

UNIVERSIDADE FEDERAL DE ALAGOAS

INSTITUTO DE COMPUTAÇÃO

COMPILADORES 2021.1

Tokens da Linguagem JORG

Gabriel Luiz Leite Souza

João Victor Falcão Santos Lima

Rodrigo Santos da Silva

Maceió

2021

Linguagem para desenvolvimento

A linguagem de programação Java foi escolhida para o desenvolvimento do analisador léxico (e será utilizada futuramente para o sintático) da linguagem JORG, utilizando a versão 11.0.12 da JDK (Java Development Kit). A escolha foi baseada nas funcionalidades que a linguagem oferece nativamente, além da familiaridade de seu uso pelos desenvolvedores do trabalho.

Enumeração com as categorias dos Tokens

```
public enum TokenCategory {  
    // Identificador  
    ID(1),  
  
    // Operadores  
    OP_ADD(2),  
    OP_SUB(3),  
    OP_MULT(4),  
    OP_DIV(5),  
    OP_POT(6),  
    OP_MOD(7),  
    OP_NOT(8),  
    OP_OR(9),  
    OP_AND(10),  
    OP_MAIOR(11),  
    OP_MENOR(12),  
    OP_IGUAL(13),  
    OP_MAIOR_IG(14),  
    OP_MENOR_IG(15),  
    OP_N_IGUAL(16),  
    OP_ATR(17),  
    OP_CONCAT(18),  
    OP_SIZE(19),  
  
    // Delimitadores  
    ABRE_PAR(20),  
    FECHA_PAR(21),  
    ABRE_COL(22),  
    FECHA_COL(23),  
    ABRE_CHAVE(24),  
    FECHA_CHAVE(25),  
    TERMINAL(26),  
}
```

```
SEP(27),  
ASPAS(28),  
EOF_TOKEN(29),
```

```
// Constantes Literais  
CONST_INT(30),  
CONST_FLOAT(31),  
CONST_CHAR(32),  
CONST_STRING(33),  
CONST_BOOL(34),
```

```
// Palavras Reservadas  
PR_FUNCTION(35),  
PR_MAIN(36),  
PR_RETURN(37),  
PR_INPUT(38),  
PR_WRITE(39),  
PR_Writeln(40),  
PR_IF(41),  
PR_ELSE(42),  
PR_FOR(43),  
PR_WHILE(44),  
PR_BREAK(45),  
PR_CONST(46),  
TIPO_INT(47),  
TIPO_FLOAT(48),  
TIPO_CHAR(49),  
TIPO_STRING(50),  
TIPO_BOOL(51),  
TIPO_VOID(52),
```

```
// Erros Léxicos  
BAD_TOKEN(53);
```

```
private final int numericCode;
```

```
private TokenCategory(int numericCode) {  
    this.numericCode = numericCode;  
}
```

```
public int getNumericCode() {  
    return this.numericCode;  
}
```

```
}
```

Expressões Regulares dos Lexemas

Identificador		
ID	'(letter)(letter digit _)*'	1
Operadores		
OP_ADD	'+'	2
OP_SUB	'-'	3
OP_MULT	'*'	4
OP_DIV	'/'	5
OP_POT	'^'	6
OP_MOD	'%'	7
OP_NOT	'!'	8
OP_OR	' '	9
OP_AND	'&'	10
OP_MAIOR	'>'	11
OP_MENOR	'<'	12
OP_IGUAL	'"=="'	13
OP_MAIOR_IG	'">="'	14
OP_MENOR_IG	'"<="'	15
OP_N_IGUAL	'"!="'	16
OP_ATR	'='	17
OP_CONCAT	':'	18
OP_SIZE	'"size"'	19
Delimitadores		

ABRE_PAR	' ('	20
FECHA_PAR	')'	21
ABRE_COL	' ['	22
FECHA_COL	']'	23
ABRE_CHAVE	' {'	24
FECHA_CHAVE	' }'	25
TERMINAL	' ;'	26
SEP	' ,'	27
ASPAS	' ''	28
EOF_TOKEN	EOF	29
Constantes Literais		
CONST_INT	' (digit)*'	30
CONST_FLOAT	' (digit)+(.) (digit)+'	31
CONST_CHAR	' (')(letter)(')'	32
CONST_STRING	' (")(.)*(")'	33
CONST_BOOL	' ("false" "true")'	34
Palavras Reservadas		
PR_FUNCTION	' "function"'	35
PR_MAIN	' "main"'	36
PR_RETURN	' "return"'	37
PR_INPUT	' "input"'	38
PR_WRITE	' "write"'	39
PR_WRITELN	' "writeln"'	40
PR_IF	' "if"'	41
PR_ELSE	' "else"'	42

PR_FOR	' "for" '	43
PR_WHILE	' "while" '	44
PR_BREAK	' "break" '	45
PR_CONST	' "const" '	46
TIPO_INT	' "int" '	47
TIPO_FLOAT	' "float" '	48
TIPO_CHAR	' "char" '	49
TIPO_STRING	' "string" '	50
TIPO_BOOL	' "bool" '	51
TIPO_VOID	' "void" '	52
Erros Léxicos		
BAD_TOKEN	Símbolo não reconhecido	53

Expressões Regulares Auxiliares	
digit	' [0-9] '
letter	' [a-zA-Z] '