

ESPECIFICAÇÃO DA LINGUAGEM 2022.1

Forma geral de um programa

```
do this identificador [ ]  
    <declaração de tipo enumerado>  
    <declaração de constantes e variáveis>  
    body [  
        <lista de comandos>  
    ]  
description comentário
```

- em <lista de comandos> deve existir no mínimo um comando;
- *description comentário* é facultativo e corresponde a um comentário (constante literal) acerca do programa.

Forma geral da declaração de tipo enumerado

```
declaration type [  
    identificador is <lista de identificadores> .  
]
```

Os tipos enumerados são definidos pelo programador.

- a <lista de identificadores> deve se referir a identificadores de constantes;
- em <lista de identificadores> deve existir no mínimo um e caso existam mais identificadores, os mesmos devem ser separados uns dos outros uma vírgula (,);
- *identificador is <lista de identificadores>* . pode ocorrer uma ou mais vezes;
- a declaração de tipo enumerado pode não existir, caso não seja definido o tipo enumerado.

Forma geral da declaração de constantes e variáveis

```
declaration constant and variable [  
    as constant  
        <lista de identificadores> is <tipo> = <valor> .  
    as variable  
        <lista de identificadores> is <tipo> .  
]
```

- <lista de identificadores> **is** <tipo> = <valor> . e <lista de identificadores> **is** <tipo> . podem ocorrer uma ou mais vezes;
- <tipo> pode ser **integer**, **real**, **string**, **logic** ou um **tipo enumerado (no caso de variáveis)**;
- o <tipo> da lista de identificadores de constantes não poderá ser de um tipo enumerado;
- em <lista de identificadores> deve existir no mínimo um e caso existam mais identificadores, os mesmos devem ser separados uns dos outros por uma vírgula (,); No caso da declaração de variáveis, cada identificador poderá ou não ser seguido por { *constante numérica inteira* }, indicando uma variável indexada unidimensional cujos índices variam no intervalo de 1 até a constante numérica especificada;
- <valor> pode ser um valor inteiro, real ou literal, compatíveis com os tipos **integer**, **real** e **string**, respectivamente;
- a declaração de constantes é precedida de **as constant**;
- a declaração de variáveis é precedida de **as variable**;
- a declaração de constantes pode preceder a declaração de variáveis ou a declaração de variáveis pode preceder a declaração de constantes e isto ocorrerá apenas uma única vez;
- a declaração de constantes pode não existir, caso não seja utilizada nenhuma constante no programa;
- a declaração de variáveis pode não existir, caso não seja utilizada nenhuma variável no programa;
- a declaração de constantes e variáveis pode não existir, caso não seja utilizada nenhuma constante ou variável no programa.

Forma geral do comando de atribuição

```
designate this <lista de identificadores> as <expressão> .
```

- a <lista de identificadores> deve se referir a identificadores de variáveis;
- em <lista de identificadores> deve existir no mínimo um e caso existam mais identificadores de variáveis, os mesmos devem ser separados uns dos outros uma vírgula (,);

- <expressão> pode ser qualquer expressão aritmética, relacional ou lógica envolvendo identificadores e constantes do tipo **integer**, **real**, **string**, **logic** ou **um tipo enumerado (no caso de variáveis)**;
- o resultado da avaliação de <expressão> deve ser um valor do mesmo tipo (ou de tipo compatível) da lista de identificadores.

Forma geral do comando de entrada de dados

read this [<lista de identificadores>] .

- em <lista de identificadores> deve existir no mínimo um e, caso existam mais identificadores de variáveis, os mesmos devem ser separados uns dos outros por uma vírgula (,).

Forma geral do comando de saída de dados

write all this [<lista de identificadores e/ou constantes>] .

- em <lista de identificadores e/ou constantes> deve existir no mínimo um identificador de constante/variável ou uma constante (numérica ou literal) e, caso existam mais identificadores de constantes/variáveis e/ou constantes, os mesmos devem ser separados uns dos outros por uma vírgula (,).
- as constantes e os identificadores de constantes/variáveis e respectivos conteúdos serão apresentados no dispositivo padrão de saída.

write this [<lista de identificadores e/ou constantes>] .

- em <lista de identificadores e/ou constantes> deve existir no mínimo um identificador de constante/variável ou uma constante (numérica ou literal) e, caso existam mais identificadores de constantes/variáveis e/ou constantes, os mesmos devem ser separados uns dos outros por uma vírgula (,).
- as constantes e os conteúdos dos identificadores de constantes/variáveis serão apresentados no dispositivo padrão de saída.

Forma geral do comando de seleção

avaliare this <expressão>

true result [<lista de comandos>]

untrue result [<lista de comandos>] .

- < expressão> pode ser qualquer expressão relacional ou lógica envolvendo identificadores e constantes do tipo **integer**, **real**, **string**, **logic** ou **um tipo enumerado (no caso de variáveis)**;
- o resultado da avaliação de <expressão> deve ser um valor lógico (**true** ou **untrue**);
- os comandos da cláusula **true result** serão executados se o resultado da avaliação da expressão for **true**; caso contrário, serão executados os comandos da cláusula **untrue result**;
- as cláusulas **true result** e **untrue result** só podem ocorrer uma vez, em qualquer ordem, e são opcionais, mas deve existir no mínimo uma delas.

Forma geral do comando de repetição

repeat this <expressão>

[<lista de comandos>] .

- <expressão> pode ser qualquer expressão relacional ou lógica envolvendo identificadores e constantes do tipo **integer**, **real**, **string**, **logic** ou **um tipo enumerado (no caso de variáveis)**;
- o resultado da avaliação de <expressão> deve ser um valor lógico (**true** ou **untrue**);
- os comandos da estrutura serão repetidos sempre que o resultado da avaliação da expressão for **true**.

São operadores:

d) aritméticos: + - * / ** (potência) % (divisão inteira) %% (resto da divisão inteira)

e) relacionais: == (igual), != (diferente), << (menor), >> (maior), <=< (menor igual) e >=> (maior igual)

f) lógicos: & (e), | (ou) e ! (não)

Podem ser usados para agrupar as expressões aritméticas, relacionais ou lógicas os parênteses (e).

São constantes lógicas: **true** e **untrue**.

As palavras reservadas podem ser escritas com letras minúsculas e/ou maiúsculas.

Todos os comandos são finalizados com um ponto.