MiniAlg é uma linguagem *case sensitive*.

**Exemplo de código correto**

(Precisa ser ajustado de acordo com a gramática)

**programa** correto

**inteiro** a , b , c ;

**booleano** d , e , f ;

**procedimento** proc ( **inteiro** a1 )

**inteiro** a , b , c ;

**booleano** d , e , f ;

a = 1 ;

**se** ( a < 1 ) **entao**

a = 12 ;

**fimse**

**fimprocedimento**

a = 2 ;

b = 10 ;

c = 11 ;

a = b + c ;

d = **verdadeiro** ;

e = **falso** ;

f = **verdadeiro** ;

**escreva** ( b ) ;

**se** ( d ) **entao**

a = 20 ;

b = 10 **mul** c ;

c = a **div** b ;

**fimse**

**enquanto** ( a > 1 ) **faca**

**se** ( b > 10 ) **entao**

b = 2 ;

**senão**

a = a - 1 ;

**fimse**

**fimenquanto**

**fimprograma**

Um programa MiniAlg deverá estar codificado em um único arquivo fonte, sem fazer referências a detalhes externos a ele.

**ESPECIFICAÇÃO LÉXICA**

* **Comentários**

Aparecem delimitados por “/\*” e “\*/”. Tudo que segue os símbolos“/\*” é ignorado pelo MiniAlg até que se encontre “\*/”.

/\* comentário \*/

* **Identificadores (variáveis)**

Podem ser criados somente com letras de **a** até **z** (maiúsculas ou minúsculas).

Value

companyNumber

12Val /\* inválido \*/

Val12 /\* inválido \*/

Ex$ /\* inválido \*/

company-name /\* inválido \*/

company\_name /\* inválido \*/

* **Tipos de dados**

Numerais : Inteiros (tipo inteiro)

Booleano : **falso** | **verdadeiro** (tipo booleano)

* **Operadores**

+ - div mul < > <> <= >= == =

* **Delimitadores**

**(** **)** , : ;

* **PalavrasReservadas**

**programa se escreva**

**procedimento fimse verdadeiro**

**fimprograma inteiro falso**

**fimprocedimento booleano faca**

**enquanto então**

**fimenquanto senao**

**SINTAXE**

1. **〈**programa**〉**::=

**programa 〈**identificador**〉**

**〈**bloco**〉**

**fimprograma**

2. **〈**bloco**〉** ::=

[ **〈**parte de declarações de variáveis**〉** ]

[ **〈**parte de declarações de funções**〉** ]

**〈**comando composto**〉**

**Declarações**

3. **〈**parte de declarações de variáveis**〉** ::=

**〈**declaração de variáveis**〉**

{ **〈**declaração de variáveis**〉** }

4. **〈**declaração de variáveis**〉**::= **〈**tipo**〉** : **〈**lista de identificadores**〉** ;

5. **〈**tipo**〉** ::= **inteiro** | **booleano**

**OBS**: Todos os identificadores devem ser declarados antes de serem utilizados.

6. **〈**lista de identificadores**〉** ::= **〈**identificador**〉** [**, 〈**identificador**〉** ]

7. **〈**parte de declarações de subrotinas**〉** ::=

{ **〈**declaração de procedimento**〉 ;** }

8. **〈**declaração de procedimento**〉** ::=

**procedimento 〈**identificador**〉** ( [ **〈**parâmetros formais**〉** ] ) **〈**bloco**〉**

**fimprocedimento**

9. **〈**parâmetros formais**〉** ::=  **〈**parâmetro formal**〉** { **, 〈**parâmetro formal**〉** }

10. **〈**parâmetro formal**〉** ::= **〈**tipo**〉 : 〈**identificador**〉**

**Comandos**

11. **〈**comando composto**〉** ::= **〈**comando**〉** ; [ **〈**comando**〉** ; ]

12. **〈**comando**〉** ::=

**〈**atribuição**〉**

| **〈**chamada de procedimento**〉**

| **〈**comando condicional**〉**

| **〈**comando repetitivo**〉**

| **escreva** ( **〈**identificador**〉** ) ;

13. **〈**atribuição**〉** ::= **〈**variável**〉 = 〈**expressão**〉** ;

14. **〈**chamada de procedimento**〉** ::=

**〈**identificador**〉** [ **( 〈**lista de parâmetros**〉** **)** ];

15. **〈**lista de parâmetros**〉** ::=

[ ( **〈**identificador**〉** | **〈**número**〉** | **〈**bool**〉** )

{, ( **〈**identificador**〉** | **〈**número**〉** | **〈**bool**〉** ) } ]

16. **〈**comando condicional**〉** ::=

**se ( 〈**expressão**〉 )** **entao**

**〈**comando composto**〉**

[ **senao 〈**comando composto**〉** ]

**fimse**

17. **〈**comando repetitivo**〉** ::=

**enquanto ( 〈**expressão**〉 )** **faca**

**〈**comando composto**〉**

**fimenquanto**

**Expressões**

18. **〈**expressão**〉** ::=

**〈**expressão simples**〉** [ **〈**relação**〉** **〈**expressão simples**〉** ]

19. **〈**relação**〉** ::= **==** | **<>** | **<** | **<=** | **>=** | **>**

20. **〈**expressão simples**〉** ::= [ **+** | **-** ] **〈**termo**〉** { [ **+** | **-** ] **〈**termo**〉** }

21. **〈**termo**〉** ::= **〈**fator**〉** { ( **mul** | **div** ) **〈**fator**〉** }

22. **〈**fator**〉** ::=

**〈**variavel**〉**

| **〈**número**〉**

| **〈**bool**〉**

| **( 〈**expressão simples**〉** **)**

23. **〈**variável**〉** ::= **〈**identificador**〉**

**Números e Identificadores**

24. **〈**bool**〉** ::= **verdadeiro** | **falso**

25. **〈**número**〉** ::= **〈**dígito**〉** { **〈**dígito**〉** }

26. **〈**dígito**〉** ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

27. **〈**identificador**〉** ::= **〈**letra**〉** { **〈**letra**〉** }

28. **〈**letra**〉** ::= a | b | ... | z | A | B | ... | Z

**EBNF:**

{α} = repetição da cadeia a zero ou mais vezes

α | β = α ou β devem ser escolhidos

Não terminais aparecem entre **〈** e **〉**.