

Университет ИТМО  
Факультет программной инженерии и компьютерной техники

# Системное программное обеспечение

## Лабораторная работа №1

Выполнил:  
магистрант Рекубрацкий Н.О.  
Группа Р4115  
Преподаватель: Кореньков Ю.Д.

г. Санкт-Петербург  
2023 г

## Оглавление

1. Задание.....	3
2. Основные этапы.....	4
2.1. Выбор средства синтаксического анализа.....	4
2.2. Спецификация для средства синтаксического анализа.....	4
2.3. Реализация модуля, использующего средство синтаксического анализа.....	4
2.4. Реализация тестовой программы для демонстрации работоспособности созданного модуля.....	4
2.5. Примеры исходных анализируемых текстов и соответствующие результаты разбора....	5
3. Вывод.....	6

# 1. 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 обобщенный 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 об 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 (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

## 2. Основные этапы

### 2.1. Выбор средства синтаксического анализа

В качестве средств лексического и грамматического анализа текста выбраны Flex и Bison.

### 2.2. Спецификация для средства синтаксического анализа

Пример правила спецификации:

```
optionalStatements: THEN statement optionalStatements {{$$ = createNode("then", $2, $3, "");}}
| THEN statement {{$$ = createNode("then", $2, NULL, "");}}
| ELSE statement optionalStatements {{$$ = createNode("else", $2, $3, "");}}
| ELSE statement {{$$ = createNode("else", $2, NULL, "");}}
| {{$$ = NULL;}};
```

При разборе языка входного файла правилами Bison, создаются узлы AST-дерева.

### 2.3. Реализация модуля, использующего средство синтаксического анализа

AST-дерево формируется следующей структурой:

```
typedef struct ASTNode ASTNode;
```

```
struct ASTNode {
    char *type;
    ASTNode *left;
    ASTNode *right;
    char *value;
    int id;
};
```

Полученное AST-дерево преобразуется в файл dot-формата, а затем из dot-файла преобразуется в png-файл для графического представления указанного дерева.

### 2.4. Реализация тестовой программы для демонстрации работоспособности созданного модуля

Программа принимает на вход текстовые файлы и на выходе создает 2 файла форматов \*.dot и \*.png, а также выводит сообщения об ошибках, полученных в ходе разбора текста.

```
$ ./result home3.txt
Нераспознанный символ: {
Нераспознанный символ: {
Нераспознанный символ: {
Нераспознанный символ: }
```

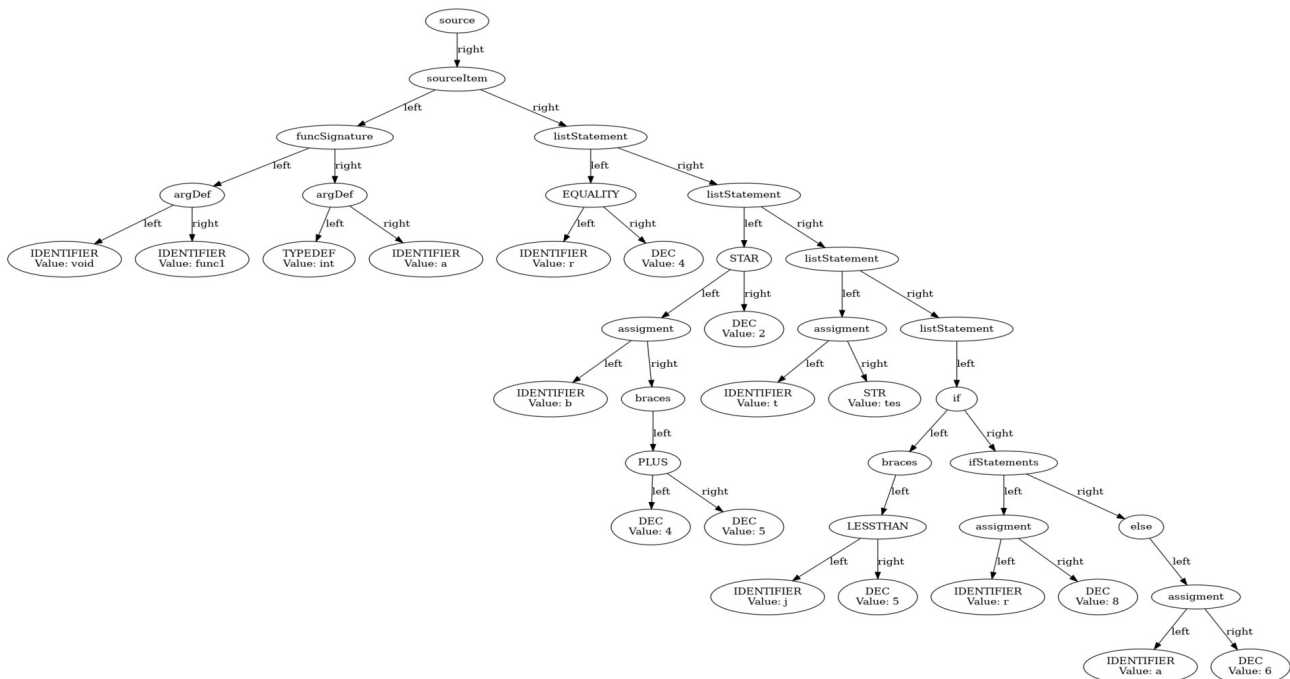
## 2.5. Примеры исходных анализируемых текстов и соответствующие результаты разбора

### Пример 1

Исходный текст (файл — **input1.txt**):

```
void func1(int a)
  r == 4;
  b = (4 + 5) * 2;
  t = "test";
  if (j < 5)
    r = 8;
  else
    a = 6;
```

Результат разбора:



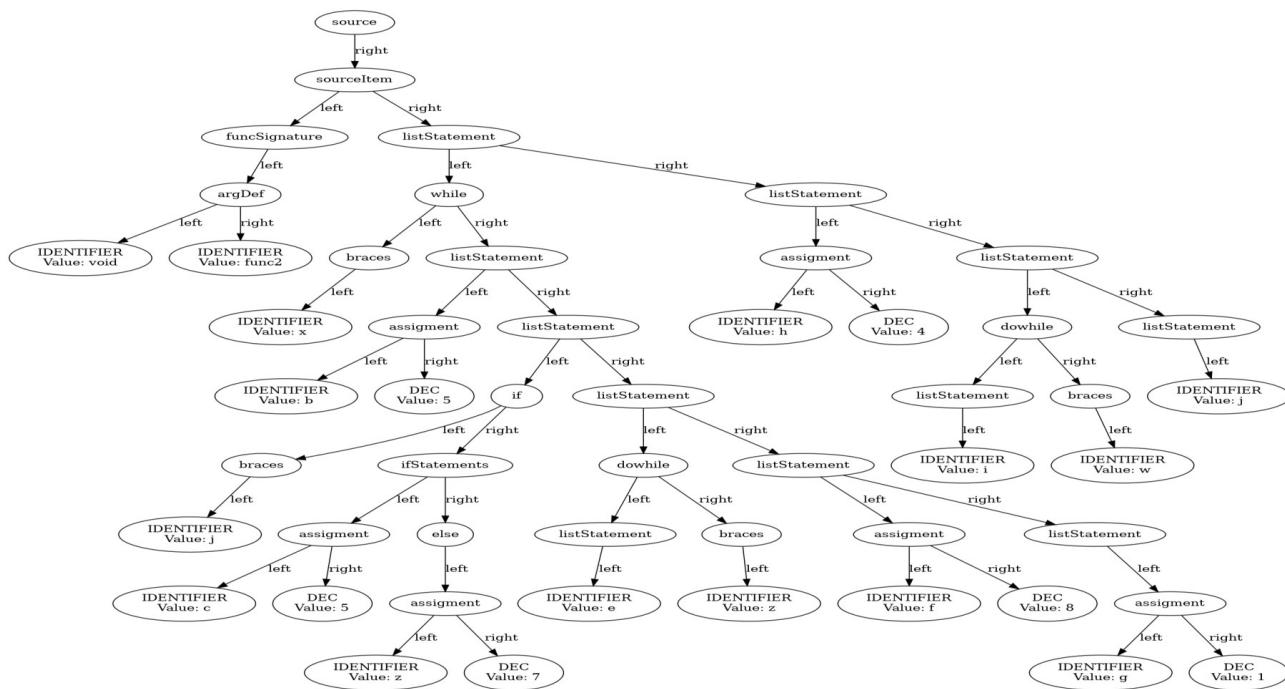
## Пример 2

Исходный текст (файл — **input2.txt**):

```
void func2()
  while (x)
    b = 5;
    if (j)
      c = 5;
    else
      z = 7;
      do e;
      loop
      while (z)
        f = 8;

  g = 1;
wend
  h = 4;
  do i;
  loop
  while (w)
    j;
```

Результат разбора:



### 3. Вывод

В данной работе удалось реализовать программу разбора текста и вывода AST-дерева в графическом представлении в формате \*.png, в соответствии с грамматикой языка текста, с использованием инструментов для анализа текста Flex (лексер) и Bison (парсер).