

Especificação dos tokens - Linguagem L--

Lucas Montenegro e Letícia Medeiros

20/03/2021

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

Conteúdo

1	Especificações dos tokens da linguagem	3
1.1	Linguagem utilizada	3
1.2	Categorias de Tokens	3
1.3	Lista de Tokens	3
1.3.1	Expressões Regulares Auxiliares	3
1.4	Expressões Terminais	3
1.4.1	Main	3
1.4.2	Identificador	3
1.4.3	Comentários	4
1.4.4	Delimitadores	4
1.4.5	Palavras Reservadas de estruturas condicionais e de repetição	4
1.4.6	Operadores	4
1.4.7	Tipos	5
1.4.8	Constantes	5

1 Especificações dos tokens da linguagem

1.1 Linguagem utilizada

A linguagem utilizada para o desenvolvimento do analisador léxico foi o Java.

1.2 Categorias de Tokens

1. Identificador
2. Palavras Reservadas
3. Operadores
4. Símbolos especiais

1.3 Lista de Tokens

Segue abaixo a lista de tokens da linguagem definida:

unknown, main, id, typeInt, typeFloat, typeChar, typeBool, typeString, constInt, constFloat, constBool, constChar, constString, optAdd, optPow, optMult, optDiv, optSub, optModu, optAttrib, optLess, optLessEqu, optGrea, optGreaEqu, optEqu, optNotEqu, optAnd, optOr, optNot, optConcat, condIf, condElsif, condElse, loopWhile, loopFor, read, write, separator, paramInit, paramEnd, arrInit, arrEnd, comment, funcDef, funcReturn, funcInit, funcEnd, sentenEnd, endFile.

1.3.1 Expressões Regulares Auxiliares

Segue abaixo as expressões regulares auxiliares:

- Letra: [a - z, A- Z]
- Número: [0 - 9]
- Símbolo: ‘ ‘ \ ; ‘ ‘ \ , ‘ ‘ . ‘ ‘ : ‘ ‘ ? ‘ ‘ ! ‘ ‘ + ‘ ‘ - ‘ ‘ * ‘ ‘ \ ‘ ‘ / ‘ ‘ _ ‘ ‘ \ % ‘ ‘ @ ‘ ‘ \ & ‘ ‘ \ # ‘ ‘ \ \$ ‘ ‘ \ < ‘ ‘ \ > ‘ ‘ \ = ‘ ‘ \ (‘ ‘ \) ‘ ‘ \ [‘ ‘ \] ‘ ‘ \ { ‘ ‘ \ } ‘ ‘ \ | ‘ ‘ \ ’ ‘ ‘ \ “ ‘ ‘ \ ^ ‘ ‘ \ \ ;

1.4 Expressões Terminais

1.4.1 Main

main: ‘main’

1.4.2 Identificador

id: ({letra}) (({letra} | {número} | ‘_’)*)

1.4.3 Comentários

comment: '\$'

1.4.4 Delimitadores

Terminador

sentenEnd: ';'

endFile: '< <EOF> >'

Separador

separator: ','

Escopos

funcInit: '{'

funcEnd: '}'

Parâmetros

paramInit: '('

paramEnd: ')'

Array

arrInit: '['

arrEnd: ']'

Função

funcDef: 'function'

funcReturn: 'return'

Leitura e Escrita

read: 'read'

write: 'write'

1.4.5 Palavras Reservadas de estruturas condicionais e de repetição

condIf: 'if'

condElsif: 'elsif'

condElse: 'else'

loopFor: 'for'

loopWhile: 'while'

1.4.6 Operadores

Lógicos

optAnd: '&&'

optNot: '!'

optOr: '||'

Relacionais

optLess: '<'
optLessEqu: '<='
optGrea: '>'
optGreatEqu: '>='
optEqu: '=='
optNotEqu: '!='

Aritméticos

optAdd: '+'
optSub: '-'
optPow: '^'
optMult: '*'
optDiv: '/'
optMod: '%'
optAttrib: '='

1.4.7 Tipos

typeInt: 'int'
typeFloat: 'float'
typeChar: 'char'
typeBool: 'bool'
typeString: 'string';

1.4.8 Constantes

constInt: {Número}+
constFloat: ({Número})+\.({Número})+
constBool: ('true' | 'false')
constChar: ('\'') ({Letra} | {Símbolo} | {Número}) ('\'')
constString: ('\"') (((Letra} | {Símbolo} | {Número})* ('\"'))