**Университет ИТМО**

**Факультет ПИиКТ**

Низкоуровневое программирование 3

Выполнил:

Болдов Олег Евгеньевич

Группа Р33301

Санкт-Петербург

2023

Цель: реализовать клиент-серверное приложение с разработанным протоколом передачи информации. Сервер должен содержать файл и принимать запросы от клиента, выполнять их и отправлять ответ назад.

Описание работы: было разработано 3 файла – server, client и transport. В transport содержатся структуры, описывающие объекты транспортного протокола.

Аспекты реализации:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Разберем эти две структуры. Question содержит имя файла, тип операции, над кем операция проводится, кол-во аргументов, аргументы, и значение результата.

Answer содержит bool успех результата, index выполненной операции, и сообщение ответа или ошибки.

ТИПЫ ОПЕРАЦИИ: 

Добавить тип, добавить объект (с помощью этой команды можно обновить существующий объект), удаление объекта, выборка по условиям, вся информация таблицы, конец работы.

Над кем совершается операция – это кого добавляем, кого удаляем, какой тип ищем.

Аргументы- это, например, атрибуты для создания нового типа или набор условий для выборки.

Результат – это значение, которое будет установлено в созданный объект.

Index – где разместился объект в файле, под каким номером создался тип в схеме.

Тесты:

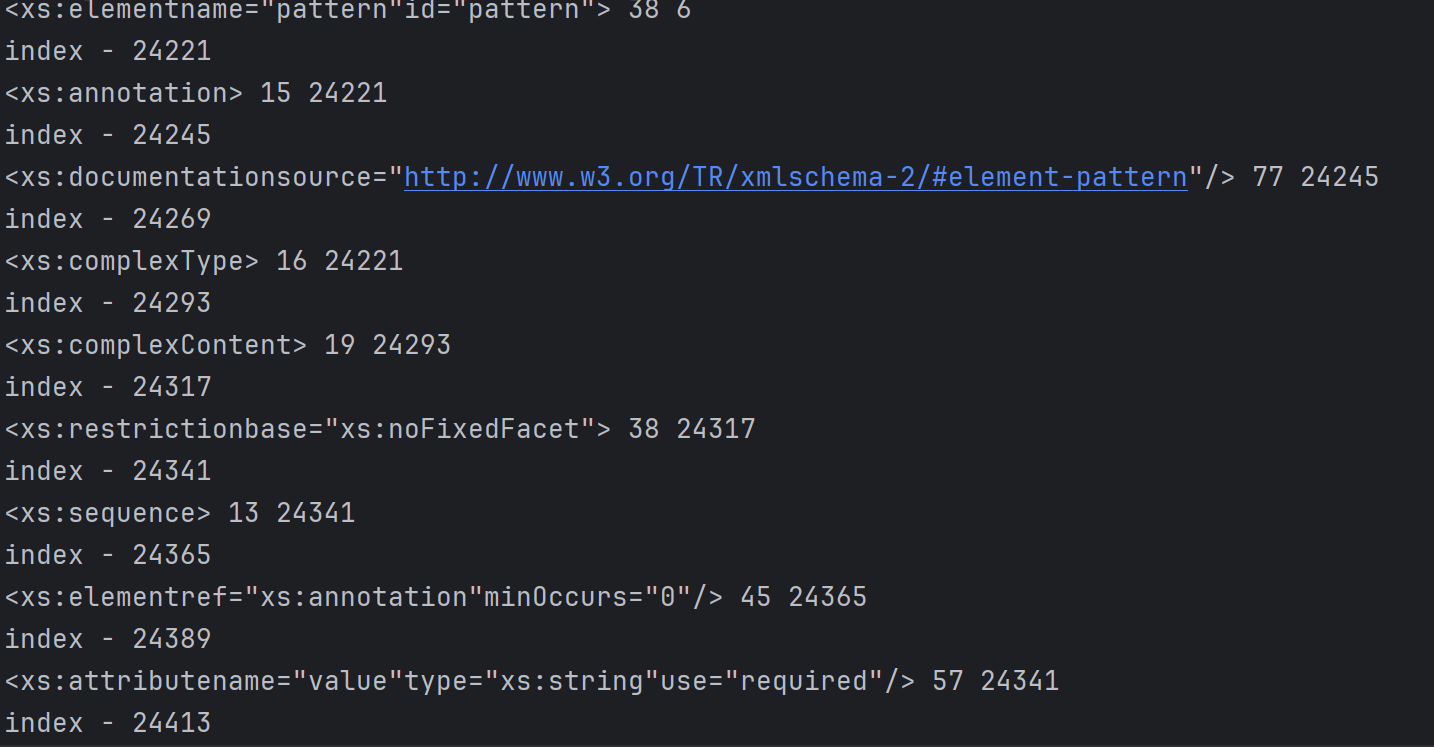
Изображение выглядит как текст, снимок экрана, Шрифт

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

Автоматически созданное описание

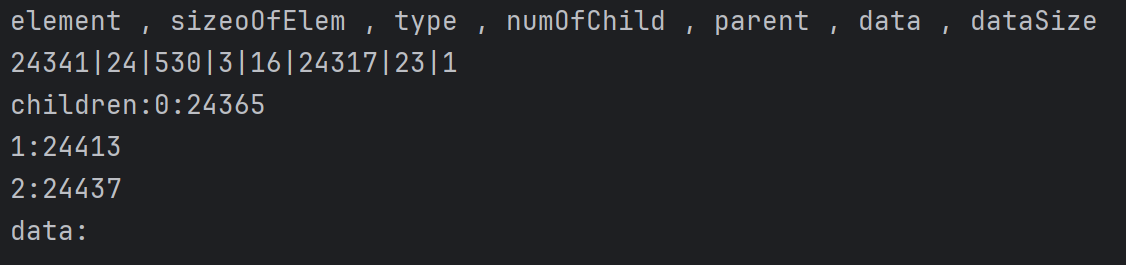
Еще одним тестом являлся распаршенный файл большого размера. Парсинг происходит в файле xml.c. Он отправляет запрос на создание в функцию client, который возвращает ему индекс созданного объекта. Этот индекс понадобится для того, чтобы при создании новых объектов мы всегда знали индекс родительской ноды. Запоминание индексов происходит через связный список link. Когда мы спускаемся глубже в дерево, мы создаем элемент связного списка, а когда мы поднимаемся наверх, мы удаляем последний элемент списка и перемещаетмся в предыдущего элемента, таким образом мы всегда знаем индекс нашего родителя.

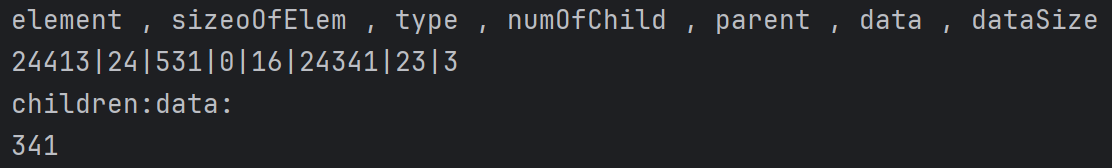
На картинке видна работа клиентской стороны. Он собирает объект, пишет индекс его родителя и свой индекс.



Название, длина названия в символах и индекс родителя.  
Для детальной информации нужно запустить принт всего файла см readme.

Рассмотрим пример:

Родитель, которого мы хотим найти  


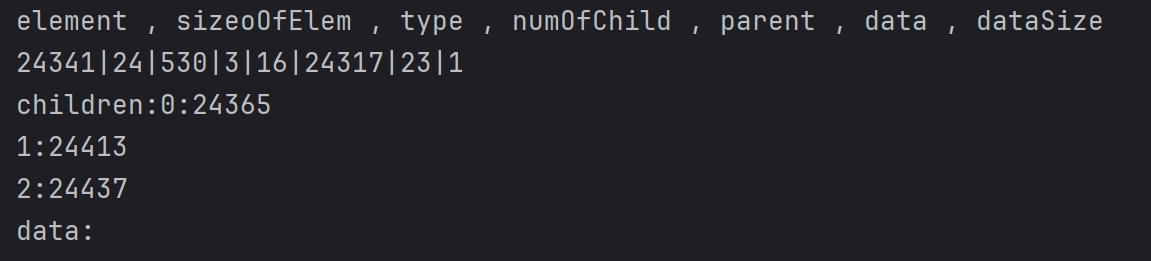
Ребенок, который имеет value=341 и являеся первым ребенком своего родителя(нумерация с нуля). 

Информацию о названии этой node можно найти согласно ее типу. У данной ноды тип 530.



Сделаем запрос  


Как это сделать см readme.

Полученный результат  


Это именно то, что нам было нужно

**Результаты:**

Мы вывели всю информацию о text.txt, добавили тип а, в котором содержится Float. Добавили объект а к ROOT, добавили объект float к a, сделали выборку, где у объекта a ребенок float имеет >2.0(в ответе вывелся наш объект, так как у него значение 2.35, после мы удаляем объект float, после чего делаем выборку снова, и уже ничего не выводится.

**Вывод:**

В ходе работы я разработал консольное сервер-клиент приложение, а также разработал спецификацию траспортного протокола обмена информации между ними.