

## Analizador Léxico

### 1. OBJETIVO

Implementar um analisador léxico para a linguagem definida a seguir:

#### Características da linguagem

1. Suporta os tipos: integer, real, char, string
2. Comandos:
  - a. Atribuição com o operador "[:="
  - b. Entrada com o comando read( )
  - c. Saída com os comandos write( ) e writeln( )
  - d. Condicional com o comando if - then - else
  - e. Repetições com os comandos: while - do, repeat - until, for - to - do
3. Constantes caracteres delimitados por ( ' ' ) e constantes strings por ( "" )
4. Operadores relacionais: "=", ">=", "<=", ">", "<", "<>"
5. Operadores lógicos: and, or, not
6. Operadores aritméticos: "+", "-", "\*", "/" e mod
7. Símbolos especiais: ",", ":", ";", "."
8. Bloco de comandos delimitados por begin e end
9. Comentário de bloco com os delimitadores "/\*" e "\*/"
10. Lista de Palavras Reservadas:

ABSOLUTE ARRAY BEGIN CASE CHAR CONST DIV DO DOWTO ELSE END  
EXTERNAL FILE FOR FORWARD FUNC FUNCTION GOTO IF IMPLEMENTATION  
IN INTEGER INTERFACE INTERRUPT LABEL MAIN NIL NIT OF PACKED PROC  
PROGRAM REAL RECORD REPEAT SET SHL SHR STRING THEN TO TYPE UNIT  
UNTIL USES VAR WHILE WITH XOR

### 2. IMPLEMENTAÇÃO

Poderá ser utilizado qualquer ferramenta geradora de analisador léxico. Deverão ser identificados os seguintes **Símbolos da Linguagem**:

- Palavra Reservada
- Identificador (Ex: x, variavel, i, var2)
- Número Inteiro (Ex: 1, 13)
- Número Real (Ex: 1.33, 24.40e-04)
- Operador aritmético (“+”, “-”, “\*”, “/”, mod)
- Operador relacional ( = >= > < <= <> )
- Operador lógico (and, or, not)
- Símbolo Especial ( = ( ) , ; : )
- Atribuição (:=)
- Fim (.)

Deverão ser desconsiderados os caracteres não significativos: espaços, tabs, enters e comentários de bloco (/\* qualquer coisa \*/).

É preciso tratar os comentários de maneira explícita, ou seja, não podem aparecer na saída do programa porque não são tokens.

Qualquer outro símbolo deverá ser considerado desconhecido.

O trabalho será testado da seguinte maneira:

./executavel entrada.txt

Onde:

- entrada.txt: é o arquivo que conterá o programa na linguagem para ser analisado.

Um exemplo de um arquivo de entrada (entrada1.txt):

```
program teste;  
var x,y: integer;  
const pi :=3.1416;  
/* inicio do programa */  
begin  
read(x);
```

```
if (x > y) then
y := x ;
else
y := -x;
writeln(x);
write(y);
end.
```

Executando ./trab1 entrada1.txt, teremos como resposta em stdout (terminal):

```
program PALAVRARESERVADA
teste IDENTIFICADOR
; SIMBOLOESPECIAL
var PALAVRARESERVADA
x IDENTIFICADOR
, SIMBOLOESPECIAL
y IDENTIFICADOR
: SIMBOLOESPECIAL
integer PALAVRARESERVADA
; SIMBOLOESPECIAL
const PALAVRARESERVADA
pi IDENTIFICADOR
:= ATRIBUIÇÃO
3.1416 NUMEROREAL
; SIMBOLOESPECIAL
begin PALAVRARESERVADA
read IDENTIFICADOR
( SIMBOLOESPECIAL
x IDENTIFICADOR
) SIMBOLOESPECIAL
; SIMBOLOESPECIAL
if PALAVRARESERVADA
```

```
( SIMBOLOESPECIAL
x IDENTIFICADOR
) SIMBOLOESPECIAL
then PALAVRARESERVADA
y IDENTIFICADOR
:= ATRIBUICAO
x IDENTIFICADOR
else PALAVRARESERVADA
y IDENTIFICADOR
:= ATRIBUICAO
- OPERADOR
x IDENTIFICADOR
; SIMBOLOESPECIAL
writeln IDENTIFICADOR
( SIMBOLOESPECIAL
x IDENTIFICADOR
) SIMBOLOESPECIAL
; SIMBOLOESPECIAL
write IDENTIFICADOR
( SIMBOLOESPECIAL
y IDENTIFICADOR
) SIMBOLOESPECIAL
; SIMBOLOESPECIAL
end PALAVRARESERVADA
. FIM
```

```
/* Outro exemplo de programa */
Program Piloto;
/* declarações de variáveis e constantes globais */
var cont, total: integer ;
Nota1, Nota2, Media_das_medias, med: real;
/* Inicio do Programa */
```

```
begin
media_das_medias := 0;
writeln("***** ENTRADA DE DADOS *****");
writeln("Digite o total de alunos");
read(total);
for cont=1 to total do
begin
writeln("Digite os valores da primeira nota do aluno ", cont);
read(Nota1);
writeln("Digite os valores da segunda nota do aluno ", cont);
read(Nota2);
med := (Nota1+Nota2)/2.0;
media_das_medias := media_das_medias + med;
write("Media = ",med);
end;
write("Media Geral = ",Media_das_medias/total);
end.
```

### 3. CONSIDERAÇÕES IMPORTANTES

- O programa deverá ser apresentado em aula juntamente com o código fonte.

### 4. AVALIAÇÃO

- Terão nota zero os trabalhos que se enquadrarem em uma ou mais situações abaixo:
  - Plágio;
  - Programa não compila;
  - Não está de acordo com as especificações.