Лекция. Лексический анализ

GNU FLEX

IDENT

DEDENT

Количество отступов в python которые преобразует лексер

Открыть файл на чтение и записать его в глобальную переменную YYIN, далее запускаем функцию yylex.

FLEX файл состоит из 3х секций, разделенных %%

Секция объявления

Преобразовать в нормальный вид, как она должна хранится в памяти

Можно разместить код между символами %{ %} (include)

В секции правил, если код размещен в скобках выше, до правил, то он будет выполнен при вызове yylex. (Все переменные советует размещать в этом блоке). Используется для объявления локальных переменных.

Задавать опции.

Есть 2 опции которые можно настроить

%option noyywrap

never-interactive

Можем задавать именованные объявления DIGIT [0-9]

Секция правил

Правило состоит из 2х частей – regex и код, который будет выполняться при срабатывании regex. Они разделяются пробелом или несколькими пробелами

Внутри кода будет переменная yytext – указатель на буфер, которая будет содержать совпадения с regex. Если надо вытащить данные из yytext, то сначала выделяем память(malloc), и копируем туда.

%%{DIGIT}+

Первое правило – если они разной длины, то выигрывает самое длинное

Если у них одинаковая длина – то выигрывает то правила, которое введено первое

Следовательно сначала ключевые слова, потом идентификаторы. Сначала частный случай, потом общий.

Секция пользовательского кода

“ abc $x abc”

“abc “.$x.” abc”

<<EOF>>

Сартовые условия / состояния

INITIAL

<STRING,CHAR>

<STRING>

<\*>

Включающие состояния %s

Исключающие состояния %x

Пример как разобрать многострочный комментарий

% x COMMENT

%%

“/\*” BEGIN(COMMENT);

<COMMENT>[^\*]+

<COMMENT>”\*”+[^\*/]

<COMMENT>”x”+”/” BEGIN(INITIAL);

Задача на 2 лр

Сделать лексический анализатор, которому скармливается файл с кодом на нашем языке, а на выходе набор лексем.