

Specifiche grammaticali e lessicali del nostro linguaggio MyFun

***** Specifica grammaticale *****

Program ::= VarDeclList FunList Main

VarDeclList ::= /* empty */
| VardDecl VarDeclList

Main ::= MAIN VarDeclList StatList END MAIN SEMI

FunList ::= /* empty */
| Fun FunList

VarDecl ::= Type IdListInit SEMI
| VAR IdListInitObbl SEMI

Type ::= INTEGER | BOOL | REAL | STRING

IdListInit ::= ID
| IdListInit COMMA ID
| ID ASSIGN Expr
| IdListInit COMMA ID ASSIGN Expr

IdListInitObbl ::= ID ASSIGN Const
| IdListInitObbl COMMA ID ASSIGN Const

Const ::= INTEGER_CONST | REAL_CONST | TRUE | FALSE | STRING_CONST

Fun ::= FUN ID LPAR ParamDeclList RPAR COLON Type
VarDeclList StatList END FUN SEMI
| FUN ID LPAR ParamDeclList RPAR
VarDeclList StatList END FUN SEMI

ParamDeclList ::= /*empty */
| NonEmptyParamDeclList

NonEmptyParamDeclList ::= ParDecl
| NonEmptyParamDeclList COMMA ParDecl

ParDecl ::= Type ID
| OUT Type ID

StatList ::= Stat
| Stat StatList

Stat ::= IfStat SEMI
| WhileStat SEMI
| ReadStat SEMI
| WriteStat SEMI
| AssignStat SEMI
| CallFun SEMI
| RETURN Expr SEMI

| /* empty */

IfStat ::= IF Expr THEN VarDeclList StatList Else END IF

Else ::= /* empty */
| ELSE VarDeclList StatList

WhileStat ::= WHILE Expr LOOP VarDeclList StatList END LOOP

ReadStat ::= READ IdList Expr // Expr deve essere di tipo stringa
| READ IdList

IdList ::= ID
| IdList COMMA ID

WriteStat ::= WRITE Expr
| WRITELN Expr
| WRITET Expr
| WRITEB Expr

AssignStat ::= ID ASSIGN Expr

CallFun ::= ID LPAR ExprList RPAR
| ID LPAR RPAR

ExprList ::= Expr
| Expr COMMA ExprList
| OUTPAR ID
| OUTPAR ID COMMA ExprList

Expr ::= TRUE
| FALSE
| INTEGER_CONST
| REAL_CONST
| STRING_CONST
| ID
| CallFun
| Expr PLUS Expr
| Expr MINUS Expr
| Expr TIMES Expr
| Expr DIV Expr
| Expr DIVINT Expr
| Expr AND Expr
| Expr POW Expr
| Expr STR_CONCAT Expr
| Expr OR Expr
| Expr GT Expr
| Expr GE Expr
| Expr LT Expr
| Expr LE Expr
| Expr EQ Expr
| Expr NE Expr
| MINUS Expr
| NOT Expr

***** Specifica lessicale *****+

I commenti iniziano con "#" or "//"

Un blocco di comment è delimitato da #* e #

MAIN 'main'
ID /[\$_A-Za-z][\$_A-Za-z0-9]*/
INTEGER 'integer'
STRING 'string'
REAL 'real'
BOOL 'bool'
LPAR '('
RPAR ')'
COLON ':'
FUN 'Fun'
END 'end'
IF 'if'
THEN 'then'
ELSE 'else'
WHILE 'while'
LOOP 'loop'
READ '%'
WRITE '?'
WRITELN '?.'
WRITEB '?,'
WRITET '?:'
ASSIGN ':='
PLUS '+'
MINUS '-'
TIMES '*'
DIVINT 'div'
DIV '/'
POW '^'
STR_CONCAT '&'
EQ '='
NE '<>' or '!='
LT '<'
LE '<='
GT '>'
GE '>='
AND 'and'
OR 'or'
NOT 'not'
NULL 'null'
 TRUE 'true'
 FALSE 'false'
 INTEGER_CONST any integer number (sequence of decimal digits)
 REAL_CONST any real number
 STRING_CONST any string between " or between '
SEMI ';' ;

COMMA ','
RETURN 'return'
OUTPAR '@'
VAR 'var'
OUT 'out'
LOOP 'loop'

PRECEDENZE (dalla più alta alla più bassa) ed ASSOCIATIVITA' (sinistra se non indicata)

()
^ #destra
* / div
+ -
&
= != <> < <= > >= #nessuna
not #destra
and
or