

Отчёт по лабораторной работе №1.

Системное программное обеспечение.

Самойлов Д. С., Р4116

Вариант 4

1) Структура данных, возвращаемая модулем разбора, представляет собой головной элемент синтаксического дерева разбора и имеет следующую структуру:

```
typedef struct AstNode {  
    int id;  
    char* token;  
    struct AstNode* parent;  
    struct Array* children;  
} AstNode;
```

id – уникальный идентификатор узла

token – содержимое узла

parent – указатель на предка

children – указатель на массив дочерних узлов

Структура Array выглядит следующим образом:

```
typedef struct Array {  
    void* items;  
    size_t itemSize;  
    size_t capacity;  
    size_t size;  
} Array;
```

items – динамический массив элементов массива

itemSize – размер типа элемента массива в байтах

capacity – ёмкость массива

size – размер

2) Для результата разбора потребовалась дополнительная обработка размерностей массива. Для этого была создана функция fixAst, которая находит все узлы с токенами RANGE_LISTS и преобразует структуру дочерних так чтобы можно было обработать многомерные массивы.

3) Примеры результатов разбора:

Пример 1.

```
def checkArrayValues(arr of int array[10])  
  begin  
    while i < 10  
      if arr[i] > 0 then  
        arr[i] = arr[i] * 2;  
      else  
        arr[i] = 0;  
      i = i + 1;  
    end  
  end  
end
```

Результат разбора:

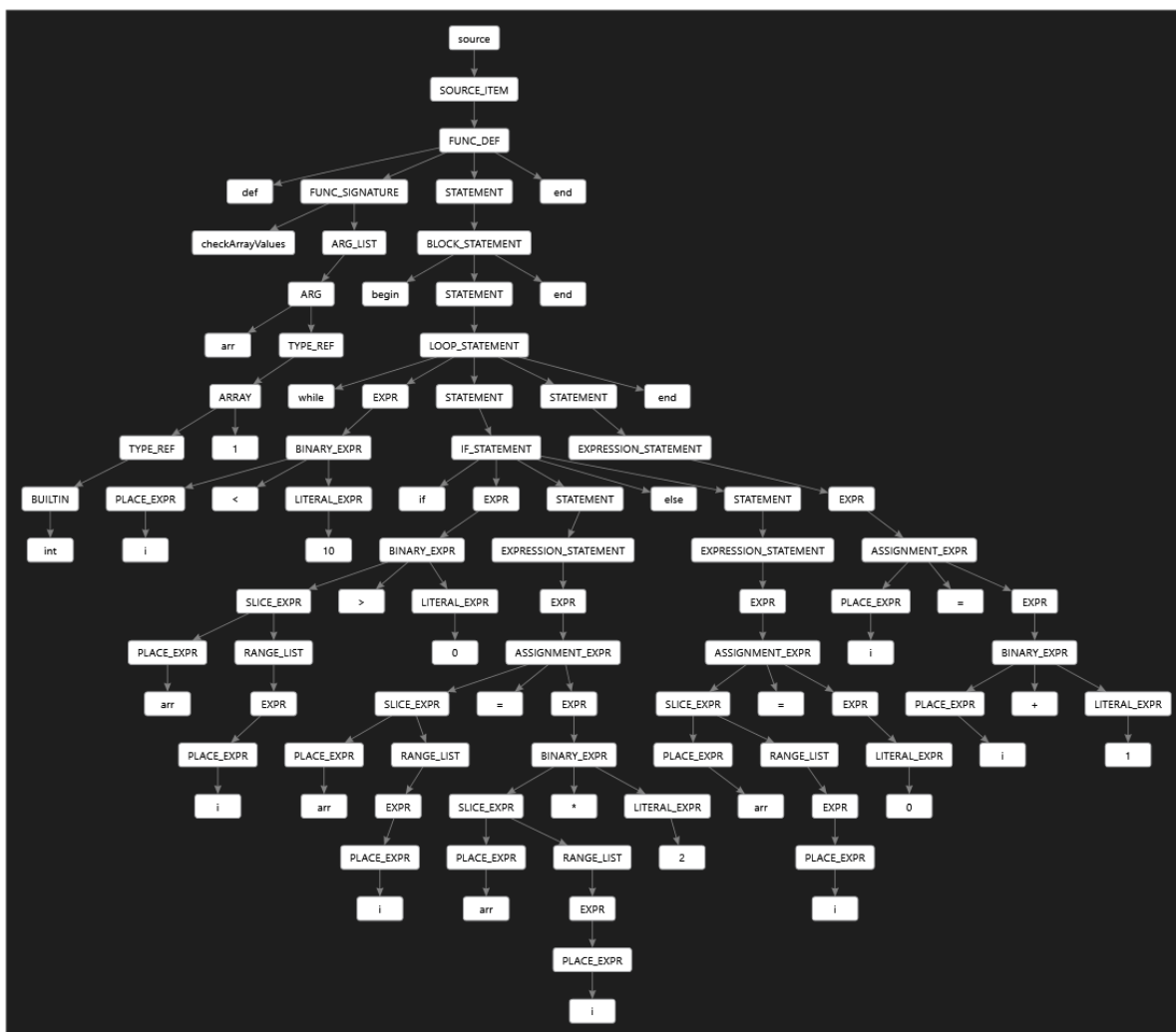


Рисунок 1 – Результат разбора примера 1.

Пример 2. (функция внутри функции)

```
def calculateSum(a of int, b of int) of int
```

```
    if a > 0 then
```

```
        begin
```

```
            def innerCalculation(x of int, y of int) of int
```

```
                return (x + y) * 2;
```

```
        end
```

```

    result = innerCalculation(a, b);

end

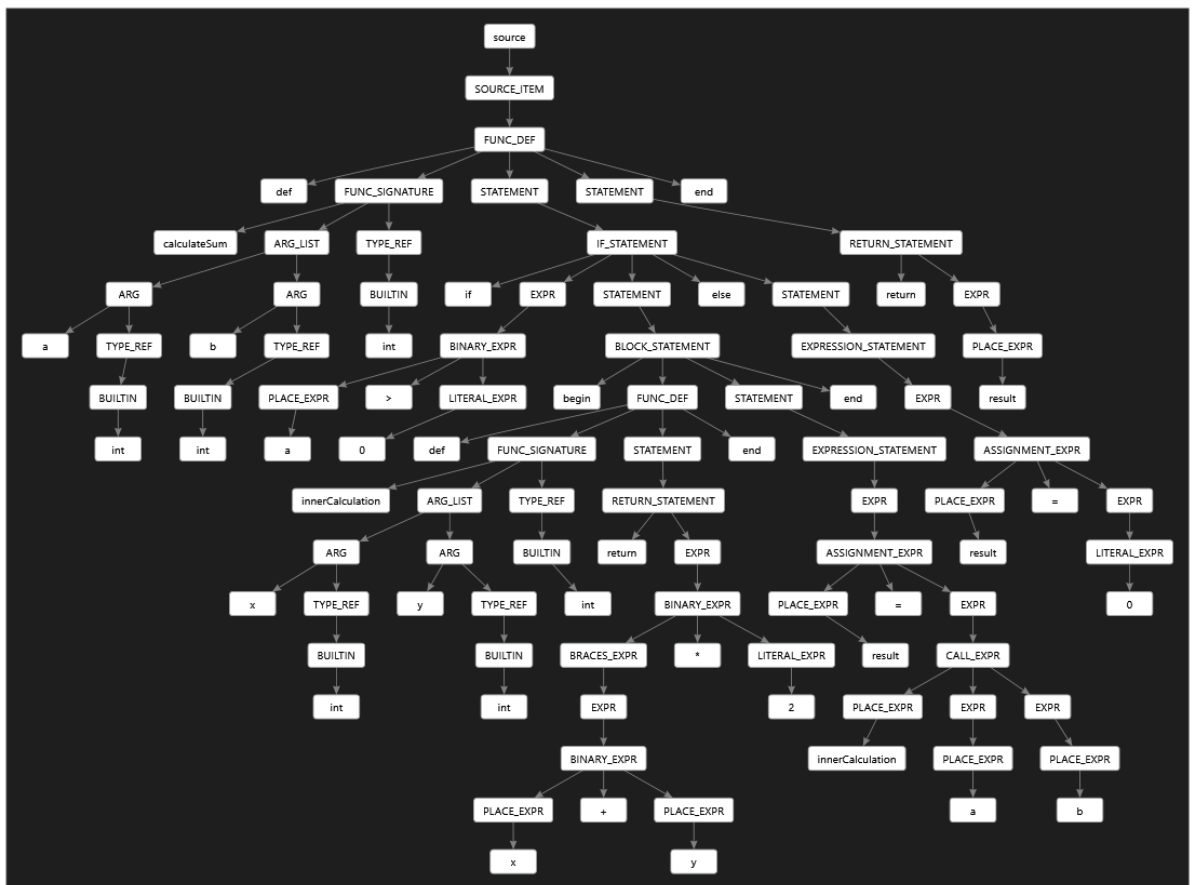
else

    result = 0;

return result;

end

```



Пример 3. (проверка многомерных массивов)

```

def main(argc of int, argv of int array[1], matrix of int array[2]) of int
    print(argv[1]);
    while(i < 10)
        while(j < 10)
            matrix[i, j] = i * j;

```

++j;

end

 $++\mathbf{i};$

end

```
return 0;
```

end

