



Universidade do Vale do Itajaí
Campus KobraSol

Trabalho Analisador Sintático

Compiladores

Katya Balvedi e Pedro André

Professor: Alessandro Mueller

Estrutura de comentário

Comentário de linha

// Qualquer símbolo/palavra

Comentário de bloco

/* qualquer símbolo/palavra */

Tabela de símbolos terminais

CÓDIGO	TOKEN	Tipo	Descrição
14	DO	Palavra-chave	do
15	MAKE	Palavra-chave	make
16	END	Palavra-chave	end
17	CONST	Palavra-chave	const
18	VAR	Palavra-chave	var
19	INT	Palavra-chave	int
20	REAL	Palavra-chave	real
21	CHAR	Palavra-chave	char
22	BOOL	Palavra-chave	bool
23	GET	Palavra-chave	ger
24	PUT	Palavra-chave	put
25	IF	Palavra-chave	if
26	THEN	Palavra-chave	then
27	ELSE	Palavra-chave	else
28	TRUE	Literal	true
29	FALSE	Literal	false
30	WHILE	Palavra-chave	while

31	ABRE_PARENTESES	Símbolo	(
32	FECHA_PARENTESES	Símbolo)
33	PONTO	Símbolo	.
34	VIRGULA	Símbolo	,
35	DOISPONTOS	Símbolo	:
36	ATRIBUICAO	Operador	->
37	PONTOVIRGULA	Símbolo	;
38	ADICAO	Operador	+
39	SUBTRACAO	Operador	-
40	MULTIPLICACAO	Operador	*
41	DIVISAO	Operador	/
42	POTENCIA	Operador	**
43	DIVISAO_INTEIRA	Operador	%%
44	RESTO	Operador	%
45	IGUAL	Operador	==
46	DIFERENTE	Operador	!=
47	MENOR	Operador	<
48	MAIOR	Operador	>
49	MENOR_IGUAL	Operador	<=
50	MAIOR_IGUAL	Operador	>=
51	E	Operador lógico	&
52	OU	Operador lógico	
53	NAO	Operador lógico	!
54	DIGITO	Categoria	Dígito (0-9)
55	CONSTANTE_INTEIRA	Literal	uma constante inteira

56	CONSTANTE_REAL	Literal	uma constante real
57	CONSTANTE_LITERAL	Literal	' ' ou ""
58	LETRA	Categoria	Letra (a-z, A-Z)
59	IDENTIFICADOR	Categoria	um identificador
60	SIMBOLO_INVALIDO	Inválido	Simbolos fora da gramática
61	CONSTANTE_INTEIRA_INVALIDA	Inválido	Constante incorreta de acordo com a gramática
62	CONSTANTE_REAL_INVALIDA	Inválido	constante real incorreta de acordo com a gramática
63	CONSTANTE_LITERAL_INVALIDA	Inválido	constante literal incorreta de acordo com a gramática
64	IDENTIFICADOR_INVALIDO	Inválido	identificador incorreto de acordo com a gramática

Mensagem de erros léxicos

(código do erro e descrição do erro)

Código do erro	Descrição
60	Símbolo invalido
61	Constante inteira inválida
62	Constante real inválida
63	Constante literal inválida
64	Identificador Invalido

Mensagens

|Erro: 60 - Símbolo invalido, linha: 1- coluna: 2|

|Erro: 61 - Constante inteira invalida, linha: 1- coluna: 2|

|Erro: 62 - Constante real invalida, linha: 1- coluna: 2|

|Erro: 63 - Constante literal invalida, linha: 1- coluna: 2|

|Erro: 64 - Identificador Invalido, linha: 1- coluna: 2|

Mensagem de erros Sintáticos

- Esperado(s): (exibição do que era esperado)

GLC da linguagem 2024.2 - Notação BNF

```
<programa> ::= make identificador  
              <declaracao de constantes ou variaveis>  
              <lista de comandos>  
              end.
```

```
<declaracao de constantes ou variaveis> ::= <declaracao de constante>  
                                           | <declaracao de variavel>  
                                           | <declaracao de constantes ou variaveis>  
                                           | £
```

```
<declaracao de constante> ::= const <constantes> end;
```

```
<constantes> ::= <tipo> : <lista de identificadores> = <valor> . <constantes'>
```

```
<valor> ::= constante inteira  
          | constante real  
          | constante literal
```

```
<tipo> ::= int  
         | real  
         | char
```

```
<lista de identificadores> ::= identificador <lista de identificadores'>
```

```
<lista de identificadores'> ::= , identificador <lista de identificadores'>  
                               | £
```

```
<constantes'> ::= <constantes>  
                 | £
```

```
<declaracao de variavel> ::= var <variaveis> end ;
```

```
<variaveis> ::= <tipo'> : <lista de identificadores> . <variavel>
```

```
<variavel> ::= <variaveis>  
             | £
```

```
<tipo'> ::= <tipo>  
          | bool
```

```
<atribuicao> ::= <expressao> -> identificador .
```

<expressão> ::= <expressão aritmética ou lógica> <expressão'>

<expressão'> ::= = <expressão aritmética ou lógica>
| <> <expressão aritmética ou lógica>
| < <expressão aritmética ou lógica>
| > <expressão aritmética ou lógica>
| <= <expressão aritmética ou lógica>
| >= <expressão aritmética ou lógica>
| £

<expressão aritmética ou lógica> ::= <termo2> <menor prioridade>

<menor prioridade> ::= + <termo2> <menor prioridade>
| - <termo2> <menor prioridade>
| | <termo2> <menor prioridade>
| £

<termo2> ::= <termo1> <media prioridade>

<media prioridade> ::= * <termo1> <media prioridade>
| / <termo1> <media prioridade>
| % <termo1> <media prioridade>
| %% <termo1> <media prioridade>
| & <termo1> <media prioridade>
| £

<termo1> ::= <elemento> <maior prioridade>

<maior prioridade> ::= ** <elemento> <maior prioridade>
| £

<elemento> ::= identificador
| constante inteira
| constante real
| constante literal
| true
| false
| (<expressão>)
| ! (<expressão>)

<entrada> ::= get (<lista de identificadores>) .

<saida> ::= put (<lista de identificadores ou constantes>)

<lista de identificadores ou constantes> ::= <item> <lista de identificadores ou constantes'>

<lista de identificadores ou constantes'> ::= , <lista de identificadores ou constantes'>
| £

<item> ::= identificador

- | constante inteira
- | constante real
- | constante literal

<lista de comandos> ::= <comando> <lista de comandos'>

<lista de comandos'> ::= <lista de comandos>

| £

<comando> ::= <atribuicao>

| <entrada>

| <saida>

| <selecao>

| <repeticao>

<selecao> ::= if <expressao> then <lista de comandos> <resultado> end .

<resultado> ::= else <lista de comandos>

| £

<repeticao> ::= while <expressao> do <lista de comandos> end .