Aula # 1

Tema 2 – Introdução à algoritmia e algorítmos fundamentais

Sumário: I.D.E e Indentação e comentários.

**IDE** (*Integrated Development Envoirment – Ambiente de Desenvolvimento Integrado*) é uma ferrramenta (software) usado no desenvolvimento de aplicações que combinem diferentes funcionalidades em única interface gráfica de usuários.

IDEs ajudam os desenvolvedores a programar novas aplicações de forma rápida.

É possível desenvolver aplicações sem uma IDE.

As funções principais dentro da IDE são:

**1 – Editor de código fonte:** é um editor de texto que auxilia na criação de código de software por meio de funcionalidades como destaque da sintaxe com indicadores viisuais, recurso de preenchimento automático específicos da linguagem e verificação de bugs (erros) durante o desenvolvimento.

**2 – Compilador** são tarefas como compilação de código fonte em código binário, criação de pacotes de código binário e execução de testes automatizados.

**3 – Debugger** é um programa usado paa testar ouros programas e mostrar graficamente a localização do bug no código original.

**Indentação** serve para organizar o código deixando explícito as dependências que as instruções têm uma das outras.

**Comentários** é um texto que escrevemos no código para melhor explicar as instruções aplicadas.

Ambos não têm qualquer efeito na compilação e execução do programa. É uma característica pessoal de cada programador.

**Tarefa**

1 – Define IDE

2 – Dê 5 exemplos de IDEs existentes no mercado.

Aula # 2

Tema 2 – Introdução à algoritmia e algorítmos fundamentais

Sumário: Armazenamento em ficheiros.

**VISUAL G**

arquivo “caminho”

**PORTUGOL**

inclua biblioteca ***Arquivos --> a***

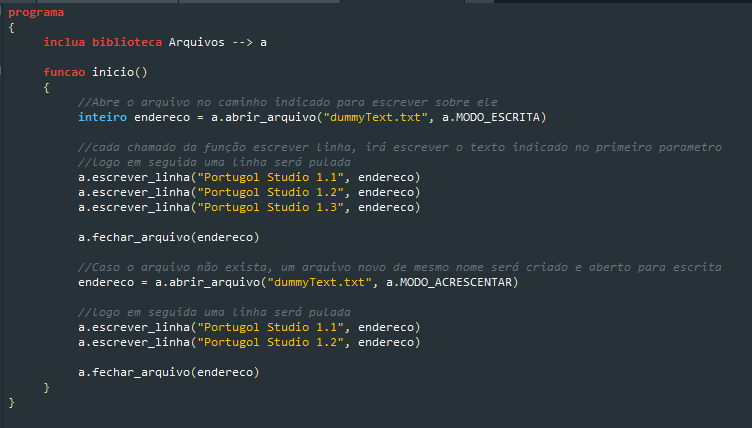
Modos de Abertura:

- MODO\_ESCRITA

- MODO\_LEITURA

- MODO\_ACRESCENTAR

Após ler os dados desejados, é necessário fechar o arquivo. Isso é feito para liberar a memória que está sendo utilizada pelo arquivo e também para que outros programas possam acessá-lo.





Aula # 3

Tema 2 – Introdução à algoritmia e algorítmos fundamentais

Sumário: variáveis indexadas.

TIPOS DE DADOS

**Tipos Primitivos de Dados** Como estudado anteriormente, são pré-definidos pela linguagem e permitem a definição de novos tipos.

Ex: Inteiro, Real, Caractere, Lógico etc.

**Tipos da Dados Estáticos** São gerados a partir de tipos já existentes e não sofrem alterações em suas “características” (escopo) durante a execução do programa.

Ex: Vetor, Matriz e Registro.

**Tipos de Dados Dinâmicos** São aqueles que sofrem alterações durante a execução do programa.

Ex: Ponteiros (Pilha, Fila, Lista, Árvore etc).

**VETOR**

O vetor é uma estrutura de dados homogênea (de mesmo tipo) e unidimensional. Também conhecida como Variável Indexada, Tabela ou Array.

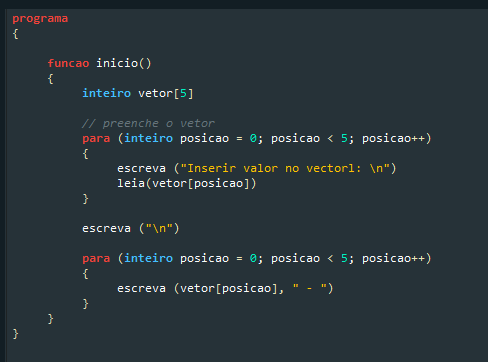
O vetor possuí um índice que identifica a posição do dado armazenado, esse índice é do tipo Inteiro e fica representado entre colchetes.

Forma geral para declaração do vetor:

<NOME> : VETOR[1..N] DE <TIPO>;

**Tarefa**

Faça um programa que leia quatro números inteiros, coloque-os em um vetor e mostre-os na ordem inversa da leitura.



Aula # 4

Tema 2 – Introdução à algoritmia e algorítmos fundamentais

Sumário: variáveis indexadas - Matriz.

A matriz é uma estrutura de dados homogênea multidimensional.



Aula # 5

Tema 2 – Introdução à algoritmia e algorítmos fundamentais

Sumário: funções e procedimentos

Funções e procedimentos também chamados de subalgoritmos - são trechos de algoritmos que efetuam uma determinada tarefa específica dentro do programa.

Ao invés de escrever um código grande, podem-se escrever vários algoritmos menores (Modularização). Em conjunto, resolvem o problema proposto.′

É conveniente utilizá-los quando uma tarefa é efetuada em diversos lugares no mesmo programa. Ao invés de escrever um trecho diversas vezes, escreve-se um subalgoritmo e chama-o diversas vezes.

Têm a seguinte importância:

Reduzem o tamanho do algoritmo

Facilitam a compreensão e visualização do algoritmo

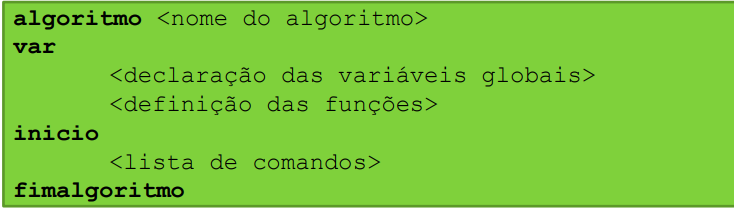
Uma vez declarados no algoritmo, podem ser chamados em qualquer ponto do programa.

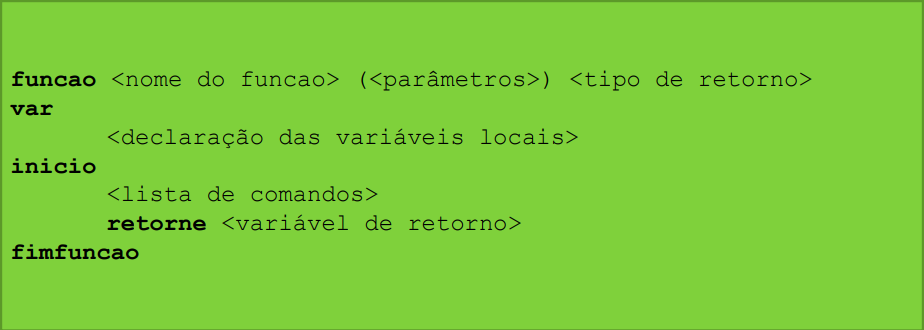
**Definição:** Uma função é um bloco de comandos que tem por objetivo retornar um valor ou uma informação.

A chamada de uma função é feita através da citação do seu nome seguido, opcionalmente, de seus argumentos iniciais entre parênteses.

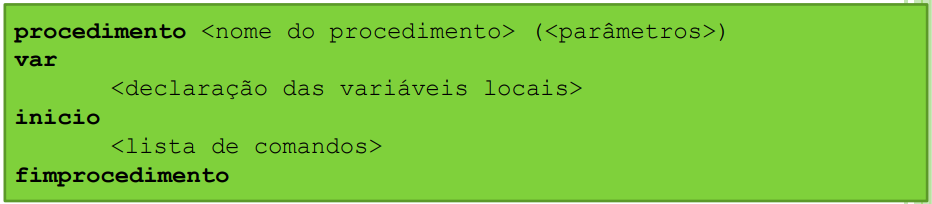
SINTAXE

Para VisualG, a criação de uma função deve ser realizada dentro da seção de variáveis.

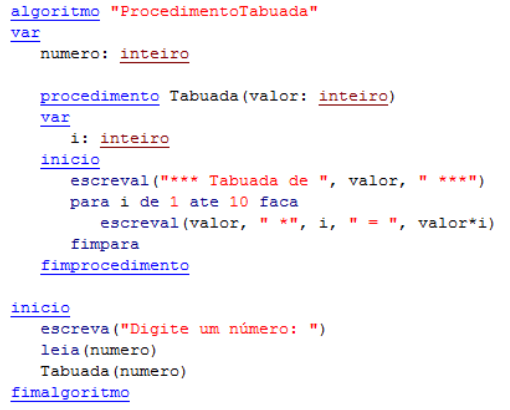




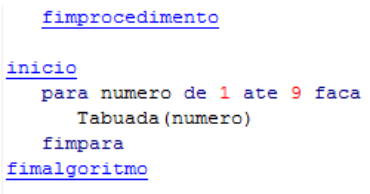
Em Visualg, os procedimentos diferem das funções apenas por não retornarem valor nenhum. A sintaxe utilizada na criação de procedimentos é:



Exercício: Crie um procedimento que receba um número como parâmetro e escreva a tabuada desse número.



Podemos alterar o programa acima fazendo com que imprima as tabuadas de 1 a 9



DIFERENÇA ENTRE FUNÇÃO E PROCEDIMENTO

Funções que retornam algum valor.

Procedimento (ou subrotina) que não retorna nada.

As funções podem ser predefinidas pela linguagem ou criadas pelo programador de acordo com o seu interesse.

**BIBLIOTECAS DE FUNÇÕES**

Armazenam um conjunto de funções que podem ser usadas pelos programas.

Por exemplo, a execução de uma instrução de entrada e saída, a classificação dos dados de um arquivo, o cálculo de funções matemáticas, etc.

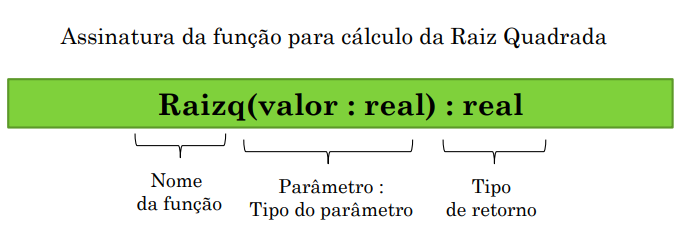
Uma linguagem de alto nível geralmente incorpora diversas rotinas prontas (que fazem parte da linguagem) e que compõem bibliotecas (librarys) de funções pré-programadas que poderão ser utilizadas pelo programador, poupando tempo, aumentando a eficiência e evitando erros.

Dessa forma, um programa em alto nível possivelmente conterá diversas chamadas de biblioteca (library calls).

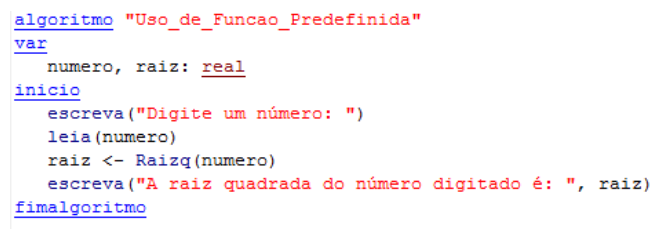
Essas funções não devem ser confundidas com as instruções da linguagem - na realidade, são pequenos programas externos que são chamados através de instruções especiais de chamada de biblioteca.

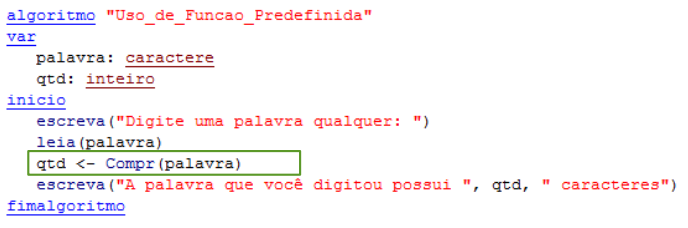
Para serem executadas, essas rotinas precisam ser incorporadas ao código do programador, isto é, a chamada de biblioteca precisa ser substituída pelo código propriamente dito, incluindo os parâmetros necessários.

**FUNÇÕES PREDEFINIDAS EM VISUALG**

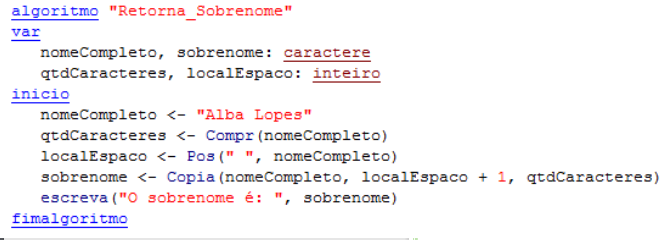


Exemplos:

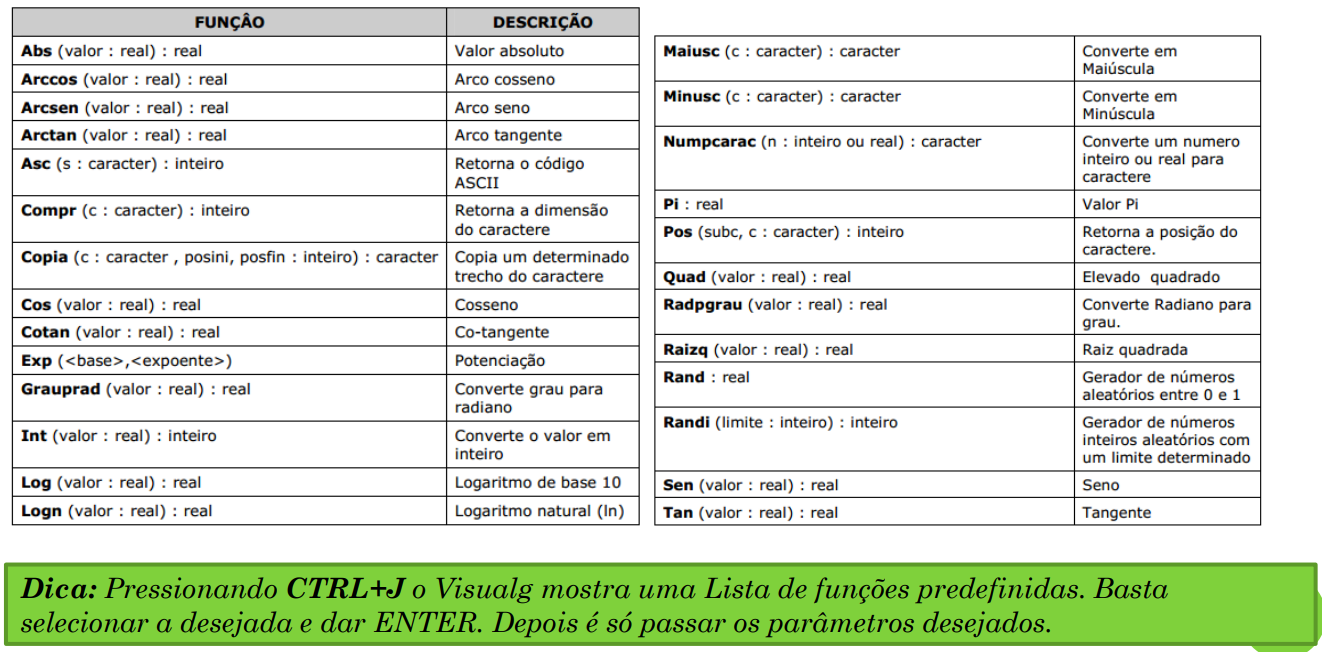




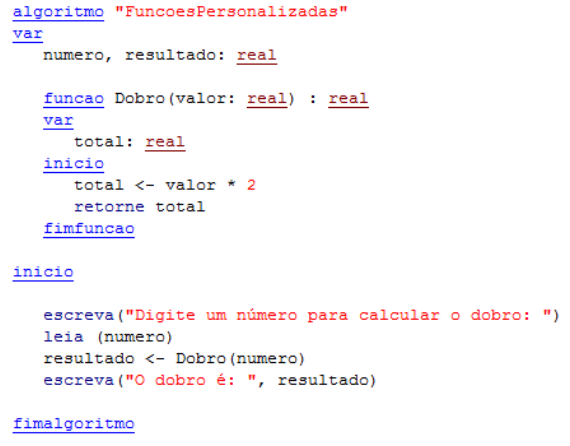
Criar um algoritmo que, dado o nome de uma pessoa (Nome + Sobrenome), retorna apenas o sobrenome. Ex: “Alba Lopes” , retorna somente “Lopes”.

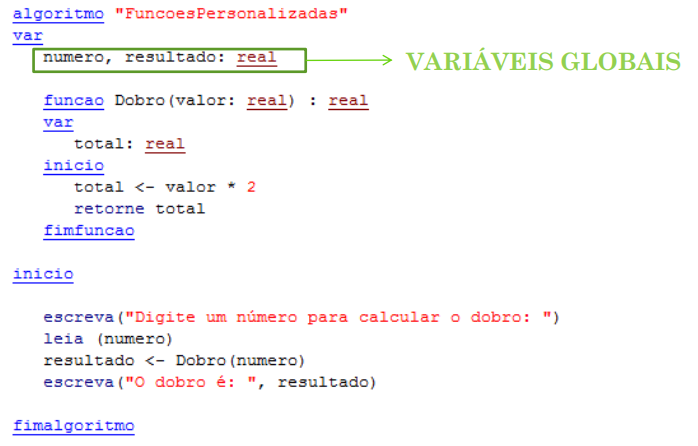


**LISTA DE FUNÇÕES PREDEFINIDAS EM VISUALG**



CRIANDO FUNÇÕES



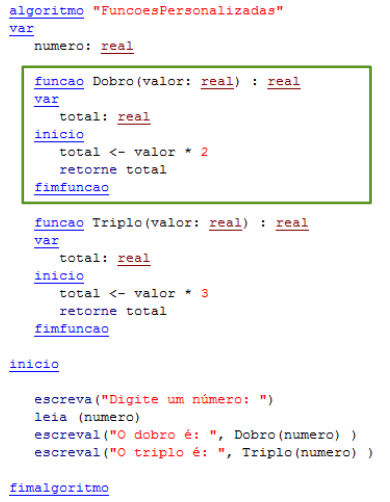


Variáveis Locais: Declaradas dentro dos subalgoritmos (funções ou procedimentos) Podem ser usadas APENAS dentro das funções

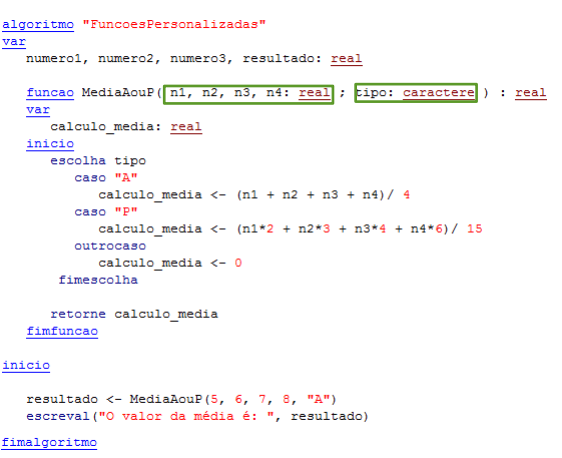
O algoritmo que chamou a função/procedimento não tem acesso à estas funções ′

Variáveis Globais São variáveis declaradas na seção var do algoritmo. Qualquer função/procedimento pode alterar o valor ou utilizá-la durante o seu processamento.

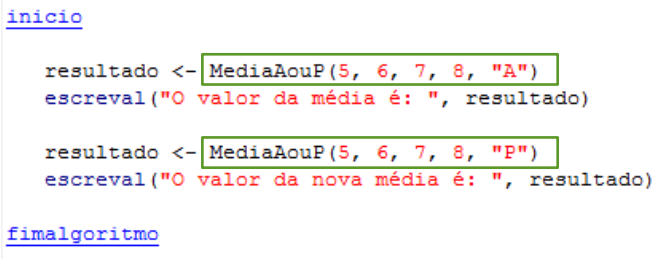
Os algoritmos podem possuir várias funções

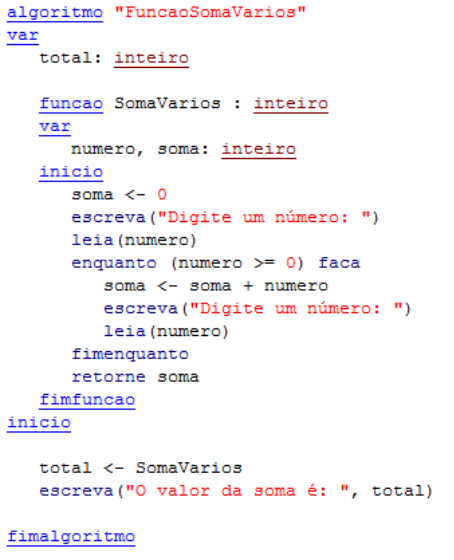


Escreva uma função que recebe as 4 notas de um aluno por parâmetro e uma letra. Se a letra for A a função calcula a média aritmética das notas do aluno, se for P, a sua média ponderada (pesos: 2, 3, 4 e 6).

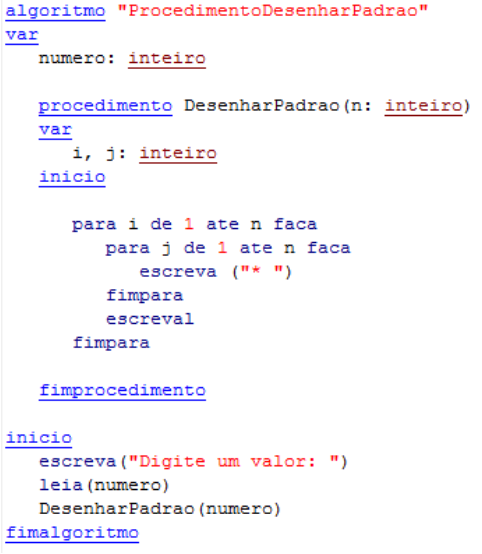


Mais saídas de uma única vez:

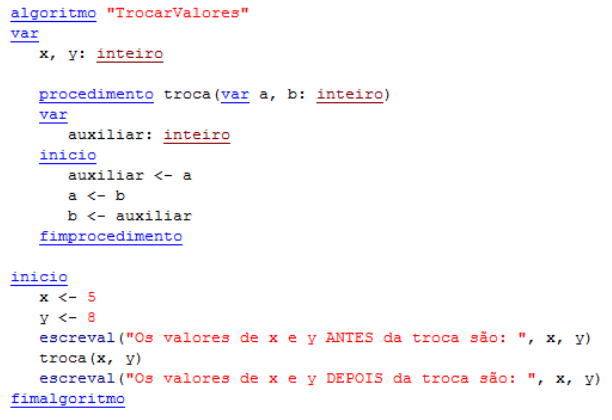




Crie um procedimento que, dado um número N por parâmetro, desenhe o seguinte padrão na tela.



Crie um procedimento que receba dois valores inteiros por parâmetro e realize a troca desses valores.



Aula # 6

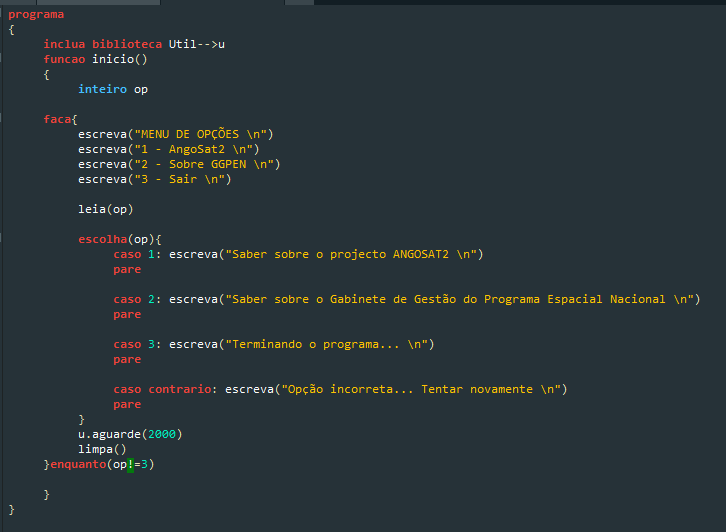
Tema 2 – Introdução à algoritmia e algorítmos fundamentais

Sumário: Comandos especias: (limpar tela, aguardar)



**Exemplo: Menu de opções**





Aula # 7

Tema 2 – Introdução à algoritmia e algorítmos fundamentais

Sumário: História do VisualG e Portugol Studio.

https://sourceforge.net/projects/visualg30/

Autor ANTONIO CARLOS NICOLODI, 38 anos na área de informática como: Analista de sistemas, desenvolvedor de softwares em várias linguagens de programação:

C++, Assembly, Pascal(Delphi), Basic, Cobol, Clipper, Java, etc.

Refiz esta nova versão e estou disponibilizando GRÁTIS o: "VISUALG 3.0".

Entre outras : (novo layout, nova roupagem até 05 tipos de peles)

e novos comandos, também reconhece comandos em português correto:

( PARA ... FAÇA, SE .. ENTÃO .. SENÃO) e

o operador lógico NÃO, mas em maiúsculo e

os comandos antigos ainda são reconhecidos para manter a compatibilidade.

Contactos por e-mail/twitter/Blog :

E-mail:professor.antonio.nicolodi@gmail.com

Twitter: @visualg30

Blog: http://antonionicolodi.blogspot.com.br/

E-mail:professor.antonio.nicolodi@gmail.com  
Twitter: @visualg30  
Blog: <http://antonionicolodi.blogspot.com.br/>

Link VisualG [VISUALG 3.0 download | SourceForge.net](https://sourceforge.net/projects/visualg30/)