Eduardo de Figueiredo Júnior
Matheus de Souza
Pedro Hardeman

Gramática---

S-> {D / C}*

D-> {byte / string / boolean / integer} id [=[-]valor] {, id [=[i] valor]}*';' / const id = [-] valor';'

C->'{' {CA / CR / CT / CL / CN}* '}' / CA / CR / CL / CT / CN

CA-> id '=' EXP ';'

CR-> 'while' EXP C

CT-> 'if' EXP 'then' C ['else' C]

CL-> 'readln' ',' id ';' / 'write' ',' EXP {',' EXP}* ';' / 'writeln' ',' EXP {',' EXP}* ';'

CN-> ';'

EXPS-> ['+' / '-'] T {['+' / '-' / 'or'] T}*

EXP-> EXPS [['<' / '>' / '<=' / '>=' / '<>' / '=='] EXPS]

T-> F {('*' / '/' / 'and') F}*

F-> valor / id / not F /'(' EXP ')'/ TRUE / FALSE