

Manual da linguagem Gomes

Apresentamos a seguir o funcionamento da linguagem de programação Gomes. Esta linguagem tem por objetivo a implementação de programas básicos, utilizando-se de dados do tipo double. Algumas considerações importantes:

- + Todas as linhas devem ser encerradas com “;”.
- + Linguagem “case sensitive”, diferindo maiúsculas e minúsculas.
- + Espaços não influenciam no funcionamento do programa, podendo ser utilizados conforme desejo do programador.
- + Estar atento às orientações e exemplos, evitando erros de digitação.
- + Utilização de variáveis não declaradas acarretará erro no funcionamento do programa, que não encontrará a variável em seu vetor de variáveis, e se encerrará.

Variáveis:

Declaração: As variáveis em Gomes são do tipo double. Essas devem ser declaradas antecedidas da palavra **Var**, dois pontos, o nome da variável a ser declarada, e ponto e vírgula, como nos exemplos seguintes:

```
Var:a;  
ou  
Var : b;
```

Atribuição: A atribuição segue o estilo da declaração. Tendo a variável a qual se atribuirá valor iniciando a linha, seguida da igualdade e o valor que será colocado nela. Podendo também conter, após a igualdade, uma operação entre dois números, duas variáveis ou entre um número e uma variável. Exemplos:

```
a = 1;  
b = a + 2;  
c = b - a;  
d = 4 * 3;
```

Atribuição com entrada do teclado: Para entrada de valores pelo teclado, deve-se utilizar a palavra **Digite**. Após ela, entre parênteses, colocar a variável na qual o valor deve ser inserido. Vide abaixo exemplo:

```
Var: a;  
Var: b;  
Imprime([Digite um valor: ]);  
Digite(a);  
Imprime([Digite um valor: ]);  
Digite(b);
```

Operações:

As operações seguem as seguintes regras: devem conter dois elementos, podendo estes ser número ou variável. Executam apenas multiplicação, divisão, adição e subtração. Exemplos, vide as três últimas atribuições da imagem acima.

Impressão:

A impressão na tela deve ser chamada pela palavra *Imprime* e, entre parênteses, os elementos a serem impressos. Estes podem ser variáveis ou strings, conforme exemplos abaixo. No primeiro, imprime apenas a variável. No segundo, imprime o texto definido pelo usuário entre colchetes, e a variável, definida após a vírgula. Por fim, tem a opção de imprimir apenas o texto, definido entre colchetes.

```
Imprime (a);  
Imprime ([texto],a);  
Imprime ([texto]);
```

SE:

O controle de fluxo da linguagem deve ser chamado da seguinte forma: utilizando-se a palavra *Inicio.se*, e entre parênteses a comparação a ser feita, podendo conter apenas a comparação entre dois. Estes podem ser um números ou variáveis. Em relação aos comparadores, podem ser utilizados a igualdade (==), sinais de maior ou menor (>, <), maior ou igual, menor ou igual (>=, <=) e diferente de (!=). Para finalizar o Se, utiliza-se a palavra *Fim.se*. Dentro do controle de fluxo, todas as funções já apresentadas da linguagem podem ser utilizadas. Temos ainda o *Senao.se*, que

somente será executado caso tenhamos “falso” na primeira condição. Para finalizar o comando se utiliza **Fim.senao** .

```
Inicio.se(a>b);  
    Imprime([Primeiro maior que o segundo]);  
Fim.se;  
Senao;  
    Imprime([Primeiro menor ou igual ao segundo]);  
Fim.senao;
```