**Представление данных в S-выражениях**

**Требования**

1. Определите формат представления данных с древовидной структурой в виде S выражений. В качестве семантической основы рекомендуется рассмотреть форматы XML и/или JSON. Допускается использование как классических S-выражений, так и расширенных, используемых в языке Clojure.

2. Разработайте язык навигации и простых запросов для заданного формата. Должны поддерживаться: относительный или абсолютный путь к узлу дерева, путь с условиями на свойства промежуточных узлов, путь с переменной вложенностью. Реализуйте функции поиска и модификации заданного формата посредством этого языка.

3. Разработайте представление схемы (по аналогии с XML Schema). Реализуйте функции проверки документа по схеме.

**Дополнительные требования**

• Разработайте язык трансформации в HTML, как упрощенный аналог XSLT. • Реализуйте модель разбора по аналогии с XML SAХ, обеспечьте поддержку трансформации и валидации по схеме в этом режиме.

Мы приняли решение использовать в качестве структуры данных формат edn и представлять его в виде многоуровневой структуры map, в которую также могут входить вектора. В таком виде Clojure их, судя по всему использует как S-выражения.

Первым делом был создан язык запросов:

\* - match all

[] - optional selector

   [0] - select by index, [="text"] - specific value equals "text", [%"text"] - one of the values equals "text"

catalog/\* - return all books

catalog/genre[="Fantasy"] - return books with genre Fantasy

catalog[%"Fantasy"] - same, but without telling that it is genre, or if something else also equals "Fantasy" it should be returned

catalog[2] - return the third object on catalog

orders/\*/name[%"Ellen Adams"] - return everything in "orders" that has a field name

~name или ~/name – relative search

Далее определена структура проекта. Не считая примеров, проект структурно состоит из 4 элементов: парсера для языка запросов, валидатора, который по схеме будет проверять документ, поиск по документу и модификация документов, два последних пункта с использованием языка запросов.

Парсер, валидатор и преобразование в HTML выполняет Тотмина Екатерина. Поиск, модификацию, частичную валидация, покрытие тестами и документацию – Масеевский Антон.

|  |  |
| --- | --- |
| Дата | Этап |
| 9.11.2023 | Составление этого документа |
| 7.12.2023 | Реализация парсера и поиска |
| 21.12.2023 | Реализация модификации и валидации |
| 28.12.2023 | Покрытие тестами и отладка основной функциональности |
| 21.01.2024 | Дополнительные требования (преобразование в HTML, частичная валидация) |