

<http://www.bottlecaps.de/rr/ui>

Module ::= module id '{' (Vars | Func)* Main '}'

Type ::= bool | int | float | char | string

Statute ::= Assignment | Condition | Read | Write | Call | Cycle

Exp ::= Term (('+' | '-') Term)*

Term ::= Factor (('*' | '/') Factor)*

Factor ::= (('+' | '-')? Constant) | ('(' Expression ')')

Expression ::= Exp (('<' | '>' | '<=' | '>=' | '==' | '!=' | '||' | '&&') Exp)?

Cycle ::= while '(' Expression ')' Block

Call ::= id '(' (Exp (',' Exp)*)? ')' ';' ;

Main ::= 'main' Block

Block ::= '{' Statute* '}'

Read ::= 'read' '(' id ')' ';' ;

Write ::= 'print' '(' Exp ')' ';' ;

Vars ::= Type id ('[' cteN ']')? '=' Constant (',' id ('[' cteN ']')? '=' Constant)* ';' ;

Func ::= func (void | Type) id '(' Arguments? ')' '{' Statute* (return Exp ';')? '}'

Arguments ::= Type id (',' Type id)*

Constant ::= Cte

Condition ::= 'if' '(' Expression ')' Block (('elseif' '(' Block ')')* ('else' Block)?)

Cte ::= (id | cteN | cteS | VarArr)

Assignment ::= (id | VarArr) '=' (Call | Expression) ';' ;

VarArr ::= id '[' Exp ']'