часть 1

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 8. Полученный файл в дополнительном задании №1. Часть 2

В полученном файле вместо тега <root> используется тег <all>, потому что так предусмотрено реализацией библиотеки, однако на содержимое файла это не влияет.

Кроме того, элементы исходного файла, являющиеся массивами, оформляются иначе: вместо написания названия массива в тегах для каждого элемента (в моем случае это теги <schedule> - </schedule>) прописывается название массива (<schedule>), а для его элементов – теги <item>.

Также в тегах прописан атрибут type, указывающий на тип значения элемента.

# Дополнительное задание №2

Код программы:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 9. Код программы для дополнительного задания №2. Часть 1

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 10. Код программы для дополнительного задания №2. Часть 2

Полученный файл аналогичен файлу из обязательного задания.

# Дополнительное задание №3

Код программы:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 11. Код программы для дополнительного задания №3. Часть 1

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 12. Код программы для дополнительного задания №3. Часть 2

Исходный файл JSON (расписание на два дня):

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 13. Исходный файл JSON для дополнительного задания №3. Часть 1

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

Рисунок 14. Исходный файл JSON для дополнительного задания №3. Часть 2

Полученный файл XML:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 15. Полученный в дополнительном задании №3 файл XML. Часть 1

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Рисунок 16. Полученный в дополнительном задании №3 файл XML. Часть 2

Написанная для обязательного задания программа использовала формальные грамматики, т.е. код может осуществлять парсинг и конвертацию любых данных, представленных в исходном формате.

Добавление расписания второго дня в JSON-файл не повлияло на работоспособность программы, данные конвертировались корректно.

# Дополнительное задание №4

Код программы:

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 17. Код программы для дополнительного задания №4

Вывод программы:

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 18. Вывод программы в дополнительном задании №4

Самое долгое выполнение у программы из дополнительного задания №1, в котором использовалась готовая библиотека. Очевидно, именно реализация этой библиотеки повлияла на маленькую скорость выполнения.

Самое быстрое выполнение оказалось у программы из обязательного задания, т. е. у использующей простую замену метасимволов. Вероятно, это связано с тем, что данные обрабатываются линейно, без использования рекурсии или сложных блоков.

Программа, использующая регулярные выражения, оказалась немного медленнее, чем программа для обязательного задания. Это объясняется со скоростью обработки регулярных выражений и с тем, что регулярные выражения компилируются заново каждый раз.

# Дополнительное задание №5

Я выбрала табличный язык данных CSV.

Код программы:

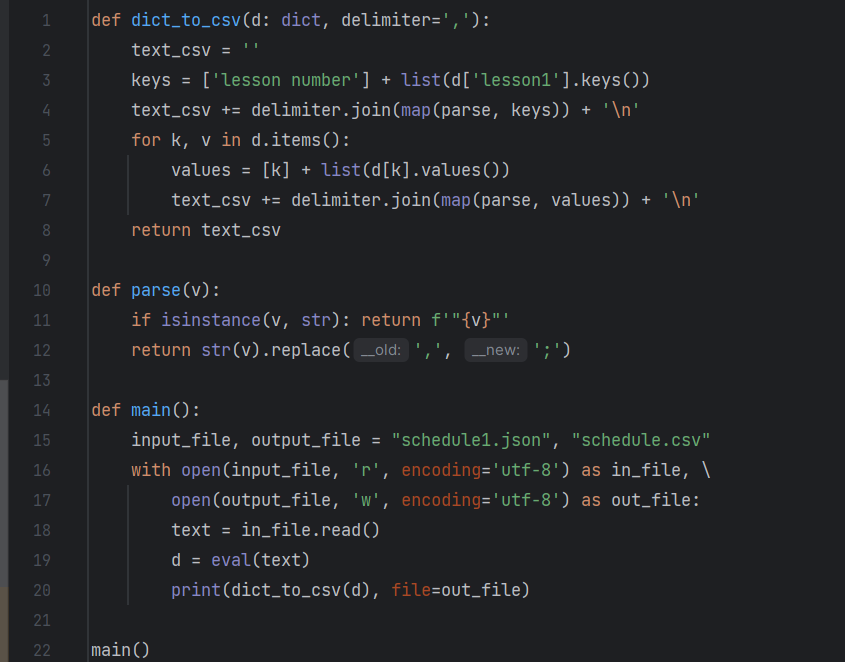


Рисунок 19. Код программы для дополнительного задания №5

Исходный файл JSON:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 20. Исходный файл JSON для дополнительного задания №5. Часть 1

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 21. Исходный файл JSON для дополнительного задания №5. Часть 2

Полученный файл CSV:

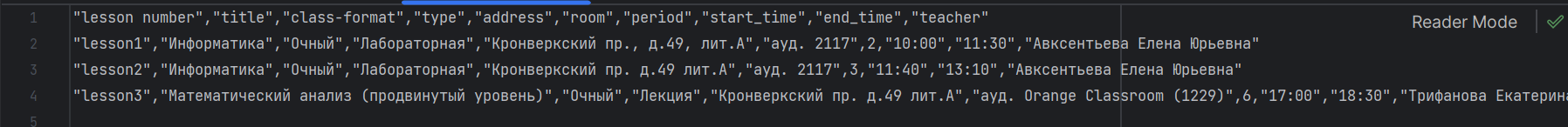


Рисунок 22. Полученный CSV файл

Для выполнения этого задания мне пришлось изменить исходный файл JSON, так как формат CSV не предусматривает содержания массива в качестве значения элемента, поэтому я вынесла поля из блоков «place» и «time» в основной блок.

В результате получила файл формата CSV, в котором каждая строка представляет отдельную запись, а значения разделяются запятыми. Это делает формат CSV удобным для обмена данными между различными программами и системами.

# Ответы на вопросы

1. В чём разница между Markup и Markdown?

Markup (разметка): это общий термин, который включает в себя различные языки разметки, такие как HTML, XML, LaTeX и другие. Markup является более общим понятием и включает в себя разные языки с разными синтаксисами.

Markdown: это простой, легковесный язык разметки, который создан для простоты чтения и написания. Он использует простой синтаксис и часто преобразуется в HTML или другие форматы разметки.

2. В чём заключается особенность PROTOBUF по сравнению с другими форматами?

Protocol Buffers (Protobuf) — это формат сериализации данных. Он хранит структурированные данные в двоичной форме, что позволяет быстрее передавать их по сети.

Поскольку структурированные данные хранятся в двоичном формате, они гораздо меньше текстовых форматов, таких как XML или JSON, что позволяет быстрее передавать их по сети. Кроме того, Protobuf разработан так, чтобы его было легко расширять, что делает его идеальным для работы с быстро меняющимися структурами данных и новыми возможностями.

3. Чем формат CSV отличается от формата TSV?

В CSV формате значения разделяются запятыми, а в TSV – символами табуляции.

4. Чем обусловлено постоянное появление новых форматов представления данных?

Стремлением создать универсальный формат представления данных, который будет подходить для любых задач, а также постоянно изменяющимися потребностями разработчиков.

5. Каким образом в формате XML представляются символы ‘>’ и ‘<’?

“<” (знак "меньше чем") представляется с помощью сущности “&lt;”.

“>” (знак "больше чем") представляется сущностью “&gt;”.

6. Что такое сериализация данных?

Сериализация — процесс перевода структур данных или состояния объекта в формат, который может быть сохранён или передан и реконструирован позже.

7. Каким образом в YAML обозначаются комментарии?

# comment text

8. Пояснить, как в языке разметки Markdown создать заголовки разных уровней, оформить код, вывести полужирный, курсивный и зачеркнутый текст?

Заголовки разных уровней оформляются с помощью решеток #, их количество определяет уровень заголовка. Код оформляется внутри обратных апострофов или внутри тройных обратных апострофов. Полужирный текст создается с помощью двойных звездочек \*\*. Курсивный – одинарные звездочки \*. Зачеркнутый – с помощью двух тильд ~~.

9. Какие форматы обмена данных используются в современных популярных мессенджерах (Viber, WhatsApp, Telegram и т.д.)?

WhatsApp, Telegram используют собственные протоколы (MTProto – Telegram), Viber – TSL, SRTP.

10. Как расшифровывается аббревиатура SVG?

Scalable Vector Graphics

11. Привести пример использования в языке HTML тега, который создаёт гиперссылку на url.

<a href="https://www.example.com">

12. Какое число структур может представлять собой в закодированном виде JSON-текст?

6 структур (объект, массив, числа, строки, булевы значения, значения null).

# Заключение

В ходе работы я познакомилась с языками разметки JSON, YAML, XML, CSV, написала несколько вариантов парсера-конвертера из JSON в XML и парсер-конвертер из JSON в CSV. Также я попрактиковалась в работе с регулярными выражениями и в подключении сторонних библиотек, научилась работать с библиотекой json2xml.

# Список использованных источников

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: Учебно-методическое пособие / Рецензент: Поляков В.И. - Санкт-Петербург: Университет ИТМО, 2019. - 56 с. - экз. - Режим доступа:

<https://books.ifmo.ru/book/2248/informatika:_laboratornye_raboty_i_testy:_uchebno-metodicheskoe_posobie_/_recenzent:_polyakov_v.i..htm>

1. Грошев А.С. Г89 Информатика: Учебник для вузов / А.С. Грошев. – Архангельск, Арханг. гос. техн. ун-т, 2010. -470с. -Режим доступа <https://narfu.ru/university/library/books/0690.pdf>