

**НИУ ИТМО**

**Факультет ПИИКТ**

Лабораторная работа №4

Исследование протоколов, форматов обмена информацией и языков  
разметки документов

Вариант 8

Выполнил Бойко Г. А., группа Р3116

г. Санкт-Петербург, 2024



Задание: .....	3
Дополнительное задание:.....	3
Заключение: .....	5
Список использованной литературы:.....	5

### Задание:

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.

Ссылка на код: <https://github.com/georgeboiko/informatics-lab4/blob/main/scripts/task1.py>

### Дополнительное задание:

- 1) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать. Сравнить полученные результаты и объяснить их сходство/различие.**

Для выполнения данного задания были использованы библиотеки json, yaml.

Ссылка на код: <https://github.com/georgeboiko/informatics-lab4/blob/main/scripts/addTask1.py>

Результаты выполнения программ на исходном файле, очевидно, идентичны. Однако программа, написанная с использованием библиотек позволяет конвертировать из json в yaml произвольные файлы, что не может исходная программа.

- 2) Переписать исходный код, добавив в него использование регулярных выражений. Сравнить полученные результаты и объяснить их сходство/различие.**

Для написания данной программы были использованы регулярные выражения.

Ссылка на код: <https://github.com/georgeboiko/informatics-lab4/blob/main/scripts/addTask2.py>

Результаты выполнения программ на исходном файле, очевидно, идентичны.

- 3) 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.**

Для написания данной программы были использованы формальные грамматики. Также применялся подход ООП. Корректность работы этой программы была проверена на наборе из 20 тестов.

Ссылка на код: <https://github.com/georgeboiko/informatics-lab4/blob/main/scripts/addTask3.py>

Результаты выполнения программ на исходном файле, очевидно, идентичны. Данная программа позволяет конвертировать из json в yaml произвольные файлы, что не может исходная программа.

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

Была написана программа, запускающая каждое решение в стократном числе на исходных данных.

```
Stupid convert only for in.json: 0.011094331741333008
Convert with yaml.dump: 0.1256093978881836
Convert with Regex: 0.059522390365600586
Convert with grammar: 0.11788535118103027
```

Можно заметить, что наиболее быстро работает программа основного задания, что очевидно. Программа с регулярными выражениями работает немного дольше. Теперь поговорим о программах, которые могут конвертировать любые файлы из json в yaml. Разумеется, их нельзя сравнивать с двумя предыдущими. Программа, написанная с использованием формальных грамматик работает немного быстрее, чем программа, использующая библиотеки. Возможно, это происходит из-за того, что программа с грамматиками написана и оптимизирована только для данной задачи, она не имеет другого, избыточного функционала, что нельзя сказать про программу, использующую библиотеки.

Ссылка на код: <https://github.com/georgeboiko/informatics-lab4/blob/main/scripts/addTask4.py>

- 5) 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. Проанализировать полученные результаты, объяснить особенности использования формата.**

На основе исходной, была написана программа, конвертирующая исходный файл в формат csv.

Ссылка на код: <https://github.com/georgeboiko/informatics-lab4/blob/main/scripts/addTask5.py>

На выходе получаем файл в формате csv. Ключи из json файла представлены в виде заголовков столбцов, а их значения - в соответствующих столбцах. Строки - сами объекты.

### Заключение:

В процессе выполнения лабораторной работы я работал с файлами в формате json и yaml. Узнал про готовые решения для конвертации из одного формата в другой, освежил знания регулярных выражений. Изучил формальные грамматики. Написал полноценный парсер из json в yaml.

### Список использованной литературы:

- 1) Лаздин А.В. ФОРМАЛЬНЫЕ ЯЗЫКИ, ГРАММАТИКИ, АВТОМАТЫ Санкт-Петербург: 2019
- 2) Lindsay Bassett Introduction to JavaScript Object Notation 2015