

Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №2**  
по «Низкоуровневое программирование»  
Вариант – 1(xpath)

Выполнил:

Студент группы Р3214

Абульфатов Р.М.

Преподаватели:

Кореньков Юрий Дмитриевич

Санкт-Петербург

2023

## Задание 2

Использовать средство синтаксического анализа по выбору, реализовать модуль для разбора некоторого достаточного подмножества языка запросов по выбору в соответствии с вариантом формы данных. Должна быть обеспечена возможность описания команд создания, выборки, модификации и удаления элементов данных.

Порядок выполнения:

1) Изучить выбранное средство синтаксического анализа

- a. Средство должно поддерживать программный интерфейс совместимый с языком C
- b. Средство должно параметризоваться спецификацией, описывающий синтаксическую структуру разбираемого языка
- c. Средство может функционировать посредством кодогенерации и/или подключения необходимых для его работы дополнительных библиотек
- d. Средство может быть реализовано с нуля, в этом случае оно должно быть основано на обобщённом алгоритме, управляемом спецификацией

2) Изучить синтаксис языка запросов и записать спецификацию для средства синтаксического анализа

- a. При необходимости добавления новых конструкций в язык, добавить нужные синтаксические конструкции в спецификацию (например, сравнения в GraphQL)
- b. Язык запросов должен поддерживать возможность описания следующих конструкций:

порождение нового элемента данных, выборка, обновление и удаление существующих элементов данных по условию

• Условия

- На равенство и неравенство для чисел, строк и булевских значений
- На строгие и нестрогие сравнения для чисел
- Существование подстроки

- Логическую комбинацию произвольного количества условий и булевских значений
- В качестве любого аргумента условий могут выступать литеральные значения (константы) или ссылки на значения, ассоциированные с элементами данных (поля, атрибуты, свойства)

• Разрешение отношений между элементами модели данных любых условий над сопрягаемыми элементами данных

• Поддержка арифметических операций и конкатенации строк не обязательна

c. Разрешается разработать свой язык запросов с нуля, в этом случае необходимо показать

отличие основных конструкций от остальных вариантов (за исключением типичных выражений

типа инфиксных операторов сравнения)

3) Реализовать модуль, использующий средство синтаксического анализа для разбора языка запросов

a. Программный интерфейс модуля должен принимать строку с текстом запроса и возвращать

структуру, описывающую дерево разбора запроса или сообщение о синтаксической ошибке

b. Результат работы модуля должен содержать иерархическое представление условий и других выражений, логически представляющие собой иерархически организованные

данные, даже если на уровне средства синтаксического анализа для их разбора было использовано линейное представление

4 Реализовать тестовую программу для демонстрации работоспособности созданного модуля,

принимаящую на стандартный ввод текст запроса и выводящую на стандартный вывод результирующее дерево разбора или сообщение об ошибке

5 Результаты тестирования представить в виде отчёта, в который включить:

а. В части 3 привести описание структур данных, представляющих результат разбора запроса

б. В части 4 описать, какая дополнительная обработка потребовалась для результата разбора, представляемого средством синтаксического анализа, чтобы сформировать результат работы созданного модуля

с. В части 5 привести примеры запросов для всех возможностей из п.2.б и результирующий вывод тестовой программы, оценить использование разработанным модулем оперативной памяти

### **Исходный код:**

[https://github.com/ThisAster/llp\\_lab2/tree/master](https://github.com/ThisAster/llp_lab2/tree/master)

### **Описание работы мною созданной**

#### **системы:**

Происходит считывание пользовательского ввода из стандартного потока ввода (stdin). Полученная строка затем направляется в модуль парсера, ответственный за создание структуры request. После этого полученный request передается в модуль path\_view, который осуществляет отображение распарсенного запроса на экран.

### **На основе просмотра языка XPath были созданы следующие типы запросов.**

add/1/[name=qwe][age=10][height=50.22][healthy=1]] – некорректный запрос.

add/1/[string=qwe][int=10][double=50.22][bool=1] – добавление записи по id 1.

remove/1 – удаление записи по id1

find/1 – поиск записи с id1

update/1/[int=11] – обновление вершины с id1

find/\*/[int=10] | [bool=1]&[double>=5.0] – поиск записей с определенными характеристиками

remove/\* - удаление всех записей

find/\* - поиск всех записей

update/1/[int=11] – обновление записи с id1

## Описание структур данных:

```
struct attribute {
    char *left;
    char *right;
    char *condition;
    struct attribute *next_attribute;
    char *combined_condition;
};

struct request {
    char *operation;
    char *parent_id;
    char *star;
    struct attribute *attributes;
};
```

## Результаты запросов:

```
thisaster@DESKTOP-453S0M5:/mnt/c/Users/freiz/CLionProjects/untitled$ ./main
enter your request
add/1/[string=qwe][int=10][double=50.22][bool=1]

action - add

id - 1
  left part - string
    condition - =
      right part - qwe
        left part - int
          condition - =
            right part - 10
              left part - double
                condition - =
                  right part - 50.22
                    left part - bool
                      condition - =
                        right part - 1
```

```
thisaster@DESKTOP-453S0M5:/mnt/c/Users/freiz/CLionProjects/untitled$ ./main
enter your request
remove/1

action - remove

id - 1
```

```

thisaster@DESKTOP-453S0M5:/mnt/c/Users/freiz/CLionProjects/untitled$ ./main
enter your request
find/1

action - find

id - 1

thisaster@DESKTOP-453S0M5:/mnt/c/Users/freiz/CLionProjects/untitled$ ./main
enter your request
update/1/[int=11]

action - update

id - 1
    left part - int
        condition - =
            right part - 11
                left part - (null)
                    condition - (null)
                        right part - (null)

```

```

thisaster@DESKTOP-453S0M5:/mnt/c/Users/freiz/CLionProjects/untitled$ ./main
enter your request
find/*/[[int=10]][double=22.2]

action - find

id - *
    left part - int
        condition - =
            right part - 10
                between pair - |
                    left part - double
                        condition - =
                            right part - 22.2
thisaster@DESKTOP-453S0M5:/mnt/c/Users/freiz/CLionProjects/untitled$ ./main
enter your request
add/1/[name=qwe][age=10][height=50.22][healthy=1]]
incorrect

```

## Конструкции, которые реализованы в лабораторной работе:

Операции:

- 1) + - добавление
- 2) - - удаление
- 3) ? - поиск
- 4) = - обновление

Булевы операторы для атрибутов:

- 1) = - равно
- 2) > - больше
- 3) < - меньше
- 5) >= - больше или равно
- 6) <= - меньше или равно
- 7) ! - инвертировать
- 8) & - логическое И
- 9) | - логическое ИЛИ

Элемент:

- 1) \* - все элементы
- 2) number - id элемента
- 3) / - дети элемента (запрос на поиск по id родителя)

## Вывод:

В рамках текущей лабораторной работы был разработан парсер для обработки запросов к ранее созданной базе данных. В целях упрощения были внесены небольшие изменения в язык XPath, такие как использование символов ->, +-= и других. Также была внедрена структура request для хранения распарсенных запросов. В ходе выполнения работы я реализовал модуль, который парсит строку в запрос, сделал отображение запроса, а также добавлена тестовая часть, которая принимает входные данные в виде строки, передавая их на вход парсеру. Затем тестовый блок вызывает соответствующий модуль для визуализации запроса.