

Universidade Federal de Alagoas
Instituto de Computação
Curso de Ciência da Computação

Jvlal
Tabela de Símbolos - Tokens

João Victor Ribeiro Ferro
Lucas Albuquerque Lisboa

Maceió
2020.1

Linguagem para desenvolvimento	3
Enumeração com as categorias de Tokens	3
Expressões Regulares Auxiliares	4
Expressões Regulares dos Lexemas	4

Linguagem para desenvolvimento

A linguagem jvllal foi desenvolvida na linguagem de programação Java os seus analisadores léxicos e sintáticos. Sendo utilizado a versão Java Development Kit na versão 11.0.10

Enumeração com as categorias de Tokens

```
public enum Lexeme {  
    EOF,  
    ID,  
    CT_VAR,  
    CT_FLOAT,  
    CT_INT,  
    CT_CHAR,  
    CT_STRING,  
    CT_BOOL,  
    ON_BRACKET,  
    OFF_BRACKET,  
    OP_ATR,  
    OP_REL,  
    OP_RELNOT,  
    OP_AD,  
    OP_SUB,  
    OP_MULT,  
    OP_DIV,  
    OP_MOD,  
    OP_GREATER,  
    OP_LESS,  
    OP_GRTEREQ,  
    OP_LESSEQ,  
    OP_NOT,  
    OP_AND,  
    OP_OR,  
    OP_CONC,  
    RW_FUN,  
    RW_RETURN,  
    RW_IF,  
    RW_ELSE,  
    RW_WHILE,  
    RW_FOR,  
}
```

RW_INT,
RW_FLOAT,
RW_STRING,
RW_TOSTRING,
RW_CHAR,
RW_BOOL,
RW_READ,
RW_PRINT,
RW_PRINTLN,
RW_TRUE,
RW_FALSE,
RW_NULL,
RW_LENGTH,
ON_PAR,
OFF_PAR,
ON_BRACE,
OFF_BRACE,
QUOTE,
COLON,
APOSTROPHE,
RW_PROC,
RW_MAIN,
SEMICOLON,
RW_VOID,
UN_ID,
UN_CHAR,
UN_SYMBOL,
UN_OP,
UN_NUMBER,
SEP;}

Expressões Regulares Auxiliares

digit: [0-9] => Reconhece um dígito

letter: [a-zA-Z] => Reconhece uma letra (comum = sem acentos)

* => 0 (nenhuma) ou mais ocorrências

Expressões Regulares dos Lexemas

Identificadores

ID	(letter)(digit letter)*
-----------	----------------------------------

Palavras reservadas	
RW_MAIN	'main'
RW_PROC	'proc'
RW_FUN	'fun'
PR_IF	'if'
PR_ELSE	'else'
PR_WHILE	'while'
PR_FOR	'for'
RW_FLOAT	'float'
RW_INT	'int'
RW_CHAR	'char'
RW_STRING	'string'
RW_READ	'read'
RW_PRINTLN	'println'
RW_PRINT	'print'
RW_BOOL	'bool'
RW_TRUE	'True'
RW_FALSE	'False'
RW_TOSTRING	'toString'
RW_VOID	'void'
RW_RETURN	'return'
RW_NULL	'Null'
RW LENGHT	'length'

Operadores	
OP_ATR	'='
OP_REL	'=='
OP_RELNOT	'!='
OP_AD	'+'
OP_SUB	'-'
OP_MULT	'*'
OP_DIV	'/'
OP_MOD	'%'
OP_GREATER	'>'
OP_LESS	'<'
OP_GRTEREQ	'>='
OP_LESSEQ	'<='
OP_NOT	'!'
OP_AND	'&'
OP_OR	' '
OP_CONC	'#'

Delimitadores	
ON_PAR	‘(’
OFF_PAR	‘)’
ON_BRACKET	‘[’
OFF_BRACKET	‘]’
ON_BRACES	‘{’
OFF_BRACES	‘}’
TERMINAL	‘.’ ,’

COLON	‘.’
SEP	‘,’

Constante Literal	
CT_FLOAT	(digit)*(.) (digit)*
CT_INT	(digit)*
CT_STRING	(“)(letter)*(”)
CT_CHAR	(‘)(letter)(’)
CT_BOOL	‘True’ ‘False’

Erros Léxicos	
UN_SYMBOL	Símbolo não reconhecido
UN_ID	Identificador mal formado
UN_CHAR	Char mal formado
UN_OP	Operador mal formado
UN_NUMBER	Número mal formado