

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»
Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №4

по дисциплине

«Информатика»

Вариант №11

Выполнил студент группы Р3115

Федоров Егор Владимирович

Преподаватель:

Малышева Татьяна Алексеевна

Содержание

1	Текст задания	2
2	Основные этапы выполнения	3
3	Исходный код	4
4	Вывод	5
	Список литературы	6

1 Текст задания

1. Обязательное задание. Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый.
2. Дополнительное задание №1.
 - (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
3. Дополнительное задание №2.
 - (a) Переписать исходный код, добавив в него использование регулярных выражений.
 - (b) Сравнить полученные результаты и объяснить их сходство/различие.
4. Дополнительное задание №3.
 - (a) Используя свою исходную программу из обязательного задания, программу из дополнительного задания №1 и программу из дополнительного задания №2, сравнить стократное время выполнения парсинга + конвертации в цикле.
 - (b) Проанализировать полученные результаты и объяснить их сходство/различие.
5. Дополнительное задание №4.
 - (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

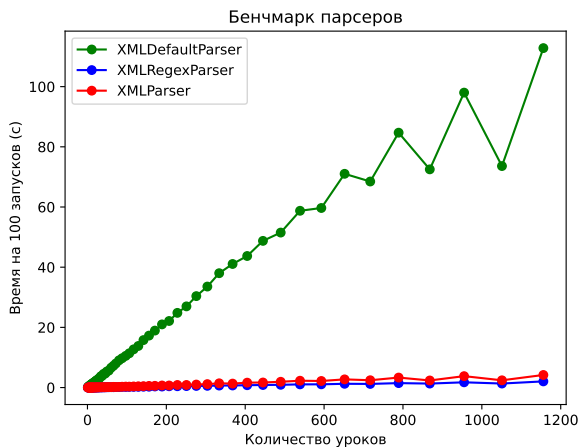
2 Основные этапы выполнения

Для выполнения первого задания были созданы абстрактные классы `Parser` и `Presenter`. Далее от них были унаследованы и реализованы классы `XMLDefaultParser` и `YAMLDefaultPresenter`. Парсинг реализован с помощью поиска необходимых тегов в тексте самым тривиальным способом.

Для выполнения первого дополнительного задания были созданы новые классы `XMLParser` и `YAMLPresenter` с помощью библиотек `xml.etree.ElementTree` и `pyyaml` соответственно. Различий в работе парсеров нет, различия в работе между `YAMLDefaultPresenter` и `YAMLPresenter` заключаются лишь в отступах.

Для выполнения второго лабораторного задания был создан класс `XMLRegexParser`. Алгоритм остался таким же, как и в первом задании, но поиск выполнен с помощью регулярных выражений.

Для выполнения третьего дополнительного задания были использованы библиотеки `matplotlib` и `multiprocessing` для начертания графика и более быстрого теста производительности парсеров. На рисунке 1a представлены все парсеры, на рисунке 1b представлены парсер на регулярных выражениях и с помощью библиотеки. На рисунке 2 представлено сравнение презентеров.



(a) Сравнение всех парсеров



(b) Сравнение самых быстрых парсеров

Рис. 1: Сравнение парсеров

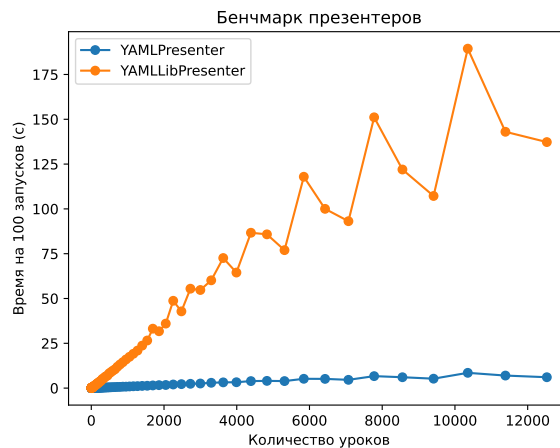


Рис. 2: Сравнение презентеров

Слишком медленная работа `XMLDefaultParser` объясняется неэффективным алгоритмом нахождения подстроки в строке за $O(n^2)$. Более быстрая работа `XMLRegexParser` по сравнению с `XMLParser` объясняется тем, что `XMLParser` полностью строит синтаксическое дерево и проверяет его корректность и лишь затем достает необходимые теги. `XMLRegexParser` просто ищет нужные теги среди других тегов.

Более медленная работа `YAMLLibPresenter` объясняется переводом сначала в объект типа `dict`, и лишь затем переводом в формат YAML. Для работы `YAMLPresenter` этого не нужно.

Для выполнения четвертого задания был выбран формат `protobuf` и были написаны классы `ProtobufPresenter` и `ProtobufParser`. Описание структур данных содержится в файле `schedule/schedule.proto`

3 Исходный код

Исходный код программы доступен в git-репозитории по ссылке <https://github.com/FEgor04/labs/tree/main/informatics/lab4>

4 Вывод

Во время выполнения данной лабораторной работы я научился работать с XML, YAML, PROTOBUF, изучил библиотеки Python для работы с ними.

Список литературы

- [1] Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем: Учебник для вузов. 2-е изд. – СПб.: Питер, 2011. – 688 с.
- [2] Алексеев Е. Г., Богатырев С. Д. Информатика. Мультимедийный электронный учебник. – Режим доступа: <http://inf.e-alekseev.ru/text/toc.html>