Algoritmos - Anotações Geral

▼ Comandos de Saídas

- **Escreva -** Para escrever qualquer coisa na tela preciso colocar “Escreva (”qualquer coisa”)”
- Escreval - O “l” significa “linha” então ele escreve e salta para linha de baixo. Ele funciona da mesma forma que o comando “Escreva”. “Escreval (”qualquer coisa”)”

▼ Comandos de Entrada

- **Leia** - Esse comando basicamente irá “ler” a variável referida a ele.

▼ Variáveis

Uma variável é um nome que definimos para armazenar dados de forma simples. O valor de uma variável pode ser alterado no andamento do algoritmo, por isso o nome de variável. No VisualG, uma variável deve ter sempre um identificador: tipo

▼ Identificadores

Existem 6 regras para determinar um identificador no VisualG

1. O identificador deve sempre começar com uma letra.
2. Os próximos caracteres podem ser letras ou números.
3. Não pode utilizar nenhum símbolo, exceto _ (under-line ou sublinhado)
4. Não podem conter espaços em branco.

5. Não pode conter palavras com acentos.
6. Não pode ser uma palavra reservada.

▼ **Tipos**

▼ Primitivos

Existem 4 tipos no VisualG, que são eles:

1. Inteiro: Permite que a Variável guarde apenas números inteiros.
2. Real: Permite que a Variável guarde apenas números reais.
3. Caractere: É tudo aquilo que é colocado entre aspas. Ex: “Pedro” “Algoritmo” “123”
4. Lógico: Guarda apenas 2 valores, verdadeiro e falso.

▼ Operadores Aritméticos

- **+** - Adição
- **-** - Subtração
- * - Multiplicação
- **/** - Divisão
- **** - Divisão Inteira
- **^** - Exponenciação
- **%** - Módulo

▼ *Alguns Exemplos*

Vamos considerar que a letra A vale o número 5 e que a letra B vale o número 2. Então fica:

A + B = 7

A - B = 3

A * B = 10

A / B = 2.5

A \ B = 2

A ^ B = 25

A % B = 1

▼ Ordem de Precedência

É basicamente tudo aquilo que irá vir em primeiro lugar. Veja abaixo em ordem numerada:

1. ( ) Parênteses - Sempre será considerado em primeiro lugar
2. ^ Exponenciação - Será considerado em segundo lugar
3. / * Divisão Multiplicação - Será sempre considerado em terceiro lugar
4. (+ -) Adição e Subtração - Sempre será em quarto lugar

▼ Funções Aritméticas

- Abs - Valor Absoluto
- Exp - Exponenciação
- Int - Valor Inteiro
- RaizQ - Raiz Quadrada
- Pi - Retorna o PI
- Sen - Seno (Em radiano)
- Cos - Cosseno (Em radiano)
- Tan - Tangente (Em radiano)
- GraupRad - Graus para radiano)

▼ Operadores Relacionais

Operadores Relacionais criam a relação entre variáveis ou expressões e gerar um valor lógico, come verdadeiro ou falso.

- > - Maior que
- < - Menor que
- >= - Maior ou Igual a
- <= - Menor ou Igual a
- = - Igual a
- <> - Diferente de

▼ Operadores Lógicos

- E - Eu quero uma coisa e outra, preciso que duas determinadas situações aconteçam como eu quero.
- OU - Diferente do Operador E, eu preciso apenas que uma das duas determinadas situações aconteça da forma que eu quero. (Pode também as duas situações acontecer)
- NÃO -

▼ Estrutura Condicionais

Cria uma condição como já sugere o nome, permitindo que o programa execute diversos comandos de acordo com as condições estabelecidas. Segue os comandos:

▼ Simples / Composta / Aninhada

- Se - “ apertar a tecla “W” pule”

▼ Escolha-Caso

- Escolha - Escolhe uma variável para funcionar
- Caso - Cria uma “escolha”

- Fimescolha - Delimita o final do código

▼ Estruturas de Repetição

▼ Enquanto

- Enquanto - Quando entrar no Enquanto, ele vai testar uma expressão. Se esse expressão for verdadeira ele.. (Leia o Faca)
- Faca - O Faca, sempre que o Enquanto for verdadeiro, ele vai executar algum bloco, e vai (Leia o FimEnquanto)
- FimEnquanto - Ao chegar no FimEnquanto, ele retorna para o Enquanto, e até que a expressão de falso, o ciclo se mantém.

▼ Repita

- Repita - Vai repetir um “acontecimento” que no caso é um bloco, Ate que uma expressão ocorra
- FimRepita - Fecha o comando Repita, encerra a repetição Ate que a expressão que está dentro do comando aconteça.

▼ Para

- Para -
- Ate -
- Passo -
- Faca -
- FimPara -

▼ Procedimentos

Basicamente um Procedimento é um “novo” local onde criamos códigos que serão utilizados no programa principal. Podemos por exemplo definir um procedimento chamado “Rotina” onde nele tem os códigos, quando eu quiser

usar novamente a “rotina” ao invés de escrever todas as linhas de códigos, posso simplesmente pedir pra executar o procedimento “rotina” onde já está os códigos já prontos, com isso eu economizo tempo e faço todo o código do programa ser menor.

O procedimento no VisualG sempre irá abaixo do campo das variáveis e acima do inicio.

Simplificando, fica entre as duas.

▼ Comandos

Procedimento Nome do Procedimento ( ) - Toda vez que eu quiser criar um Procedimento novo, devemos escrever Procedimento o nome do procedimento logo a frente. Os parênteses ( ) vou explicar abaixo como funciona.

Inicio - Onde vamos escrever todos os códigos que estrão dentro Procedimento

FimProcedimento - O comando para finalizar o Procedimento.

( ) - O parênteses serve para passar parâmetros para o Procedimento, esses parâmetros podem ser do tipo Valor e do tipo Referência.

▼ Funções

Funcao Nome Funcao ( ) : Tipo do retorno - Toda vez que eu for criar uma Função, ela deve seguir dessa forma, igual ao Procedimento, mas com um acréscimo, que é o “Tipo de Retorno” que vou explicar abaixo.

Inicio - Onde vamos escrever todos os códigos que estrão dentro da Função.

FimFunção - O comando para finalizar a Função.

Tipo de Retorno - O Retorno é o que o nome sugere. As funções podem retornar um valor, que são eles igual aos das variáveis, ou seja: Inteiro, Real, Caractere

ou Lógico. Esse é o valor que uma Função vai retornar para o programa principal por meio de uma variável.

▼ Vetores

Vetores são um tipo de Variável, mas ao contrário das Variáveis primitivas que armazenam apenas um valor por variável, os vetores pode armazenar vários números em apenas uma variável.

Nome Vetor: vetor [1..3] tipo da variável

O que esse nome nos diz é.

- Nome Vetor nada mais é que o nome que daremos ao nosso Vetor, igual já fazemos com as Variáveis, segue as mesmas regras.
- A palavra Vetor a frente do nome é para indicar que isso que estamos criando é um Vetor.
- [1..3] Esse é o índice, que como o nome sugere, é para indicar quantas "memórias" vamos criar nesse Vetor, quantas variáveis ele tem. Nesse caso, o Vetor irá ter 3 variáveis dentro dele, porque está indo de 1 a 3. É sempre necessário colocar os dois pontos entre os números, como no exemplo.
- Tipo Variável é o tipo de Vetor que estamos criando, se é um Vetor com valores Inteiros, Reais, Caractere ou Lógico assim como nas Variáveis. Sempre irá ter um "de" antes do tipo, como no exemplo.

Exemplo:

> N1: Vetor[1..5] de real - Aqui estou criando um Vetor chamado N1, declarando que ele é um Vetor. Logo a frente, como o [1..5] estou atribuindo a ele seu número de "memória" no caso, seus números de Variáveis, e em seguida declaro seu tipo, com o "de real"

Tudo isso que falei logo acima também tem o nome técnico de: Variáveis Compostas Homogêneas Unidimensionais

▼ Matrizes