EBNF Grammar

Blue indicates a terminal token

Grey indicates a meta operator

Black indicates a non-terminal

func -> func id (paramList) block

paramList -> (param | param ,)*

param -> identifier : (type | identifier)

identifier -> ID(String)

type -> TYPE_INT32 | TYPE_FLT32 | TYPE_CHAR | TYPE_BOOL

let -> KW_let Program -> $(\mathrm{func} \mid \mathrm{let})^{+}$

identifier (: type (= identifier)) ;

block -> [(statement)*]

statement -> let | if | return | while | print | assign | funcCall

if -> KW_IF expression block KW_ELSE block

return -> KW_RETURN expression ;

while -> KW_WHILE expression block

print -> KW_PRINT expression ;

assign -> identifier = expression ;

expression -> (identifier | literal) expressionTail

literal -> LIT_INT32(i32) | LIT_FLT32(f32) | LIT_CHAR(char) | LIT_STRING(String) | LIT_BOOL(bool)

expressionTail -> $(\mathrm{operator} \mid \mathrm{identifier})^{*}$

operator -> OP_ADD | OP_NUL | OP_SUB | OP_LT | OP_GT | OP_EQ | OP_NGT | OP_NLT | OP_NEQ

funcCall -> identifier (arguments) ;

arguments -> ($\mathrm{identifier}\ (\,,\mathrm{identifier})^{*}$)