Especificação dos tokens - Linguagem L−−

Lucas Montenegro e Letícia Medeiros

05/04/2021

Universidade Federal de Alagoas - Instituto de Computação
Compiladores
Professor Alcino Dall lgna Junior

Conteúdo

1 Especificações dos tokens da linguagem

1.1 Linguagem utilizada

Para o desenvolvimento do analisador léxico foi utilizada a linguagem Java.

1.2 Lista de Tokens

public enum TokenClass {
UNKNOWN, MAIN, ID, TYPEVOID, TYPEINT, TYPEFLOAT, TYPECHAR, TYPEBOOL, TYPESTRING, TYPEARRAY, NULL, BREAK, CONSTINT, CONSTFLOAT, CONSTBOOL, CONSTCHAR, CONSTSTRING, OPTADD, OPTPOW, OPTMULT, OPTDIV, OPTSUB, OPTMOD, OPTCONCAT, ATRIB, OPTLESS, OPTLESSEQ, OPTGREAT, OPTGREATEQ, OPTEQ, OPTNOTEQ, OPTAND, OPTOR, OPTNOT, CONDIF, CONDELSIF, CONDELSE, LOOPWHILE, LOOPFOR, READ, WRITE, SEPARATOR, PARAMINIT, PARAMEND, ARRINIT, ARREND, COMMENT, FUNCDEF, FUNCRETURN, FUNCINIT, FUNCEND, SENTENEND, ENDFILE
}

1.2.1 Expressões Regulares Auxiliares

Segue abaixo as expressões regulares auxiliares:

- letra: [a - zA- Z]
- dígito: [0 - 9]
- símbolo: [‘ ’ | ‘ ‘ | ‘ ; ’ | ‘ , ’ | ‘ . ’ | ‘ : ’ | ‘ ? ’ | ’~’ | ‘ ! ’ | ‘ + ’ | ‘ - ’ | ‘ ∗ ’ | ‘ \ \ ’ | ‘ / ’ | ‘ _ ’ | ‘ % ’ | ‘ @ ’ | ‘& ’ | ‘ \# ’ |’# ’ | ‘ $ ’ | ‘ < ’ |‘ > ’ | ‘ = ’ | ‘ (‘ | ‘) ’ | ‘ [‘ | ‘] ’ | ‘ { ‘ | ‘} ’ | ‘ | ’ | ‘ \’ ’ | ‘ \“ ‘ | ‘ ∧ ’| ’ \n’];

1.3 Expressões Terminais

1.3.1 Main

MAIN: ‘main’

1.3.2 Identificador

ID: (’{letra}’) ((’{letra}’ | ’{dígito}’ | ’_’)∗)

1.3.3 Comentários

COMMENT: ‘$’

1.3.4 Delimitadores

Terminador

SENTENCEEND: ‘;’
ENDFILE: ‘< <EOF> >’

Separador

SEPARATOR: ‘,’

Escopos

FUNCINIT: ‘{‘
FUNCEND: ‘}’

Parâmetros

PARAMINIT: ‘(‘
PARAMEND: ‘)‘

Array

ARRINIT: ‘[‘
ARREND: ‘]’

1.3.5 Atribuição

ATRIB: ’=’

1.3.6 Função

FUNCDEF: ‘function’
FUNCRETURN: ‘return’

1.3.7 Leitura e Escrita

READ: ’read’
WRITE: ‘write’

1.3.8 Palavras Reservadas de estruturas condicionais e de repetição

CONDIF: ‘if’
CONDELSIF: ‘elsif’
CONDELSE: ‘else’
LOOPFOR: ‘for’
LOOPWHILE: ‘while’
BREAK: ’break’

1.3.9 Operadores

Lógicos

OPTAND: ‘&&’
OPTNOT: ’!’
OPTOR: ‘||’

Relacionais

OPTLESS: ‘<’
OPTLESSEQ: ‘<=’
OPTGREAT: ‘>’
OPTGRATEQ: ‘>=’
OPTEQ: ‘==’
OPTNOTEQ: ‘!=‘

Aritméticos

OPTADD: ‘+’
OPTSUB: ‘-‘
OPTPOW: ‘∧‘
OPTMULT: ‘∗’
OPTDIV: ‘/‘
OPTMOD: ‘%’

1.3.10 Tipos

TYPEINT: ‘int’
TYPEFLOAT: ‘float’
TYPECHAR: ‘char’
TYPEBOOL: ‘bool’
TYPESTRING: ‘string’;
TYPEARRAY: ‘array’;
TYPEVOID: ‘void’;

1.3.11 Constantes

CONSTINT: ’{dígito}’+
CONSTFLOAT: (’{dígito}’)+\.(’{dígito}’)+
CONSTBOOL: ($'true'$ | $'false'$)
CONSTCHAR: (‘ \’ ‘) ((’{letra}’ | ’{símbolo}’ | ’{dígito}’) ’(∧’#”)) (‘ \’ ‘)
CONSTSTRING: (’\“’) ((’{letra}’ | ’{símbolo}’ | ’{dígito}’)∗) (’\”’)
NULL: ’null’

1.3.12 Token Não reconhecido

UNKNOWN: Token inválido.