

REFERENCE GRAMMAR

$G(L) = \{V_n, V_t, S, P\}$, where

V_n – nonterminal symbols,

V_t – terminal symbols,

S – starting symbol,

P – finite set of production rules.

The "|" symbol is used to separate the alternatives.

The text between ? ? is the description of the terminal symbol

Symbols written in **bold** are terminal

$V_n = \{ \langle \text{start} \rangle, \langle \text{functions} \rangle, \langle \text{main} \rangle, \langle \text{function} \rangle, \langle \text{parameters} \rangle, \langle \text{code} \rangle, \langle \text{parameter} \rangle, \langle \text{valid_identifier} \rangle, \langle \text{letter} \rangle, \langle \text{identifier} \rangle, \langle \text{digit} \rangle, \langle \text{statement} \rangle, \langle \text{if_statement} \rangle, \langle \text{assignment} \rangle, \langle \text{variable_declaration} \rangle, \langle \text{function_call} \rangle, \langle \text{return} \rangle, \langle \text{bool_condition} \rangle, \langle \text{elif} \rangle, \langle \text{else} \rangle, \langle \text{bool_prefix} \rangle, \langle \text{bool_member} \rangle, \langle \text{bool_operator} \rangle, \langle \text{calculation} \rangle, \langle \text{data_type} \rangle, \langle \text{calculation_prefix} \rangle, \langle \text{calculation_members} \rangle, \langle \text{calculation_member} \rangle, \langle \text{operator} \rangle, \langle \text{numerical} \rangle, \langle \text{string} \rangle, \langle \text{bool} \rangle, \langle \text{int} \rangle, \langle \text{float} \rangle, \langle \text{text} \rangle, \langle \text{char} \rangle, \langle \text{call_parameters} \rangle, \langle \text{call_parameter} \rangle, \langle \text{valid_returns} \rangle \}$

$V_t = \{ \text{fun, a...z, A...Z, 0...9, if, elif, else, (,), \{, \}, _, !, >, <, =, !, >=, <=, !=, ==, \&\&, ||, -, +, *, /, var, ., main, ~, } \}$

$S = \{ \langle \text{start} \rangle \}$

$P = \{$

$\langle \text{start} \rangle \rightarrow \langle \text{functions} \rangle \langle \text{main} \rangle$

$\langle \text{functions} \rangle \rightarrow \varepsilon \mid \langle \text{function} \rangle \mid \langle \text{function} \rangle \langle \text{functions} \rangle$

$\langle \text{function} \rangle \rightarrow \text{fun } \langle \text{valid_identifier} \rangle (\langle \text{parameters} \rangle) \{ \langle \text{code} \rangle \}$

$\langle \text{parameters} \rangle \rightarrow \varepsilon \mid \langle \text{parameter} \rangle$

$\langle \text{parameter} \rangle \rightarrow \langle \text{valid_identifier} \rangle \mid \langle \text{parameter} \rangle, \langle \text{parameter} \rangle$

$\langle \text{valid_identifier} \rangle \rightarrow \langle \text{letter} \rangle \langle \text{identifier} \rangle \mid _ \langle \text{identifier} \rangle$

$\langle \text{identifier} \rangle \rightarrow \langle \text{letter} \rangle \mid _ \mid \langle \text{digit} \rangle \mid \varepsilon \mid \langle \text{identifier} \rangle \langle \text{identifier} \rangle$

$\langle \text{letter} \rangle \rightarrow \text{a...z} \mid \text{A...Z}$

$\langle \text{digit} \rangle \rightarrow \text{0...9}$

$\langle \text{code} \rangle \rightarrow \langle \text{statement} \rangle \mid \langle \text{statement} \rangle \langle \text{code} \rangle$

$\langle \text{statement} \rangle \rightarrow \langle \text{if_statement} \rangle \mid \langle \text{assignment} \rangle \mid \langle \text{variable_declaration} \rangle \mid \langle \text{function_call} \rangle \mid \langle \text{return} \rangle$

$\langle \text{if_statement} \rangle \rightarrow \text{if} (\langle \text{bool_condition} \rangle) \{ \langle \text{code} \rangle \} \langle \text{elif} \rangle \langle \text{else} \rangle$

$\langle \text{elif} \rangle \rightarrow \varepsilon \mid \text{elif} (\langle \text{bool_condition} \rangle) \{ \langle \text{code} \rangle \} \mid \langle \text{elif} \rangle \langle \text{elif} \rangle$

$\langle \text{else} \rangle \rightarrow \varepsilon \mid \text{else} (\langle \text{bool_condition} \rangle) \{ \langle \text{code} \rangle \}$

$\langle \text{bool_condition} \rangle \rightarrow \langle \text{bool_prefix} \rangle \langle \text{bool_member} \rangle \langle \text{bool_operator} \rangle \langle \text{bool_condition} \rangle \mid$

$\langle \text{bool_prefix} \rangle \langle \text{bool_member} \rangle$

<bool_member> → <valid_identifier> | <bool> | <calculation>
 <bool_prefix> → ε | !
 <bool_operator> → > | < | == | != | >= | <= | && | ||
 <assignment> → <valid_identifier> = <calculation> | <valid_identifier> = <bool_condition> |
 <valid_identifier> = <data_type> | <valid_identifier> = <function_call>
 <calculation> → <calculation_prefix><calculation_members>
 <calculation_prefix> → ε | -
 <calculation_members> → <calculation_member><operator><calculation_members> |
 <calculation_member>
 <operator> → + | - | * | /
 <calculation_member> → <numerical> | <valid_identifier> | <function_call>
 <data_type> → <string> | <numerical> | <bool>
 <numerical> → <int> | <float>
 <int> → <digit> | <digit><int>
 <float> → <int>.<int> | <int>
 <string> → ”<text>”
 <text> → <char> | <char><text>
 <char> → ε | ? any visible ASCII character ?
 <variable_declaration> → **var** <variables>
 <variables> → <valid_identifier> | <assignment> | <variables>, <variables>
 <function_call> → <valid_identifier>(<call_parameters>)
 <call_parameters> → ε | <call_parameter>
 <call_parameter> → <valid_identifier> | <data_type> | <function_call> | <call_parameter>,
 <call_parameter>
 <return> → ~ <valid_returns>
 <valid_returns> → <data_type> | <calculation> | <function_call> | <bool_condition>
 <main> → **fun main**(<parameters>){<code>}
 }