

Nome: _____ Data: ____/____/____

1. Modularização**1.1. Conceito**

A modularização consiste num método utilizado para facilitar a construção de grandes programas, através de sua divisão em pequenas etapas, que são os **módulos** ou **subprogramas**. A primeira delas, por onde começa a execução do trabalho, recebe o nome de **programa principal**, e as outras são os **subprogramas** propriamente ditos, que são executados sempre que ocorre uma chamada dos mesmos, o que é feito através da especificação de seus nomes.

1.2. Vantagens da utilização de subprogramas

- Economia de código: escreve-se menos;
- Desenvolvimento modularizado: pensa-se no algoritmo por partes;
- Facilidade de depuração (correção/acompanhamento): é mais fácil corrigir/detectar um erro apenas uma vez do que dez vezes;
- Facilidade de alteração do código: se é preciso alterar, altera-se apenas uma vez;
- Generalidade de código com o uso de parâmetros: escreve-se algoritmos para situações genéricas.

Há duas espécies de subprogramas: **PROCEDIMENTO** e **FUNÇÃO**.

PROCEDIMENTO

Um subprograma do tipo PROCEDIMENTO é, na realidade, um programa com vida própria, mas que, para ser processado, tem que ser solicitado pelo programa principal que o contém, ou por outro subprograma, ou por ele mesmo.

Procedimento sem parâmetro

Sintaxe: Procedimento Somar;
 Total: numérico
 Total ← N1 + N2
 escreva Total
 fim procedimento

Procedimento com parâmetro

Sintaxe: Procedimento Somar(A, B: numérico);
 Total: numérico
 Total ← A + B
 escreva Total
 fim procedimento

FUNÇÃO

As funções, embora bastante semelhantes aos procedimentos, têm a característica especial de retornar ao programa que as chamou um valor associado ao nome da função. Esta característica permite uma analogia com o conceito de função da Matemática.

Função sem parâmetro

Sintaxe: Função Somar: numérico;
 Total: numérico
 Somar \leftarrow N1 + N2
 fim função

Função com parâmetro

Sintaxe: Função Somar(A, B: numérico): numérico;
 Total: numérico
 Somar \leftarrow A + B
 fim função

1.3. Parâmetros

Quando se deseja escrever um subprograma que seja o mais genérico possível, deve-se usar a passagem de parâmetros.

A passagem de parâmetros formaliza a comunicação entre os módulos. Além disto, permite que um módulo seja utilizado com operandos diferentes, dependendo do que se deseja do mesmo.

1.4. Variáveis globais e variáveis locais

O uso de *variáveis globais* dentro de procedimentos e funções serve para implementar um mecanismo de transmissão de informações de um nível mais externo para um mais interno.

As *variáveis locais* dos procedimentos e funções são criadas e alocadas quando da sua ativação e automaticamente liberadas quando do seu término.

A utilização de *variáveis globais* não constitui, no entanto, uma boa prática de programação. Assim, todos subprogramas devem apenas utilizar as *variáveis locais*, conhecidas dentro dos mesmos, e a transmissão de informações para dentro e fora dos subprogramas deve ser feita através dos ***parâmetros*** de transmissão.