# Лабораторная работа № 2. Построение промежуточного представления в виде абстрактного синтаксического дерева для языка LispKit

***7.2.1. Цель работы***

Закрепление теоретических сведений и приобретение практических навыков создания промежуточных форм языковых конструкций при выполнении синтаксически управляемого перевода (на примере абстрактного синтаксического дерева языка LispKit).

***7.2.2. Контрольные вопросы***

1. Формы промежуточного представления. Синтаксические деревья и даги.

2. Формы промежуточного представления. Триады и тетррады.

3. Формы промежуточного представления. ПОЛИЗ.

4. В соответствии с рассмотренной в плдпункте 6.3. практикума реализацией конструкций языка LispKit, построить АСД для следующих программ:

4.1*.* (LET (SUM X Y) (SUM (LAMBDA (A B) (ADD A B)

(X (QUOTE 3)) (Y (QUOTE 4)) )

4.2. (COND (ATOM X) (CONS X (QUOTE (A (B C) (D)))

(CAR (CDR X)) )

***7.2.3. Задание на лабораторную работу***

1. Дополнить атрибутную грамматику языка LispKit, построенную в предыдущей лабораторной работе, типами грамматических символов и семантическими правилами для построения абстрактного синтаксического дерева программ.

2. Написать функцию обхода абстрактного синтаксического дерева для вывода программы в текстовом виде. Эта функция получает на вход «регистр» с построенным деревом (атом или указатель на список), а на выходе выдает текст программы в виде строки.

При выполнении задания студенты могут использовать как предложенную в подп. 6.3 структуру дерева, так и структуру, разработанную самостоятельно. Во избежание появления «висячих» ссылок «ненужные» ячейки в процессе построения дерева можно освобождать. Следует также учесть, что каждое *S-*выражение, указанное в грамматике, может быть как атомом, так и списком.

Результат построения абстрактного синтаксического дерева после правильного разбора всего исходного текста в виде «регистра» должен передаваться в основную программу – либо через глобальную переменную, либо через функцию, описанных в интерфейсной части модуля YACC-программы. Функция обхода дерева также должна быть доступна основной программе, а ее реализация помещается в пользовательский раздел YACC-программы.

Результатом работы основной программы является текст программы на языке расширенного лямбда- исчисления, лексикографически совпадающий с текстом исходной программы, но полученный путем обхода построенного абстрактного синтаксического дерева.