

Specifiche lessicali e grammaticali del linguaggio Toy

```
Program ::= VarDeclList ProcList

VarDeclList ::= /* empty */ | VarDecl VarDeclList

ProcList ::= Proc | Proc ProcList

VarDecl ::= Type IdListInit SEMI

Type ::= INT | BOOL | FLOAT | STRING

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

Proc ::= PROC ID LPAR ParamDeclList RPAR ResultTypeList COLON
      | PROC ID LPAR RPAR ResultTypeList COLON
      | PROC ID LPAR RPAR ResultTypeList COLON
        VarDeclList StatList Return ReturnExprs CORP SEMI

ResultTypeList ::= ResultType
                | ResultType COMMA ResultTypeList

ReturnExprs ::= ExprList
              | /* empty */

ExprList ::= Expr
           | Expr COMMA ExprList

ParamDeclList ::= ParDecl | ParamDeclList SEMI ParDecl

ParDecl ::= Type IdList

IdList ::= ID | IdList COMMA ID

ResultType ::= Type | VOID

StatList ::= Stat | Stat StatList

Stat ::= IfStat SEMI
      | WhileStat SEMI
      | ReadInStat SEMI
      | WriteStat SEMI
      | AssignStat SEMI
      | CallProc SEMI
      | /* empty */

IfStat ::= IF Expr THEN StatList ElifList Else FI

ElifList ::= /* empty */ | Elif ElifList

Elif ::= ELIF Expr THEN StatList

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

WhileStat ::= WHILE StatList Return Expr DO StatList OD
           | WHILE Expr DO StatList OD

ReadInStat ::= READ LPAR IdList RPAR

WriteStat ::= WRITE LPAR ExprList RPAR

AssignStat ::= IdList ASSIGN ExprList

CallProc ::= ID LPAR ExprList RPAR
           | ID LPAR RPAR
```

```
Expr ::= NULL
      | TRUE
      | FALSE
      | INT_CONST
      | FLOAT_CONST
      | STRING_CONST
      | ID
      | CallProc
      | Expr PLUS Expr
      | Expr MINUS Expr
      | Expr TIMES Expr
      | Expr DIV Expr
      | Expr AND 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

```
SEMI ';'
COMMA ','
ID      Any sequence of letters, digits and underscores, starting with
        a letter or an underscore
INT  'int'
STRING 'string'
FLOAT 'float'
BOOL 'bool'
PROC 'proc'
LPAR '('
RPAR ')'
COLON ':'
PROC 'proc'
CORP 'corp'
VOID 'void'
IF 'if'
THEN 'then'
ELIF 'elif'
FI 'fi'
ELSE 'else'
WHILE 'while'
DO 'do'
OD 'od'
READ 'readln'
WRITE 'write'
ASSIGN ':='
PLUS '+'
MINUS '-'
TIMES '*'
DIV '/'
EQ '='
NE '<>'
LT '<'
LE '<='
GT '>'
GE '>='
AND '&&'
OR '||'
NOT '!'
NULL 'null'
TRUE 'true'
FALSE 'false'
INT_CONST  any integer number (sequence of decimal digits)
FLOAT_CONST any floating point number
STRING_CONST any string between "
```

I commenti sono nella forma `/* */`
Le stringhe sono nella forma `" ... "`

Viene Lanciato errore lessicale nei seguenti casi:

- "Stringa costante non completata" nel caso il programma input presenti una stringa costante aperta ma non chiusa (es. "questa è una stringa non chiusa).
- "Commento non chiuso" nel caso il programma input presenti un commento non chiuso (es. `/* questo è un commento non chiuso)`