

**Министерство науки и высшего образования Российской Федерации**  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**Национальный исследовательский университет ИТМО**

**Факультет Программной Инженерии И Компьютерной Техники**

**По дисциплине:**

**Низкоуровневое программирование**

Лабораторная работа №2

**Студент:**

Ханалайнен Данил Михайлович, Р33302

**Преподаватель:**

Кореньков Юрий Дмитриевич

**Вариант:**

8 (LINQ)

г. Санкт-Петербург

2023 г.

### **Задание:**

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.

### **Аспекты реализации:**

Для построения синтаксического дерева запроса был использован bison. Правила грамматики находятся в файле parser.y и близки к настоящему LINQ.

Реализованный программный модуль принимает на вход строку и преобразует ее к древовидной структуре, объявленной в tree.h. Дерево строится из нод, которые определены так:

```
struct node {
    enum node_type type;
    union content main;
    union content additional;

    union content main_second;
    union content additional_second;

    node *left;
    node *right;
    node *center;
};
```

Соответственно, результат запроса представлен корнем данного дерева. Также для демонстрации внутренних компонентов дерева была реализована функция print\_tree. Для результата разбора, представляемого средством синтаксического анализа, потребовалось также реализовать функции, создающие ноды, чтобы сформировать результат работы созданного модуля.

Примеры запросов и результирующий вывод тестовой программы:

Запрос	Результат
from student in studs where student.number contains Joe delete student;	<pre> DELETE_QUERY ├ table name: studs ├ entity name: student └ └ FILTER_CMP ├ cmp type: SUBSTRING └ └ OLD_COLUMN ├ entity name: student ├ column name: name └ └ VARCHAR_NODE └ value: Joe </pre>
from player in players where player.power <= 90 select player;	<pre> SELECT_QUERY ├ table name: players ├ entity name: player └ └ CREATE_LIST └ └ FILTER_CMP ├ cmp type: LESS_OR_EQUALS └ └ OLD_COLUMN ├ entity name: player ├ column name: power └ └ INT_NODE └ value: 90 </pre>
from human in people where human.is_at_work == false update human set is_at_work equals true;	<pre> UPDATE_QUERY ├ table name: people ├ entity name: human └ └ CREATE_LIST └ └ FILTER_CMP ├ cmp type: EQUALS └ └ OLD_COLUMN ├ entity name: human ├ column name: is_at_work └ └ BOOLEAN_NODE └ </pre>

	<pre>       L value: false CREATE_LIST       L       L BOOLEAN_NODE       L value: true </pre>
in people insert joe, 100;	<pre> INSERT_QUERY { table name: people   L CREATE_LIST     L CREATE_LIST       L VARCHAR_NODE         L value: joe       L INT_NODE         L value: 100 </pre>

Оценка использования программой оперативной памяти: память выделяется только для структур node, а также строк внутри union внутри node. Также фиксированное значение памяти выделяется для формирования результата разбора синтаксического дерева (строка фиксированной длины).

### **Вывод:**

В ходе выполнения лабораторной работы я реализовал программный модуль, задействующий средство синтаксического анализа bison и формирующий древовидную структуру запроса для дальнейшей разработки базы данных.