

Especificação Tokens: SPL

Compiladores 2017.1

José Lucas Leite Calheiros

Agosto - 2017

Contents

1	Lingagem de implementação	2
2	Enumeração de Tokens	2
3	Especificação de expressões regulares	2
3.1	Expressões Auxiliares	2
3.2	Expressões Terminais	3

1 Linguagem de implementação

Os analisadores léxios e sintáticos serão implementados na linguagem de programação **Java**

2 Enumeração de Tokens

```
public enum Tokens {  
    tkMain ,  
    tkId ,  
    tkComent ,  
    tkTerm ,  
    sepVir ,  
    sepPon ,  
    escFunc ,  
    escDecl ,  
    escEnd ,  
    blkBegin ,  
    blkEnd ,  
    parBegin ,  
    parEnd ,  
    arrBegin ,  
    arrEnd ,  
    tInt ,  
    tFloat ,  
    tChar ,  
    tBool ,  
    tString ,  
    tArray ,  
    eIf ,  
    eElse ,  
    eFor ,  
    eWhile ,  
    cInt ,  
    cFloat ,  
    cChar ,  
    cBool ,  
    cString ,  
    opMenUn ,  
    opAdi ,  
    opMul ,  
    opRes ,  
    opComp ,  
    opIgua ,  
    opNeg ,  
    opConj ,  
    opDisj ,  
    opConc  
}
```

3 Especificação de expressões regulares

3.1 Expressões Auxiliares

letra = 'a' | 'b' | ... | 'Y' | 'Z' ;
digito = '0' | '1' | ... | '8' | '9' ;

```

simbolo = ' ' | '.' | ',' | ':' | ';' | '?' | '!' | '+' | '-' | '*' | '\ ' | '/' | '_' | '%' | '@' | '&' | '#' | '$' |
'<' | '>' | '=' | '(' | ')' | '[' | ']' | '{' | '}' | '|' | '\'' | '\"' ;

```

3.2 Expressões Terminais

Main:

```
tkMain = 'main';
```

Identificador:

```
tkId = 'letter'('letter' | 'digit')* ;
```

Comentários:

```
tkComent = '//';
```

Terminador:

```
tkTerm = ';' ;
```

Separador:

```
sepVir = ',' ;
```

```
sepPon = ':' ;
```

Escopos:

```
escFunc = 'func' ;
```

```
escDecl = 'decl' ;
```

```
escEnd = 'end' ;
```

```
blkBegin = '{' ;
```

```
blkEnd = '}' ;
```

Parametros:

```
parBegin = '(' ;
```

```
parEnd = ')' ;
```

Array:

```
arrBegin = '[' ;
```

```
arrEnd = ']' ;
```

Tipos:

```
tInt = 'int' ;
```

```
tFloat = 'float' ;
```

```
tChar = 'char' ;
```

```
tBool = 'bool' ;
```

```
tString = 'string' ;
```

```
tArray = 'array' ;
```

Estruturas:

```
eIf = 'if' ;
```

```
eElse = 'else' ;
```

```
eFor = 'for' ;
```

```
eWhile = 'while' ;
```

Constantes:

```
cInt = (digito)+ ;
```

```
cFloat = (digito)+.(digito)+ ;
```

```
cChar = (')(digito)|(letra)|(simbolo)(') ;
```

```
cBool = (true)|(false) ;
```

```
cString = (")((digito)|(letra)|(simbolo))*(") ;
```

Operadores:

```
opMenUn = '~';  
opAdi = '+'|'-';  
opMul = '*'|'/';  
opRes = '%';  
opComp = '<'|>|'<='|'>='';  
opIgua = '=='|'!='';  
opNeg = '!';  
opConj = '&';  
opDisj = '|';  
opConc = '#';
```