

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**“Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики ”**

**(НИУ ИТМО)**

---

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.04.04 Программная инженерия

**Лабораторная работа № 1**

**Вариант 3**

**По дисциплине “Системное программное обеспечение”**

Студент группы Р4114

Трофимова Полина  
Владимировна

Преподаватель

Кореньков Юрий Дмитриевич

Санкт-Петербург, 2023 г.

## Задание на лабораторную

Использовать средство синтаксического анализа по выбору, реализовать модуль для разбора текста в соответствии с языком по варианту. Реализовать построение по исходному файлу с текстом синтаксического дерева с узлами, соответствующими элементам синтаксической модели языка. Вывести полученное дерево в файл в формате, поддерживающем просмотр графического представления.

Порядок выполнения:

1. Изучить выбранное средство синтаксического анализа
  - a. Средство должно поддерживать программный интерфейс, совместимый с языком Си.
  - b. Средство должно параметризоваться спецификацией, описывающей синтаксическую структуру разбираемого языка.
  - c. Средство может функционировать посредством кодогенерации и/или подключения необходимых для его работы дополнительных библиотек.
2. Изучить синтаксис разбираемого по варианту языка и записать спецификацию для средства синтаксического анализа, включающую следующие конструкции:
  - a. Подпрограммы со списком аргументов и возвращаемым значением
  - b. Операции контроля потока управления – простые ветвления if-else и циклы или аналоги
  - c. В зависимости от варианта – определения переменных
  - d. Целочисленные, строковые и односимвольные литералы
  - e. Выражения численной, битовой и логической арифметики
  - f. Выражения над одномерными массивами
  - g. Выражения вызова функции
3. Реализовать модуль, использующий средство синтаксического анализа для разбора языка по варианту
  - a. Программный интерфейс модуля должен принимать строку с текстом и возвращать структуру, описывающую соответствующее дерево разбора и коллекцию сообщений ошибке.
  - b. Результат работы модуля – дерево разбора – должно содержать иерархическое представление для всех синтаксических конструкций, включая выражения, 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.

a. Через аргументы командной строки программа должна принимать имя входного файла для чтения и анализа, имя выходного файла записи для дерева, описывающего синтаксическую структуру разобранного текста.

5. Результаты тестирования представить в виде отчета, в который включить:

a. В части 3 привести описание структур данных, представляющих результат разбора текста (3a)

b. В части 4 описать, какая дополнительная обработка потребовалась для результата разбора,

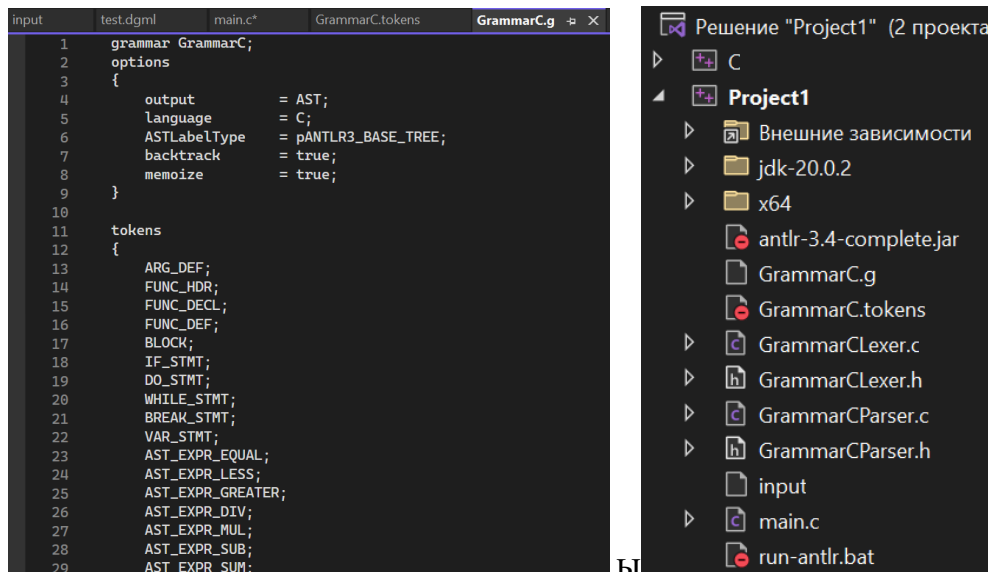
предоставляемого средством синтаксического анализа, чтобы сформировать результат работы

созданного модуля

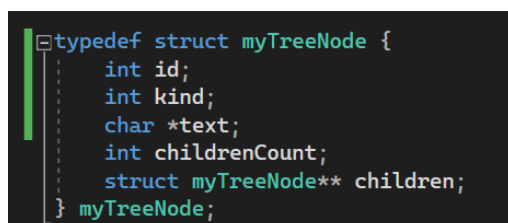
c. В части 5 привести примеры исходных анализируемых текстов для всех синтаксических конструкций разбираемого языка и соответствующие результаты разбора.

Для выполнения задания используется ANTLR3. Он выбран как одно из наиболее известных и широко используемых средств автоматизации построения синтаксических анализаторов для различных языков программирования. ANTLR является синтаксическим LL-анализатором, входная строка анализируется слева направо и строится левосторонний вывод.

На основе заданного варианта синтаксической модели языка была сформированная грамматика, сгенерированы парсер и лексер.

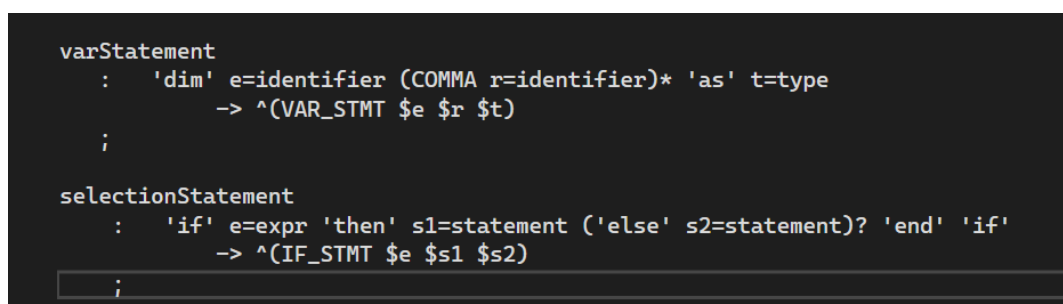


Каждый узел дерева описывается, который получается после парсинга дерева по заданной грамматике, описывается структурой:



Дерево содержит id узлов, kind – типы узлов, text – текстовое значение узла, количество дочерних узлов и массив указателей на дочерние узлы.

Грамматика описывается в отдельном файле GrammarC.g.



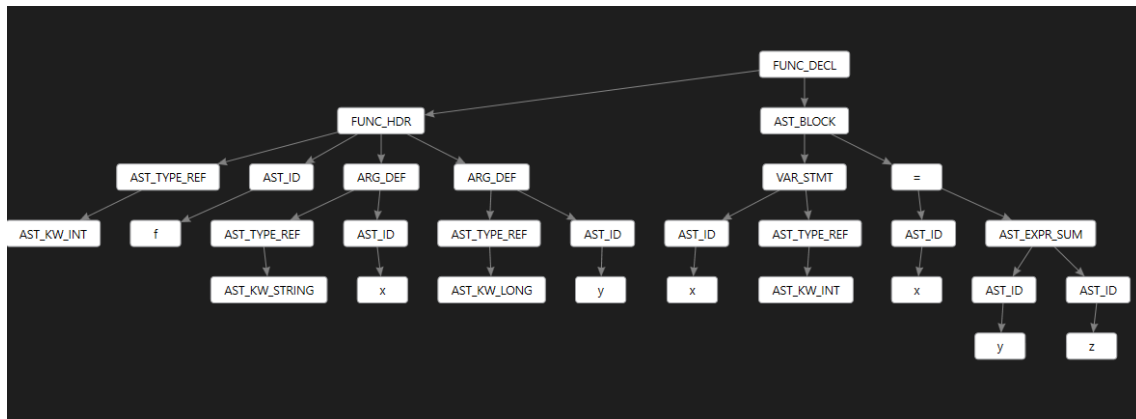
Вывод полученного дерева осуществляется в формате dgml.

Примеры входных и выходных данных:

Входные данные 1.

```
1 function f(x as string, y as long) as int
2     dim x as int
3     x = y + z;
4 end function
5
```

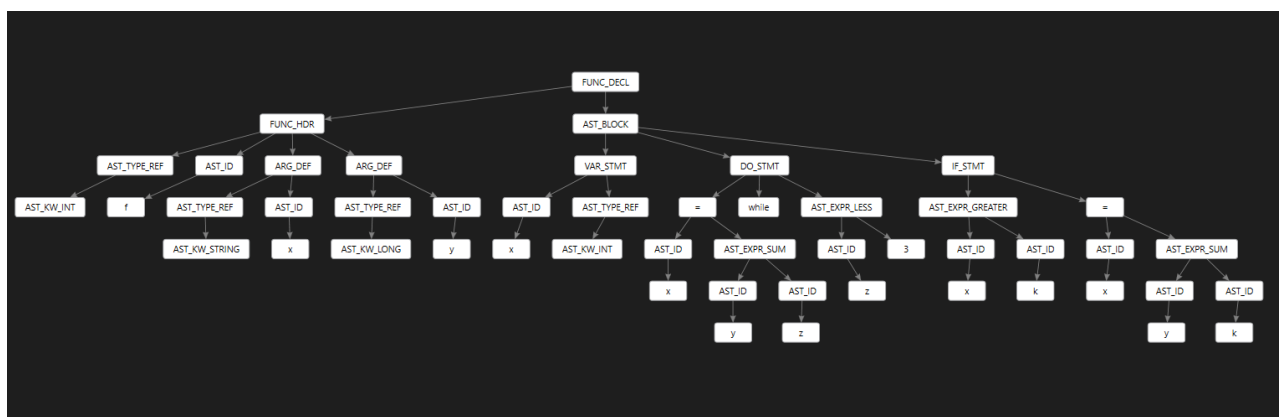
Результат 1.



Входные данные 2.

```
1 function f(x as string, y as long) as int
2     dim x as int
3
4     do
5         dim y as int
6         x = y + z;
7         loop while z < 3
8
9         if x > k then
10             x = y + k;
11         end if
12
13 end function
14
```

Результат 2.



Вывод:

В ходе выполнения лабораторной работы, мною были изучены средства синтаксического анализа, выбран и рассмотрен генератор нисходящих анализаторов для формальных языков – ANTLR3.

С помощью ANTLR3 был реализован модуль, использующий средство синтаксического анализа для разбора языка по варианту, а работоспособность модуля была проверена на тестовой программе.