Gramatica Limbajului IMP

<program> -> "%begin"<declarations><statements><procedures>"%end""%of""%program"

<program> -> "%begin"<declarations>"%end""%of""%program"

<program> -> "%begin"<declarations><statements>"%end""%of""%program"

<program> -> "%begin"<declarations><procedures>"%end""%of""%program"

<program> -> "%begin"<statements><procedures>"%end""%of""%program"

<program> -> "%begin"<statements>"%end""%of""%program"

<program> -> "%begin"<procedures>"%end""%of""%program"

<program> -> "%begin""%end""%of""%program"

<declaration> -> <delcaration><declarations>

<declarations> -> ε

<declaration> -> <var-decl>

<declaration> -> <proc-decl>

<declaration> -> <proc-defn>

<var-decl> -> <simple-decl><s>

<var-decl> -> <array-decl><s>

<simple-decl> -> <type><name><namelist-comma>

<namelist-comma> -> <namelist-comma>','<type><name>

<namelist-comma> -> ε

<array-decl> -> <type> "%array" <array-name><array-name-comma-list>

<array-name-comma-list> -> <array-name-comma-list>','<type> "%array" <array-name>

<array-name-comma-list> -> ε

<array-name> -> <name> <apair>

<apair> -> '(' <range> <rangelist-comma> ')'

<rangelist-comma> -> <rangelist-comma>','<range>

<rangelist-comma> -> ε

<range> -> <expr> ':' <expr>

<type> -> "%integer"

<type> -> "%short" "%integer"

<type> -> "%long" "%integer"

<type> -> "%byte"

<type> -> "%real"

<type> -> "%long" "%real"

<type> -> "%string" '(' <expr> ')'

<proc-decl> -> <routine-decl> <s>

<proc-decl> -> <func-decl> <s>

<routine-decl> -> "%routine" "%spec" <name> '(' <param-list> ')'

<routine-decl> -> "%routine" "%spec" <name> '(' ')'

<func-decl> -> "%function" "%spec" <name> '(' <param-list> ')'

<func-decl> -> "%function" "%spec" <name> '(' ')'

<param-list> -> <parameter><parameter-list-comma>

<parameter-list-comma> -> <parameter-list-comma>','<parameter>

<parameter-list-comma> -> ε

<parameter> -> <type> <namelist>

<parameter> -> <rt> <namelist> '(' <param-list> ')'

<parameter> -> <rt> <namelist>

<rt> -> "%routine"

<rt> -> <type> "%function"

<statements> -> <statement><statements>

<statements> -> ε

<statement> -> <block>

<statement> -> <exec-instr>

<statement> -> <cond-instr>

<statement> -> <loop-instr>

<procedures> -> <proc-defn><procedures>

<procedures> -> ε

<proc-defn> -> <routine-defn>

<proc-defn> -> <function-defn>

<routine-defn> -> <routine-head> <declarations> <statements> <procedures> "%end"

<routine-defn> -> <routine-head> <declarations> "%end"

<routine-defn> -> <routine-head> <statements> "%end"

<routine-defn> -> <routine-head> <procedures> "%end"

<routine-defn> -> <routine-head> <declarations> <statements> "%end"

<routine-defn> -> <routine-head> <statements> <procedures> "%end"

<routine-defn> -> <routine-head> <declarations> <procedures> "%end"

<routine-defn> -> <routine-head> "%end"

<routine-head> -> "%routine" <name> '(' <param-list> ')'

<routine-head> -> "%routine" <name> '(' ')'

<function-defn> -> <function-head> <declarations> <func-statements> <procedures> "%end"

<function-defn> -> <function-head> <declarations> <func-statements> "%end"

<function-defn> -> <function-head> <func-statements> <procedures> "%end"

<function-defn> -> <function-head> <func-statements> "%end"

<function-head> -> "%function" <name> '(' <param-list> ')'

<function-head> -> "%function" <name> '(' ')'

<func-statements> -> <statements> <result-instr> <statements>

<func-statements> -> <result-instr> <statements>

<func-statements> -> <statements> <result-instr>

<func-statements> -> <result-instr>

<block> -> "%begin" <declarations> <statements> <procedures> "%end" <s>

<block> -> "%begin" <declarations> "%end" <s>

<block> -> "%begin" <statements> "%end" <s>

<block> -> "%begin" <procedures> "%end" <s>

<block> -> "%begin" <declarations> <statements> "%end" <s>

<block> -> "%begin" <statements> <procedures> "%end" <s>

<block> -> "%begin" <declarations> <procedures> "%end" <s>

<block> -> "%begin" "%end" <s>

<exec-instr> -> "%return" <s>

<exec-instr> -> "%stop" <s>

<exec-instr> -> "%exit" <s>

<exec-instr> -> "%continue" <s>

<exec-instr> -> <name> <app> '=' <expr> <s>

<exec-instr> -> <name> '=' <expr> <s>

<exec-instr> -> <name> <app> <s>

<result-instr> -> "%result" '=' <expr> <s>

<cond-instr> -> <%iu> <condition> "%then" <then-instr> <else-part> <s>

<cond-instr> -> <%iu> <condition> "%then" <then-instr> <s>

<%iu> -> "%if"

<%iu> -> "%unless"

<then-instr> -> <exec-instr>

<then-instr> -> <compound-stmt>

<compound-stmt> -> "%start" <statements> "%finish"

<compound-stmt> -> "%start" "%finish"

<else-part> -> "%else" <else-instr>

<else-instr> -> <cond-instr>

<else-instr> -> <exec-instr>

<else-instr> -> <compound-stmt>

<loop-instr> -> <cycle> <s>

<loop-instr> -> <while-for> <cycle> <s>

<cycle> -> "%cycle" <statements> "%repeat" "%until" <condition>

<cycle> -> "%cycle" <statements> "%repeat"

<while-for> -> "%while" <condition>

<while-for> -> "%for" <name> '=' <expr>',' <expr>',' <expr>

<condition> -> <sc>

<condition> -> <cc>

<cc> -> <sc><sc-list>

<sc-list> -> "%and" <sc>

<sc-list> -> "%and" <sc><sc-list>

<sc-list> -> ε

<cc> -> <sc><sc-list>

<sc-list> -> "%or" <sc>

<sc-list> -> "%or" <sc><sc-list>

<sc-list> -> ε

<sc> -> <expr> <comp> <expr>

<sc> -> "%not" <sc>

<sc> -> '(' <cc> ')'

<comp> -> '='

<comp> -> '#'

<comp> -> "\="

<comp> -> "<>"

<comp> -> '>'

<comp> -> ">="

<comp> -> '<'

<comp> -> "<="

<expr> -> <unop> <operand> <bop-operand-list>

<expr> -> <operand> <bop-operand-list>

<bop-operand-list> -> <bop> <operand><bop-operand-list>

<bop-operand-list> -> ε

<operand> -> <name> <app>

<operand> -> <name>

<operand> -> <const>

<operand> -> '(' <expr> ')'

<app> -> '(' <expr> <expr-list-comma> ')'

<expr-list-comma> -> <expr-list-comma>','<expr>

<expr-list-comma> -> ε

<unop> -> '+'

<unop> -> '-'

<unop> -> '\'

<unop> -> '~'

<bop> -> '\*'

<bop> -> '/'

<bop> -> "//"

<bop> -> '+'

<bop> -> '-'

<bop> -> '\'

<bop> -> "\\"

<bop> -> "\*\*"

<bop> -> ">>"

<bop> -> "<<"

<bop> -> '!'

<bop> -> "!!"

<bop> -> '&'

<const> -> <int-const>

<const> -> <dec-const>

<const> -> <char-const>

<const> -> <str-const>

<int-const> -> <plus><integer>

<int-const> -> <integer>

<plus> -> '+'

<plus> -> '-'

<dec-const> -> <int-const> <frac> '@' <int-const>

<dec-const> -> <int-const>

<dec-const> -> <frac>

<dec-const> -> '@' <int-const>

<dec-const> -> <int-const>] [<frac>]

<dec-const> -> <frac> '@' <int-const>

<dec-const> -> <int-const> '@' <int-const>

<frac> -> '.' <integer>

<frac> -> '.'

<char-const> -> ' <charnl> '

<str-const> -> <str-const>" <charnl> "

<str-const> -> ε

<integer> -> <digit><integer>

<integer> -> ε

<name> -> <letter><letdig-list>

<letdig-list> -> <letdig><letdig-list>

<letdig-list> -> ε

<letdig> -> <letter>

<letdig> -> <digit>

<letter> -> 'A'

<letter> -> 'B'

<letter> -> 'C'

<letter> -> 'D'

<letter> -> 'E'

<letter> -> 'F'

<letter> -> 'G'

<letter> -> 'H'

<letter> -> 'I'

<letter> -> 'J'

<letter> -> 'K'

<letter> -> 'L'

<letter> -> 'M'

<letter> -> 'N'

<letter> -> 'O'

<letter> -> 'P'

<letter> -> 'Q'

<letter> -> 'R'

<letter> -> 'S'

<letter> -> 'T'

<letter> -> 'U'

<letter> -> 'V'

<letter> -> 'W'

<letter> -> 'X'

<letter> -> 'Y'

<letter> -> 'Z'

<letter> -> 'a'

<letter> -> 'b'

<letter> -> 'c'

<letter> -> 'd'

<letter> -> 'e'

<letter> -> 'f'

<letter> -> 'g'

<letter> -> 'h'

<letter> -> 'i'

<letter> -> 'j'

<letter> -> 'k'

<letter> -> 'l'

<letter> -> 'm'

<letter> -> 'n'

<letter> -> 'o'

<letter> -> 'p'

<letter> -> 'q'

<letter> -> 'r'

<letter> -> 's'

<letter> -> 't'

<letter> -> 'u'

<letter> -> 'v'

<letter> -> 'w'

<letter> -> 'x'

<letter> -> 'y'

<letter> -> 'z'

<digit> -> '0'

<digit> -> '1'

<digit> -> '2'

<digit> -> '3'

<digit> -> '4'

<digit> -> '5'

<digit> -> '6'

<digit> -> '7'

<digit> -> '8'

<digit> -> '9'

<namelist> -> <name><namelist-comma>

<namelist-comma> -> <namelist-comma>','<name>

<namelist-comma> -> ε

<text> -> <text><character>

<text> -> ε

<comment> -> "%comment" <text> <newline>

<comment> -> '!' <text> <newline>

<s> -> ';'