

НИУ ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1  
по дисциплине  
“Системное программное обеспечение”  
Вариант 3

Выполнил: студент группы Р4114

Былинин Глеб

Преподаватель: Кореньков Юрий Дмитриевич

Санкт-Петербург

2023 г.

<b>Задание:</b>	<b>3</b>
<b>Описание структур данных</b>	<b>5</b>
<b>Пример входных данных и результат обработки</b>	<b>6</b>
<b>Вывод:</b>	<b>8</b>

### **Задание:**

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
  - d. Средство может быть реализовано с нуля, в этом случае оно должно использовать обобщённый алгоритм, управляемый спецификацией
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
  - b. Сообщения об ошибке должны выводиться тестовой программой (не модулем, отвечающим за анализ!) в стандартный поток вывода ошибок
- 5. Результаты тестирования представить в виде отчета, в который включить:
  - a. В части 3 привести описание структур данных, представляющих результат разбора текста (3а)
  - 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

## Описание структур данных

Для обработки элементов синтаксической модели предложенного языка были использованы следующие структуры:

---

```
struct Ast {  
    unsigned short int counter;  
    AstNode* head;  
};
```

```
struct AstNode {  
    int id;  
    char* name;  
    char* value;  
    AstNode* left;  
    AstNode* right;  
};
```

---

Для создания узлов дерева использовалась следующая функция, вызываемая внутри правил bison:

---

```
do: DO listStatement LOOP WHILE expr { $$ =  
ast_create_node(ast, "Do While", "", $2, $5); };
```

---

После того, как все синтаксические элементы помещены в дерево, структура транслируется в формат mermaid-диаграмм и сохраняется в выходной файл.

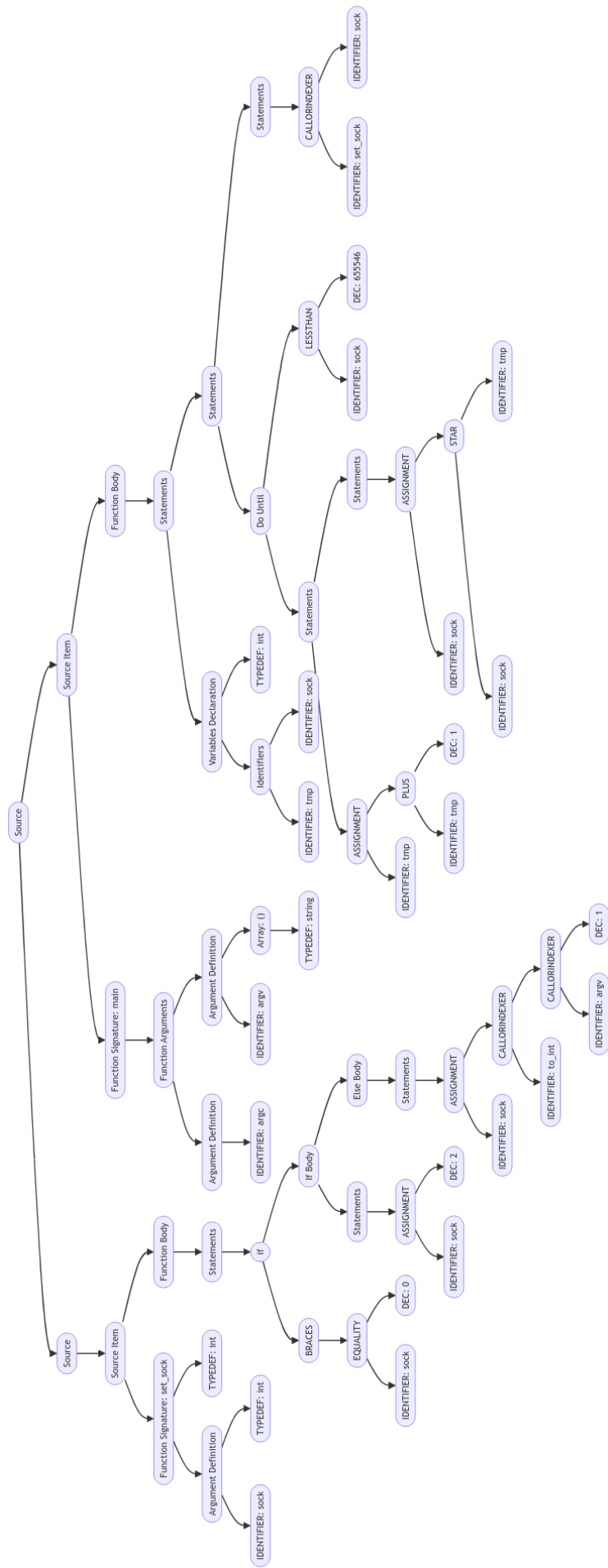
## Пример входных данных и результат обработки

---

```
function set_sock(sock as int) as int
  if (sock == 0) then
    sock = 2;
  else
    sock = to_int(argv(1));
  end if
end function
```

```
function main(argc, argv as string())
  dim tmp, sock as int
  do
    tmp = tmp + 1;
    sock = sock * tmp;
  loop until sock < 655546
  set_sock(sock);
end function
```

---



### **Вывод:**

В ходе выполнения данной лабораторной работы были имплементирован лексер, парсер и алгоритмы построения и вывода в формате mermaid-диаграммы абстрактного синтаксического дерева для текста на языке, соответствующей варианту формальной грамматике.