

Universidade Federal da Fronteira Sul – UFFS  
Ciência da Computação  
Programação I – 2015.1  
Fernando Bevilacqua

Manual da Linguagem Dibre

Acadêmicos: Jackson Henrique Hochscheidt  
Maikiel Roos

Chapecó, 14 de maio de 2015

## Variáveis

A linguagem dibre trabalha apenas com o tipo de variável “varReal”, que aceita qualquer valor decimal, seja ela positivo ou negativo.

### Declaração de variáveis :

A declaração é feita da seguinte forma :

**[tipoDeVariavel] [nomeDaVariavel] ; // sem atribuição**

**ou**

**[tipoDeVariavel] [nomeDaVariavel] = [valor] ; //com atribuição de valor**

[tipoDeVariavel] é o tipo da variável, que na linguagem dibre é varReal. [nomeDaVariavel] é o nome da variável do tipo [tipoDeVariavel].

[valor] é um valor a ser atribuído a [nomeDaVariavel]. [valor] pode ser tanto um valor decimal, como o nome de outra variavel. OBS: Se [valor] for um número decimal, o mesmo deve ser escrito da seguinte forma: “-10.10”. Pois se escrever “-10,10”, ou, “-10.10”, ou “- 10,10”, não funcionará.

\* Só é possível declarar apenas uma variável por linha, e a declaração deve ser feita toda numa linha só.

\* o nome da variável deve começar com qualquer letra maiúscula ou minúscula, apenas. Caso comece com qualquer outro caractere diferente disto, não funcionará.

\* aceita infinitos espaços em branco durante a declaração.

\* entenderá como o fim da declaração de uma “;”.

### Atribuição de valor a variável:

A atribuição de valor a variável é feita da seguinte forma:

**[nomeDaVariavel] = [valor] [operador] [valor] ; // caso não seja feita na declaração da variável. OBS : Declaração com atribuição, segue as regras descritas acima, na declaração de variável.**

[nomeDaVariavel] e [valor], seguem as mesmas regras descritas na declaração.

[operando] é a operação a ser feita. Aceita “SOMA”, “SUBTRAI”, “MULTIPLICA”, “DIVIDE”.

\* atribuição será feita através do “=”. Ex: var1 = 10;

- \* pode ser feita junto com a declaração da variável. Ex: double var1 = 10;
- \* poderá se atribuir o resultado de uma expressão a uma variável, caso está atribuição não esteja sendo feita na declaração da variável.

Exemplos declaração que funcionam:

```
varReal    teste ;
varReal    teste2    =    -10.50    ;
varReal    teste3 = teste ;
teste = teste2 SOMA teste3;
```

Exemplos declaração que não funcionam:

```
varReal    1teste;
varReal    teste2    =    teste SOMA    teste2;
varReal    teste    =    - 10.10;
varReal teste = -10,10;
varReal teste = - 10,10;
```

## Operações Aritméticas

As operações soma, subtração, multiplicação e divisão, são reconhecidas na linguagem respectivamente como, SOMA, SUBTRAI, MULTIPLICA e DIVIDE.

Funciona da seguinte maneira:

[valor1] [operador] [valor2]

OBS: uma operação deve estar sempre associada a uma atribuição à uma variável, ou em um teste de condição. Como a linguagem ainda não oferece suporte a testes de condição e laço, uma operação deverá sempre estar associada a uma atribuição.

Ex : varReal teste;  
varReal teste2 = 10;  
teste = 10 SOMA teste2;  
10 SOMA 10; // não funcionará

## Comando de saída

A impressão de informação na tela, se dará da seguinte maneira:  
Iniciará sempre com o comando MOSTRAR seguido de [\$

[nomeDaVariavel]] caso queira imprimir o valor da variável, ou apenas escrevendo o que se deseja mostrar na tela, finalizando, sempre com “.”.

Exs: MOSTRAR \$var1 ; // irá imprimir o valor de var1

    MOSTRAR o valor de var1 é \$var1; // imprime “o valor de var1 é ” e seu respectivo valor.

### **Comando de entrada**

A leitura de informação do teclado, se dará da seguinte maneira:

Iniciará sempre com o comando LER seguido de [nomeDaVariavel], que irá receber o valor a ser lido do teclado, finalizando, sempre com “;”.

Exs: LER var1 ; // irá ler para dentro de var1 o valor lido do teclado

    MOSTRAR informa um valor ;

    LER var1 ;

    MOSTRAR \$var1; // imprime o valor de var1, que foi lido do teclado.