**ГРАММАТИКА**

Язык поддерживает выражения со вложенными скобками, условные операторы if/else, циклы for/while/. Также в языке реализована поддержка функции PRINT для вывода значений одной или нескольких переменных и связанного списка LinkedList. Грамматика языка представлена в Листинге 1.

*Листинг 1 ⎯ Грамматика языка*

(r'[\n]+', 'NEWLINE'),

(r'[ \t]+', None),

(r'#.\*', 'COMMENT'),

(r'[/]\\*([^#]\*)\\*[/]', 'MULTILINE\_COMMENT'),

(r'[-]\*[0-9]+\.[0-9]+', FLOAT),

(r'[-]\*[1-9]+[0-9]\*|0', INT),

(r'\(', 'L\_BRACKET'),

(r'\)', 'R\_BRACKET'),

(r'\{', 'L\_BRACE'),

(r'\}', 'R\_BRACE'),

(r'\[', 'L\_SQUARE\_BRACKET'),

(r'\]', 'R\_SQUARE\_BRACKET'),

(r'%', 'MOD\_OP'),

(r'//', 'DIV\_OP'),

(r'\+', 'PLUS\_OP'),

(r'-', 'MINUS\_OP'),

(r'\\*\\*', 'EXPONENTIATION\_OP'),

(r'\\*', 'MULTIPLICATION\_OP'),

(r'/', 'SLASH\_OP'),

(r'input|<<', 'KW\_INPUT'),

(r'print|>>', 'KW\_PRINT'),

(r'<=', 'LESS\_EQUALLY'),

(r'<', 'LESS'),

(r'>=', 'MORE\_EQUALLY'),

(r'>', 'MORE'),

(r'==', 'EQUALS'),

(r':=|=', 'ASSIGN'),

(r'!=', 'NOT\_EQUALLY'),

(r'\:', 'COLON'),

(r'\;', 'SEMICOLON'),

(r'and', 'KW\_AND'),

(r'or', 'KW\_OR'),

(r'not', 'KW\_NOT'),

(r'if', 'KW\_IF'),

(r'in', 'KW\_IN'),

(r'is', 'KW\_IS'),

(r'else', 'KW\_ELSE'),

(r'while', 'KW\_WHILE'),

(r'for', 'KW\_FOR'),

(r'exit', 'KW\_EXIT'),

(r'func', 'KW\_FUNC'),

(r'List', 'KW\_LIST'),

(r'.pop', 'LL\_POP'),

(r'.len', 'LL\_LEN'),

(r'.size', 'LL\_SIZE'),

(r'.insertAtEnd', 'LL\_INSERT\_END'),

(r'.insertAtHead', 'LL\_INSERT\_HEAD'),

(r'.deleteAtHead', 'LL\_DELETE\_HEAD'),

(r'.delete', 'LL\_DELETE'),

(r'.search', 'LL\_SEARCH'),

(r'.isEmpty', 'LL\_IS\_EMPTY'),

(r'\.', 'POINT'),

(r'\,', 'COMMA'),

(r'[A-Za-z][A-Za-z0-9\_]\*', 'VAR')

lang -> expr+

expr -> body ENDL

body -> expr\_assign | if\_op | while\_op | for\_op | print | linkedlist

value -> VAR | DIGIT

condition -> VAR COMPARE\_OP expr\_value

condition\_in\_br -> L\_BC condition R\_BC

if\_op -> IF condition\_in\_br body+

while\_op -> WHILE condition\_in\_br body+

for\_op -> FOR IN L\_BC (INT | VAR) COMMA (INT | VAR) R\_BC body+

assign -> VAR ASSIGN\_OP expr\_value

print -> PRINT (VAR | DIGIT)\*

linkedlist -> list\_initialize | list\_op

list\_initialize -> LIST VAR

list\_op -> VAR POINT (POP | LEN | SIZE | LL\_INSERT\_END | LL\_INSERT\_HEAD | LL\_DELETE\_HEAD | LL\_DELETE | LL\_SEARCH | LL\_IS\_EMPTY) L\_BC DIGIT? R\_BC