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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант №17

Выполнил: студент группы P3108 Васильев Никита

Проверил: Балакшин Павел Валерьевич, доцент факультета ПИиКТ, кандидат технических наук

Санкт-Петербург 2023

Содержание

[Задание 3](#_Toc151153579)

[Основные этапы вычисления 4](#_Toc151153580)

[Обязательное задание 4](#_Toc151153581)

[Дополнительное задание №1 6](#_Toc151153582)

[Сравнение полученных результатов 7](#_Toc151153583)

[Дополнительное задание №2 7](#_Toc151153584)

[Сравнение полученных результатов 9](#_Toc151153585)

[Дополнительное задание №3 9](#_Toc151153586)

[Сравнение полученных результатов 12](#_Toc151153587)

[Дополнительное задание №4 13](#_Toc151153588)

[Сравнение полученных результатов 15](#_Toc151153589)

[Дополнительное задание №5 15](#_Toc151153590)

[Заключение 15](#_Toc151153591)

[Список литературы 16](#_Toc151153592)

# Задание

Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла (в формате XML) в новый (в формате YAML) путём простой замены метасимволов исходного формата на метасимволы результирующего формата.

Дополнительное задание №1

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

Дополнительное задание №2

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

Дополнительное задание №3

1. 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.
2. Проверку осуществить как минимум для расписания с двумя учебными днями по два занятия в каждом.
3. Сравнить полученные результаты и объяснить их сходство/различие.

Дополнительное задание №4

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

Дополнительное задание №5

1. 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.
2. Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте.

# Основные этапы вычисления

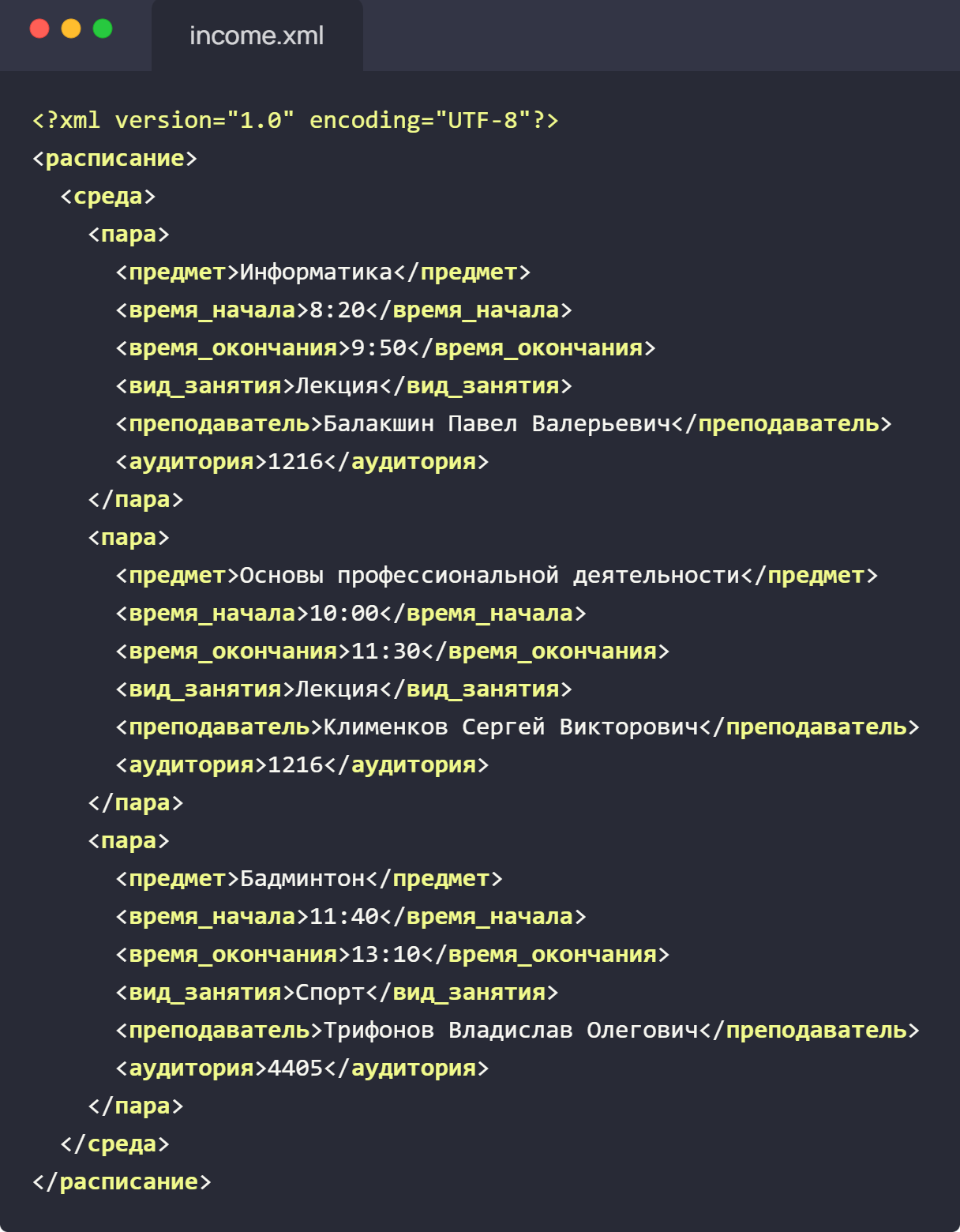


Рисунок 1 - Исходные данные в XML

## Обязательное задание

Для конвертации из XML в YAML воспользуюсь кодом на Python, представленным на Рисунке 2.



Рисунок 2 - Код на Python для основного задания

В результате получается YAML файл, представленный на Рисунке 3:

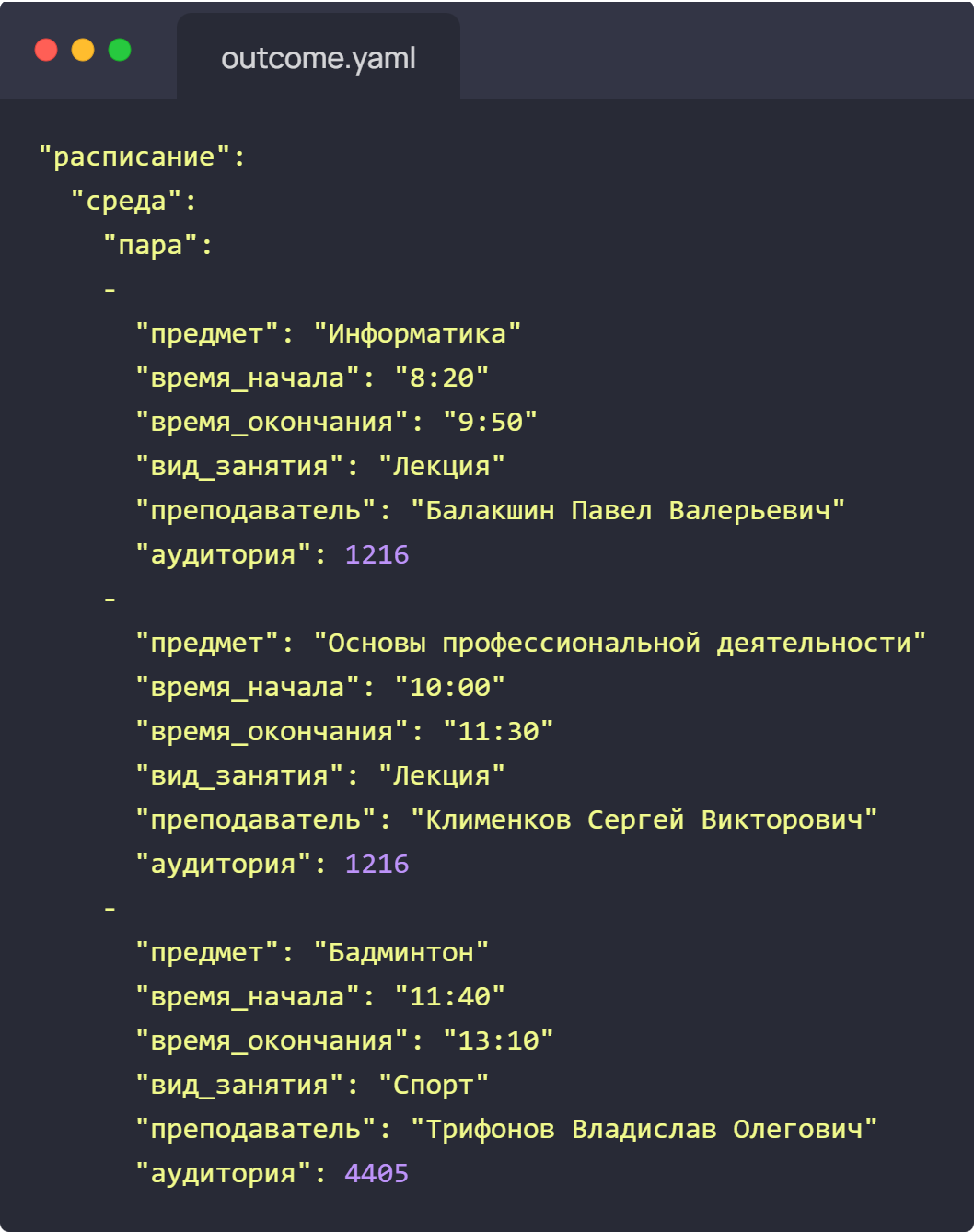


Рисунок 3 - Результат на YAML

## Дополнительное задание №1

Для перевода файла из XML в YAML применим библиотеку xmlplain. Теперь код на Python код будет выглядеть как на Рисунке 4:

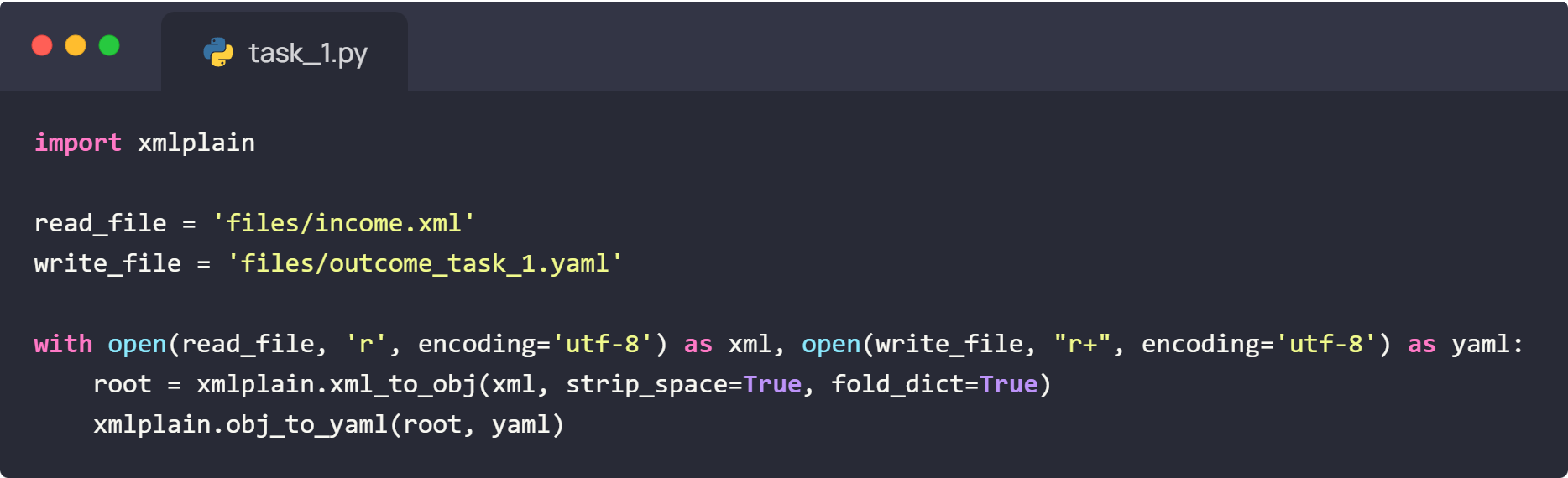


Рисунок 4 - Код на языке Python с использованием библиотеки xmlplain

### Сравнение полученных результатов

Полученный результат после работы кода из дополнительного задания №1 представлен на Рисунке 5. Он отличается от канонического оформления YAML документов (например, теги не взяты в кавычки), но при этом сохраняет свою функциональность.

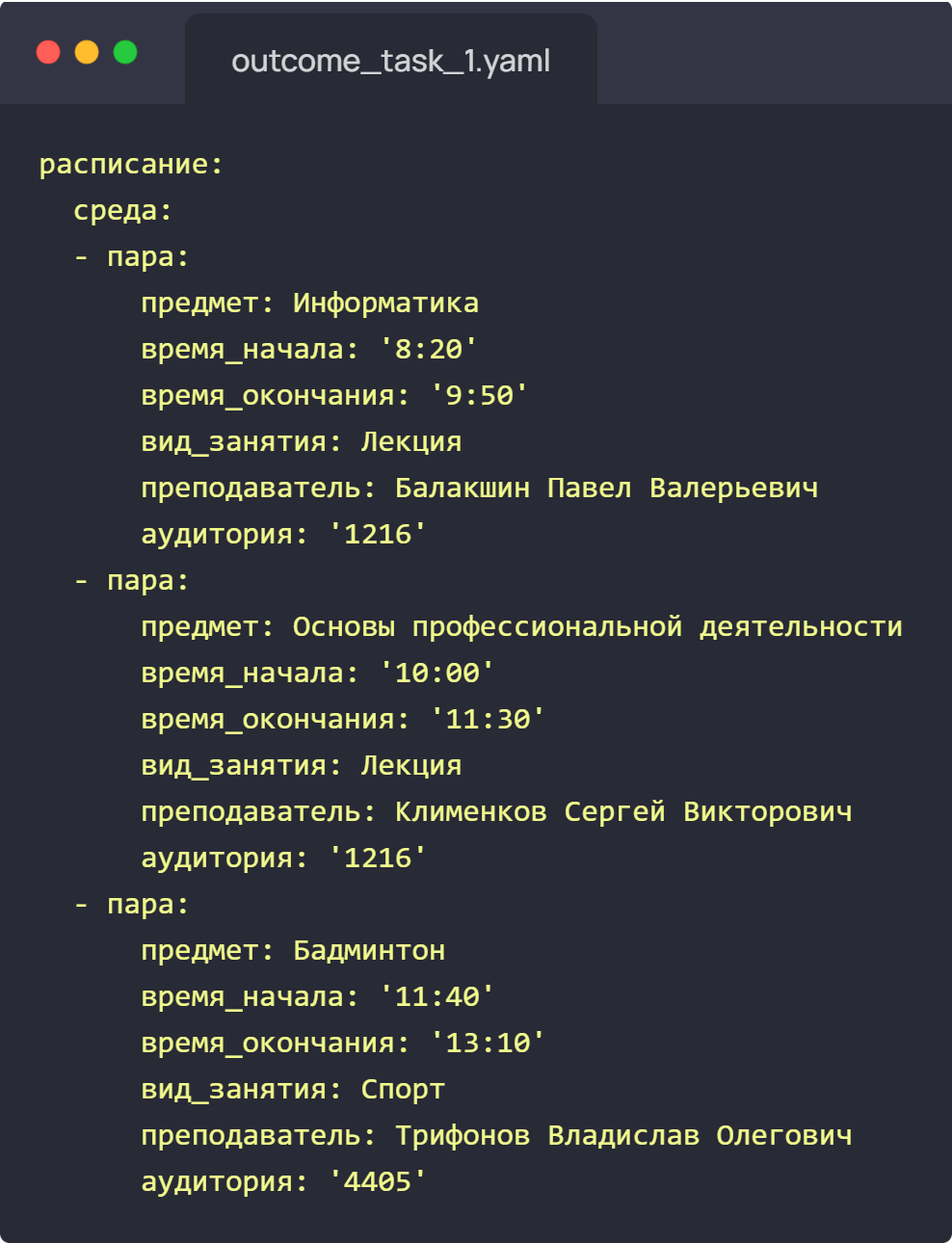


Рисунок 5 - Результат дополнительного задания №1 в YAML

## Дополнительное задание №2

Воспользуемся библиотекой re, чтобы использовать регулярные выражения и напишем код с их использованием, представленный на Рисунке 6:



Рисунок 6 - Код с использованием регулярных выражений

В результате получается YAML файл, представленный на Рисунке 7.

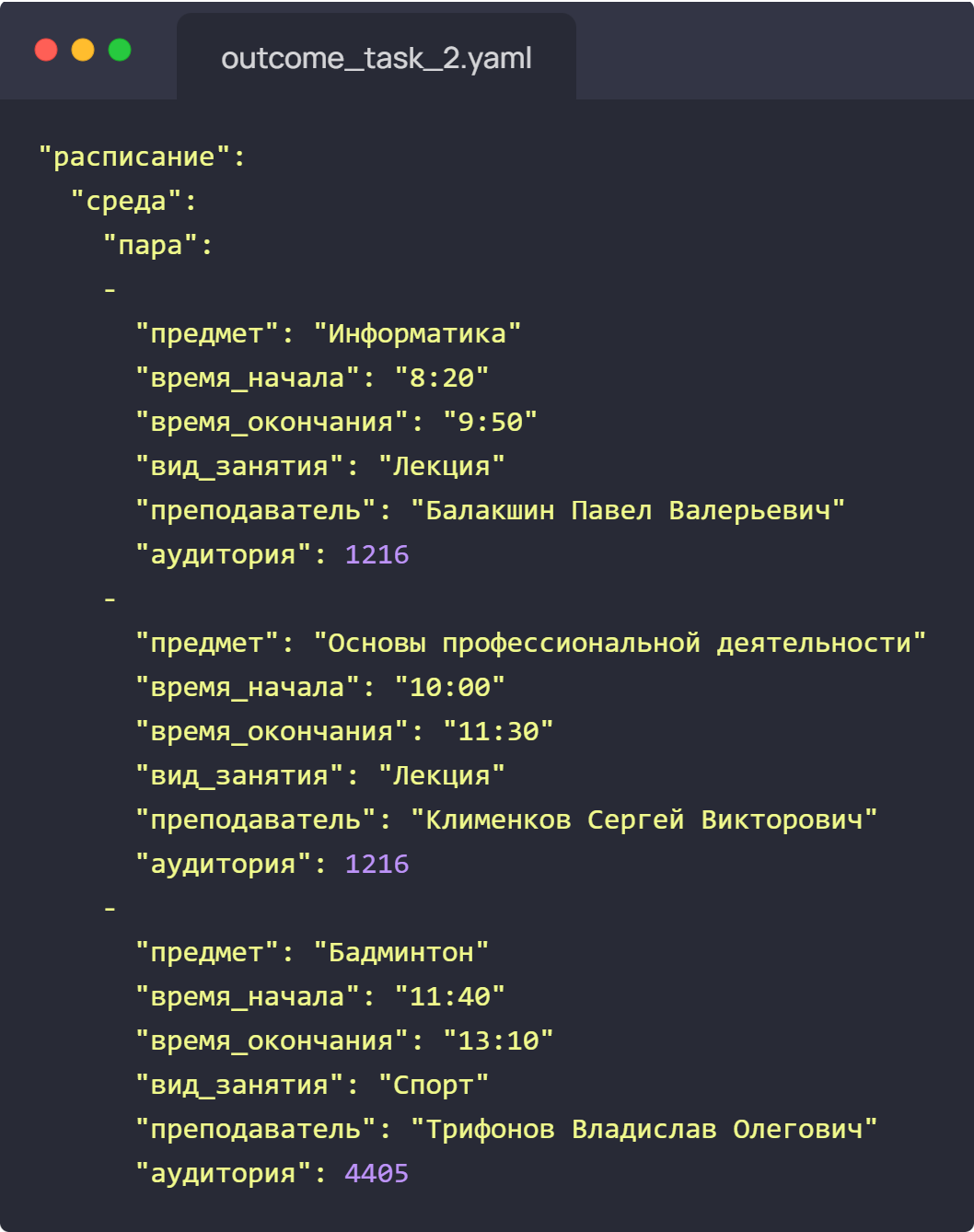


Рисунок 7 - Результат дополнительного задания №2 в YAML

### Сравнение полученных результатов

При использовании регулярных выражений результат аналогичен основному заданию, однако код, написанный с их применением, более универсален поскольку большинство регулярных выражений можно использовать при любых входных данных, написанных на языке XML.

## Дополнительное задание №3

Дополним исходные данные, добавив ещё один учебный день. Теперь XML файл выглядит так, как представлено на Рисунке 8.

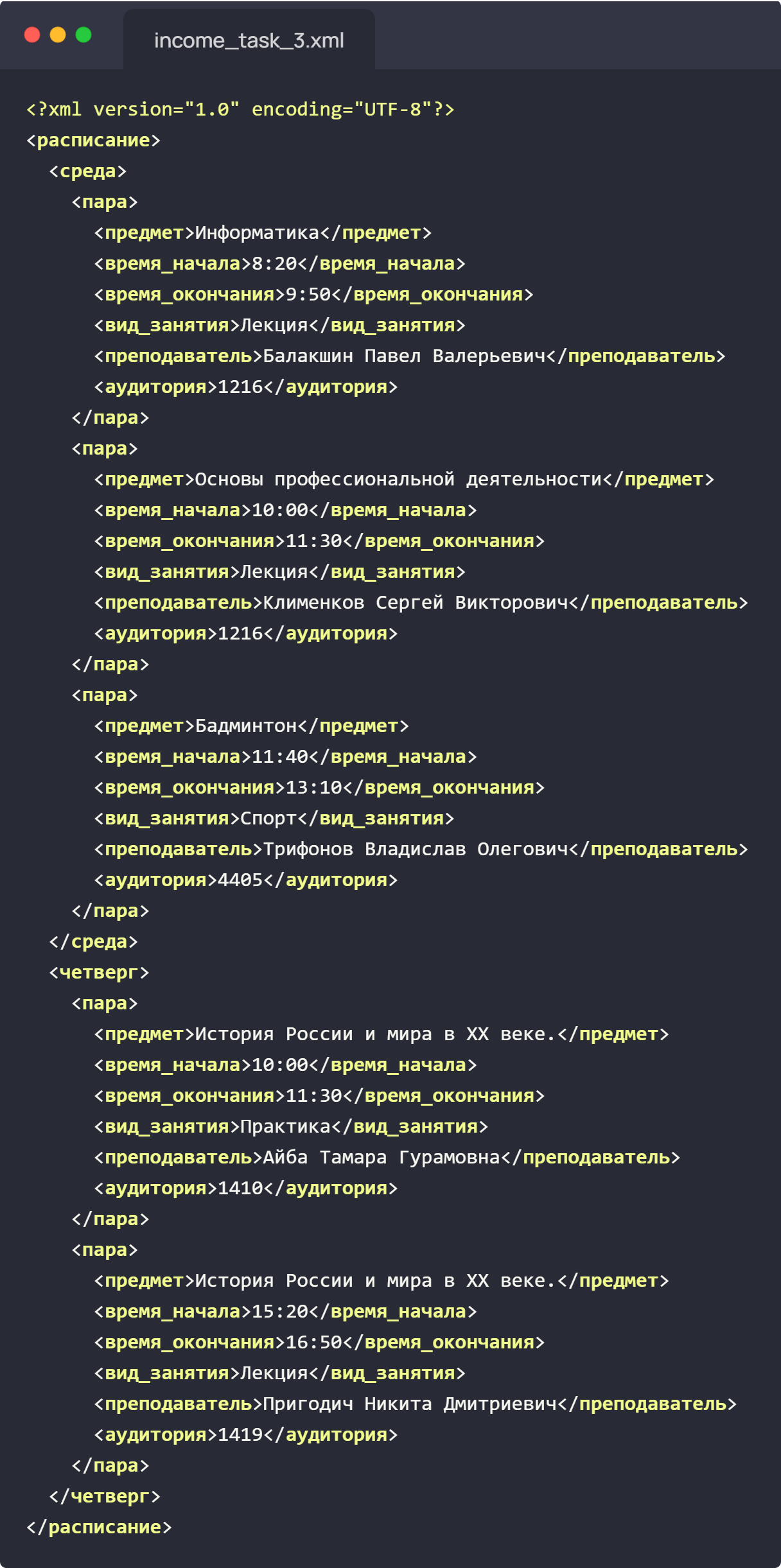


Рисунок 8 - Исходные данные в XML для дополнительного задания №3

В качестве формальной грамматики используем словарь, где α → β = α: β. Напишем программу на Python, представленную на Рисунке 9.



Рисунок 9 - Код для Дополнительного задания №3

В результате работы программы получим YAML файл, представленный на Рисунке 10.

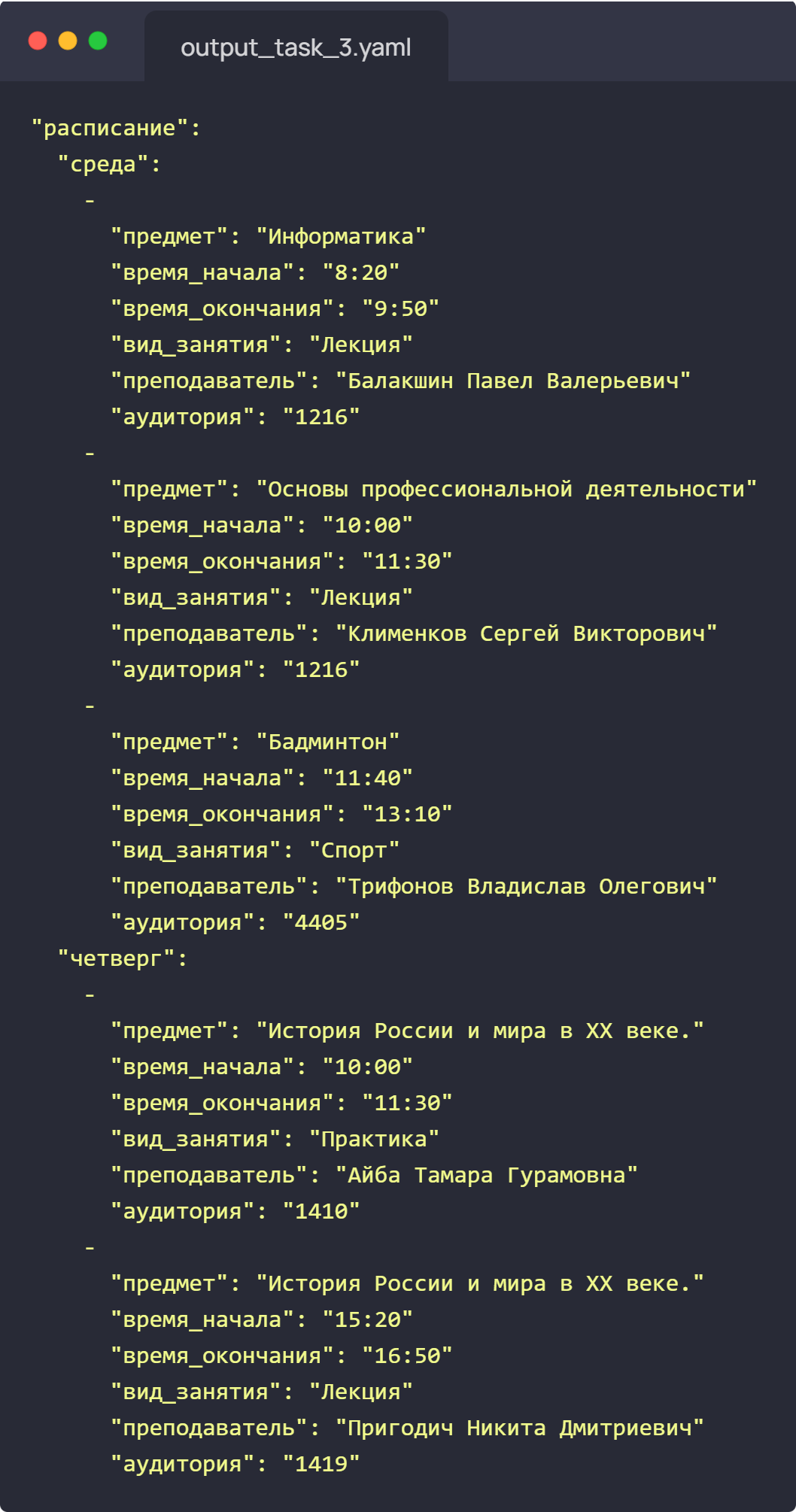


Рисунок 10 - Результат дополнительного задания №3 в YAML

### Сравнение полученных результатов

Используя формальную грамматику, мы получили такой же результат выходного файла, однако такой подход помогает обрабатывать файлы с большим объёмом данных.

## Дополнительное задание №4

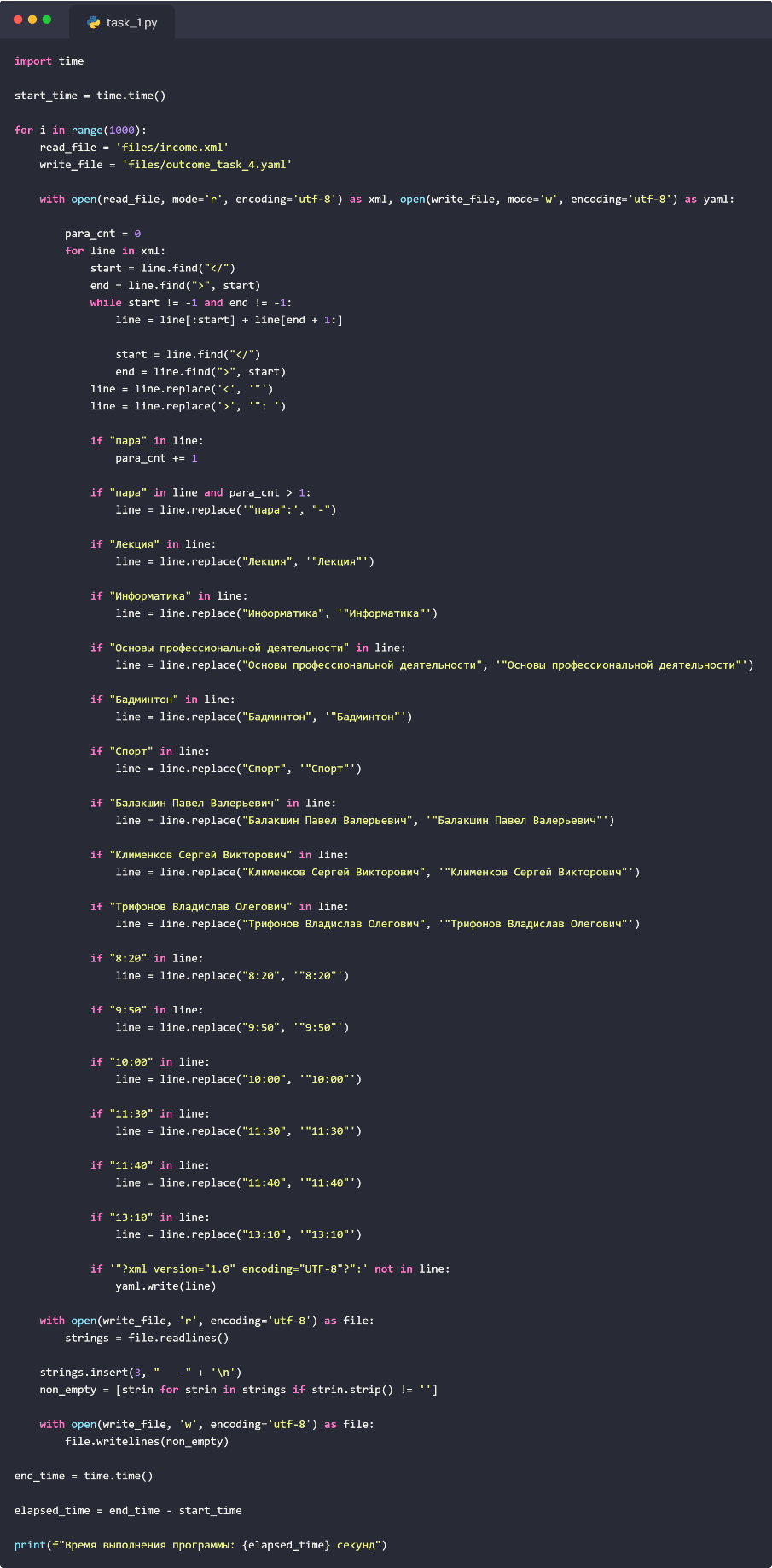
Для увеличения времени исполнения в 100 и в 1000 раз дополним исходный код программы так, как представлено на Рисунке 11.

Рисунок 11 - Код для Дополнительного задания №4

### Сравнение полученных результатов

Таблица 1 - Сравнение времени выполнения программы

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнение | **Однократное** | **Стократное** | **Тысячекратное** |
| **Время** (в секундах) | 0,013997316360473633 | 0,43395090103149414 | 2,53118896484375 |

## Дополнительное задание №5

Перепишем исходный код программы, чтобы он осуществлял парсинг и конвертацию исходного файла из XML в CSV. Необходимый код представлен на Рисунке 12.

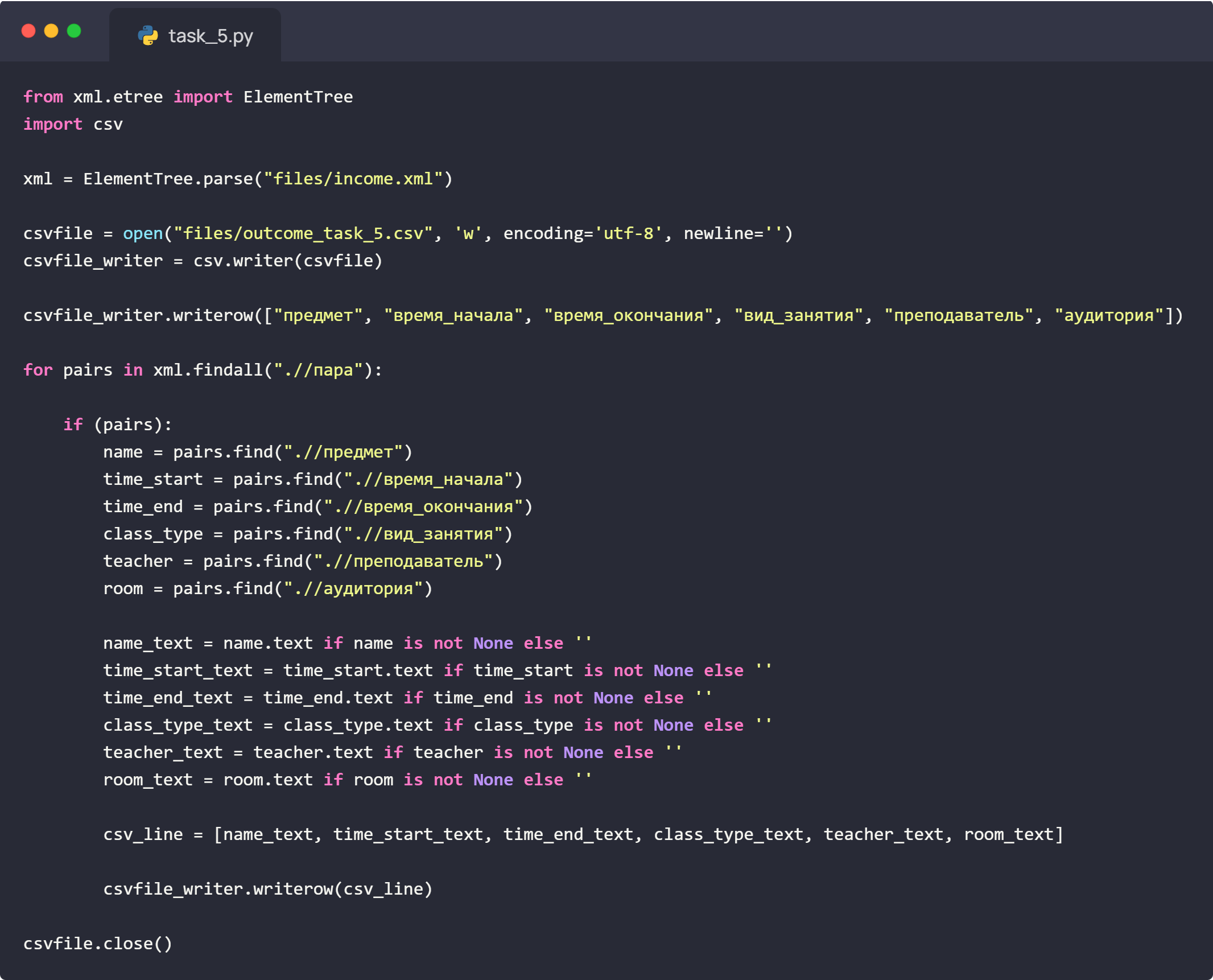


Рисунок 12 - Код для дополнительного задания №5

В результате работы программы получаем CSV файл, представленный на Рисунке 13.

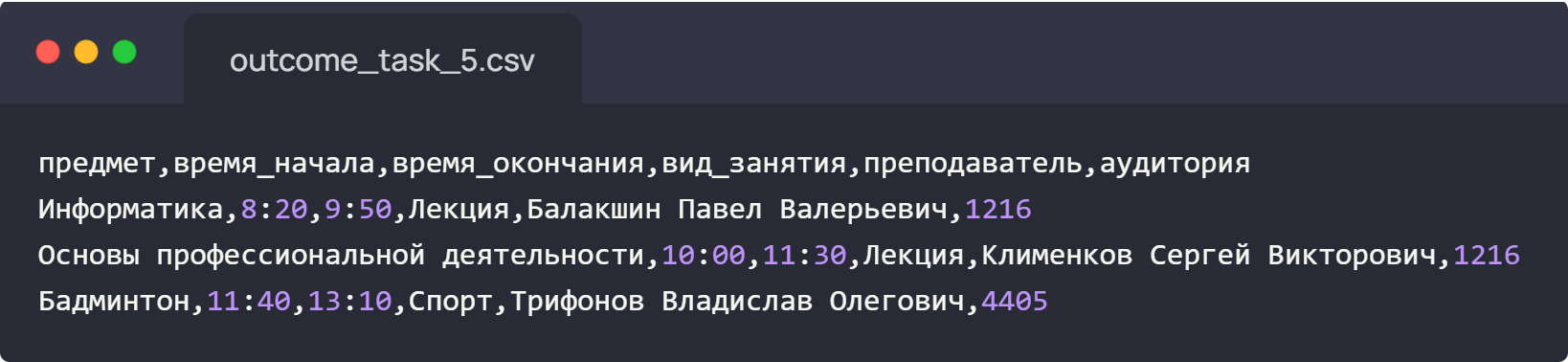


Рисунок 13 - Результат дополнительного задания №5 в CSV

### Анализ полученных результатов

В результате работы программы был получен файл в формате CSV с разделением по запятой, который можно использовать в офисных пакетах, чтобы составить электронную таблицу. Пример такой таблицы представлен на Рисунке 14.

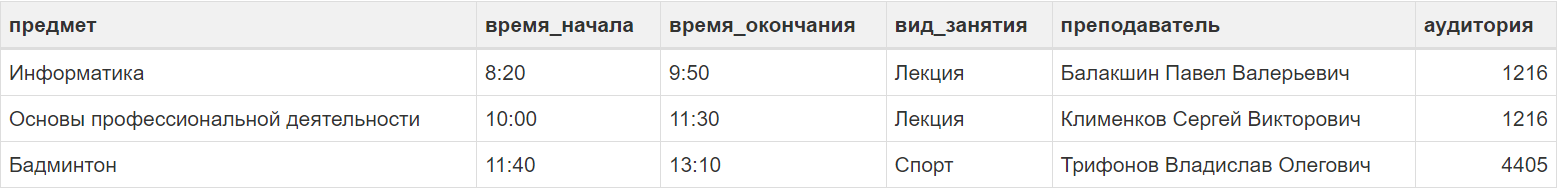


Рисунок 14 - CSV файл в виде таблицы

# Заключение

В ходе выполнения данной лабораторной работы я познакомился с различными языками разметки, научился осуществлять парсинг и конвертацию одного файла в другой, используя регулярные выражения, формальную грамматику и встроенные библиотеки.

# Список литературы

**Волкова И. А. Вылиток А. А., Руденко Т. В.** Формальные грамматики и языки. Элементы теории трансляции [Книга]. - Москва : Факультет вычислительной математики и кибернетики МГУ им. М. В. Ломоносов, 2009. - 3-е : стр. 115.

**Джеффри Фридл** Регулярные выражения [Книга] / ред. А. Галунов / перев. Матвеева Е. Киселева А.. - Санкт-Петербург - Москва : СИМВОЛ, 2008. - 3-е : стр. 598. - 9785932861219.

**Лопес Феликс Ромеро Виктор** Освоение регулярных выражений Python [Книга]. - Бирмингем : Packt Publishing, 2014. - стр. 110. - 9781783283156.