ESTUDO DIRIGITO DE ALGORITMO E LOGICA EM PROGRAMAÇÃO (AULA 1-3)

Tipo de variável (caractere, inteiro, real, logico(v/f),);

Funções básicas (nome <- "Silva, “escreva(nome), leia(x), operadores lógicos (<>, >=, <=, =))

Estruturas Condicionais (se, então; se, senão; escolha - caso)

Estruturas de Repetição (enquanto; repita; para)

Funções básicas

**Leia ():** usado para descrever o valor da variável, seja ela caractere, inteiro ou real

Exemplo: “leia (mes)”

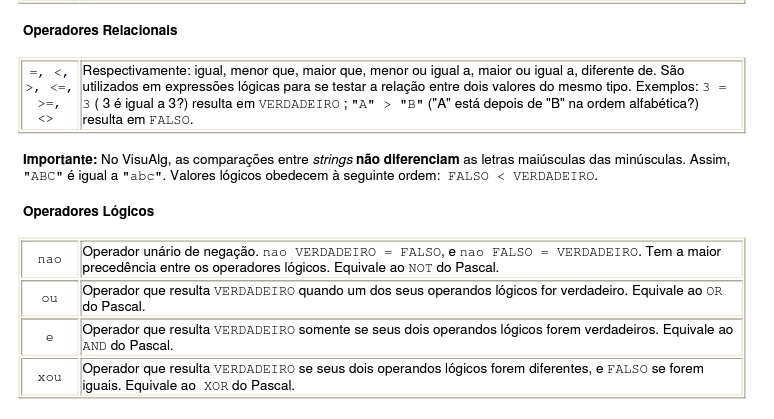
**Escreva (“...”, ...):** usado como pretexto para que o usuário escreva de acordo com a situação, quando se coloca aspas dentro dos parênteses, ela se torna uma frase explicativo, fora dos parênteses, o programa lê a função variável.

Exemplo: escreva ("digite o mes ", maio)

**Escreval**: tem a mesma função do escreva, mas pula linha.

**Atribuição** (<-): usado para indicar que a variável é resultante de uma outra var.

Exemplo: peso <- 30



ESTRUTURAS CONDICIONAIS:

**Se (...) ENTAO**- insere o comando ‘se’ sem a alternativa ‘senao’, ela dá uma instrução do “operador” caso o resultado seja correto ao que foi dito

Estrutura:

[se\* então

Fimse]

**Se(...)senao**- insere o comando ‘se’ completo, sendo o ‘senao’ opcional, ela da uma instrução do “operador”. Realiza ‘Então’ caso o resultado seja correto ao que foi dito; realiza ‘Senão’ caso o resultado seja oposto

Estrutura:

[Se\*então

Senão

Fimse]

Exemplo:

algoritmo "SacarDinheiro"

**var**

SaldoDisponivel : REAL

ValorDoSaque : REAL

inicio

SaldoDisponivel := 1000 //Assumimos que há 1000 reais de saldo na conta disponível para saque

ESCREVA ("Informe o valor do Saque: ")

LEIA (ValorDoSaque)

SE ValorDoSaque <= SaldoDisponivel ENTAO

SaldoDisponivel := SaldoDisponivel - ValorDoSaque

ESCREVAL ("Sacando R$ ", ValorDoSaque, ".")

SENAO

ESCREVAL ("O valor solicitado é maior que o valor disponível para saque!")

FIMSE

ESCREVAL ("Saldo disponível: R$ ", SaldoDisponivel)

fimalgoritmo

**Escolha\*Caso**- insere o comando ‘escolha’ (com a clausura ‘outrocaso’), quando outras verificações tenham falhado, permitindo ir direto ao bloco de notas dos códigos.

Estrutura:

[Escolha\*Caso

Outrocaso

Fimescolha]

Exemplo:

escolha (operador)  
      caso “+” // Caso o valor do operador for +  
         resultado <- numero1 + numero2 // O resultado recebe a soma dos números  
      caso “-” // Caso o valor do operador for -  
         resultado <- numero1 - numero2 // O resultado recebe a subtração do numero1 pelo numero2  
      caso “\*” // Caso o valor do operador for \*  
         resultado <- numero1 \* numero2 // O resultado recebe a multiplicação dos números  
      caso “/” // Caso o valor do operador for /  
         resultado <- numero1 / numero2 // O resultado recebe a divisão do numero1 pelo numero2

fimsescolha

**ESTRUTURAS DE REPETIÇÃO:**

**Para...faça**: essa estrutura repete uma sequência de comandos um número de vezes.

Estrutura:

para <variável> de <valor-inicial> ate <valor-limite> [passo <incremento>] faca

<sequência-de-comandos>

fimpara

Exemplo: tabuada

PARA contador DE 1 ATE 10 FACA  
      multiplicacao <- numero \* contador  
      escreval(numero, " X ", contador, " = ", multiplicacao)  
   FIMPARA

**Enquanto...Faça**: esta estrutura repete uma sequencia de comandos enquanto uma determinada condição (especificada através de uma expressão logica) for satisfeita.

Estrutura:

enquanto <expressão-lógica> faca

<sequencia-de-comandos>

Fimenquanto

Exemplo:

ENQUANTO (eh\_primo = VERDADEIRO) FACA

escreva(numero)

se ( (primo % numero) = 0 ) entao

eh\_primo <- FALSO

escreval(" Não é primo")

fimse

numero<- numero+ 1

FIMENQUANTO

**Repita...até:** esta estrutura repete uma sequencia de comandos até que uma determinada condição (especificada através de uma expressão logica) seja satisfeita.

Estrutura:

Repita

<seqüência-de-comandos>

ate <expressão-lógica>

Exemplo:

REPITA

      escreva("Digite um número para ser somado: ")

      leia(numero)

      soma <- soma + numero

      escreva("Deseja continuar digitando? [S/N]")

      leia(resposta)

   ATE (resposta <> "S")

repita

incremento<-incremento-1

soma<-soma+incremento

escreval(incremento, " -- ", soma)

ate(incremento = 1)