Лабораторная работа №4

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант 1

Выполнил:

Соколов Александр Алексеевич

Группа P3108

Проверила:

Бострикова Дарья Константиновна

**Содержание**

Задание 3

Основное задание 4

Дополнительное задание №1 4

Дополнительное задание №2 4

Дополнительное задание №3 4

Дополнительное задание №4 5

Дополнительное задание №5 5

Вывод 5

Источники 5

**Задание**

1. Определить номер варианта как остаток деления на 36 последних двух цифр своего идентификационного номера в ISU: например, 1255**98** / 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

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.
3. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.
4. Дополнительное задание No1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. a)  Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
   2. 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.

10.Дополнительное задание No2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

1. a)  Переписать исходный код, добавив в него использование регулярных выражений.
2. b)  Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

11.Дополнительное задание No3 (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 No1.

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.

12.Дополнительное задание No4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).

1. a)  Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле.
2. b)  Проанализировать полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

13.Дополнительное задание No5 (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.

**Основное задание**

Расписание, составленное для парсинга и конвертации по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/schedule.xml)

Листинг парсера из XML в JSON [ссылка](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/labinf4.py)

Результат выполнения программы дотсупен по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/output1.json)

**Дополнительное задание №1**

Листинг программы по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/dop4_1.py)

Результат выполнения программы доступен по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/dop_1.json)

Различий в выводе программ нет, единственное несущественное различие в форматах отступов перед тегами. В библиотеке они реализованы через табуляцию, а в моей версии конвертации я использую смешанный тип, и табуляцию, и обычный пробел. Код, при написании с помощью библиотек, по сравнению с первой версией программы, сократился.

**Дополнительное задание №2**

Листинг программы по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/dop4_2.py)

Результат выполнения программы доступен по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/dop_2.json)

Различий в выводе программ нет, так как регулярные выражения были использованы в блоке считывания информации о теге, т.е. сам его тег, значение, атрибут и т.д., а сама функция конвертации из XML осталась нетронутой.

**Дополнительное задание №3**

Листинг программы по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/dop4_3.py)

Результат выполнения программы доступен по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/dop_3.json)

Различий в выводе программ нет, т.к. третья дополнительная программа писалась с применением знаний о формальных грамматиках, для учета всех случаев при конвертации из XML в JSON.

**Дополнительное задание №4**

Листинг программы по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/dop4_4.py)

Результат выполнения программы:

labinf4 время: 0.0016341659999999994

dop4\_1 время: 0.06479325

dop4\_2 время: 0.0015727499999999978

dop4\_3 время: 0.002528833000000008

Дольше всего выполнялся код доп. задания 1, поскольку библиотека обрабатывает значения тегов с сохранением их типов, в то же время как я считываю XML файл построчно, следовательно все данные, которые я извлекаю при парсинге, являются строками. Также, код использовавший регулярные выражения работает быстрее, чем остальные, поскольку разделение на терминалы (значения тегов в XML) происходит быстрее.

**Дополнительное задание №5**

Листинг программы по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/dop4_5.py)

Результат выполнения программы доступен по [ссылке](https://github.com/cxxamg/ITMO/blob/main/labs/CS/Labinf4/dop_5.toml)

Табличные форматы CSV и TSV не совсем подходят для конвертации из XML, а остальные форматы довольно сложны, поэтому поискав похожий на JSON формат, мой выбор остановился на TOML.

В TOML вложенность показывается одним тегом над остальными значениями, также в нем не присутствуют отступы (из-за этого) и запятые, поэтому он довольно лаконичен.

TOML нужен для создания удобочитаемых и легко редактируемых конфигурационных файлов, особенно когда требуется простой синтаксис с возможностью организовать данные в секции и вложенные структуры.

TOML поддерживает базовые типы данных (строки, числа, даты, булевы значения), массивы и вложенные таблицы, что позволяет компактно хранить информацию и структурировать её логически. TOML успешно конкурирует с JSON и YAML, предлагая более строгий и минималистичный подход к созданию файлов конфигурации.

**Вывод**

В ходе выполнения лабораторной работы я познакомился с языками разметки XML и YAML, научился работать с ними и переводить данные между этими форматами, используя инструменты языка Python. Также узнал о формальных грамматиках и БНФ.

**Источники:**

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>
2. 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>