VSL Language Specification
TDT4205, Spring 2015

Syntax

Whitespace consists of the characters '\t', '\n' and ' '. It is ignored after lexical analysis.

Comments start with the sequence '//', and last until the next '\n'.

Keywords are FUNC, PRINT, RETURN, NEW, IF, THEN, ELSE, END, WHILE, DO, END, FOR, TO, INT, BOOL, FLOAT, START, VOID, ARRAY.

Operators are assignment, ':=', the basic arithmetic operators '+','−','*','/', the comparison operators '>','<','<=','>=','==','!=' and the logical operators '&&' (and),'||' (or), '!' (not) and method/field selection '.'.

Integers are sequences of one or more decimal digits, e.g. '0' through '9'.

Floats are sequences of one or more decimal digits, then a decimal point ('.'), then one or more decimal digits.

Boolean constants are TRUE and FALSE.

Strings are sequences of arbitrary characters (except newline, '\n') and escaped double quotes ('\"'), enclosed in double quotes, '"'. It's an error to break a string across multiple lines.

Identifiers are sequences of at least one letter followed by an arbitrary sequence of letters and digits. Letters are defined as the upper and lower case english alphabet, 'A' through 'Z' and 'a' through 'z' as well as underscore, '_'. Digits are the decimal digits, as above.

Punctuation includes ')', '(', ',', ';', '[', ']'

Grammar

The syntactic structure of the language is given in the following grammar. `Courier font` is used for terminals. For readability, some operators are represented by their lexeme (e.g. ':='), not the token returned by the scanner (e.g. ASSIGN).

program	::=	function_list
function	::=	type FUNC variable '(' parameter_list ')' START statement_list END
function_list	::=	function_list function
		ε
statement_list	::=	statement
		statement_list statement
variable_list	::=	declaration_statement
		variable_list ',' declaration_statement
expression_list	::=	expression
		expression_list ',' expression
parameter_list	::=	variable_list
		ε
argument_list	::=	expression_list
		ε

```
statement    ::=   declaration_statement ';'
             |     assignment_statement ';'
             |     if_statement
             |     while_statement
             |     for_statement
             |     print_statement ';'
             |     return_statement ';'
             |     call ';'
declaration_statement   ::=   type variable
assignment_statement    ::=   lvalue ASSIGN expression
if_statement       ::=   IF expression THEN statement_list END
                   |     IF expression THEN statement_list ELSE statement_list END
while_statement    ::=   WHILE expression DO statement_list END
for_statement   ::=   FOR assignment_statement TO expression DO statement_list END
return_statement   ::=   RETURN expression
print_statement    ::=   PRINT expression_list
expression    ::=   constant
              |     expression '+' expression
              |     expression '-' expression
              |     expression '*' expression
              |     expression '/' expression
              |     expression '>' expression
              |     expression '<' expression
              |     expression '==' expression
              |     expression '!=' expression
              |     expression '>=' expression
              |     expression '<=' expression
              |     expression '&&' expression
              |     expression '||' expression
              |     '-' expression
              |     '!' expression
              |     NEW type
              |     '(' expression ')'
              |     call
              |     lvalue
call    ::=   variable '(' argument_list ')'
lvalue   ::=   variable
         |     expression '[' expression ']'
constant   ::=   TRUE_CONST
           |     FALSE_CONST
           |     INT_CONST
           |     FLOAT_CONST
           |     STRING_CONST
type   ::=   INT
       |     FLOAT
       |     BOOL
       |     VOID
       |     type ARRAY index_list
index_list   ::=   index_list '[' index ']'
             |     '[' index ']'
index   ::=   INT_CONST
variable   ::=   IDENTIFIER
```