PROGRAMAÇÃO ESTRUTURADA (MODULAR) O que seria a Programação Estruturada ou Modular? Oque é a técnica de **Modularização**? Quais as vantagens de se usar a técnica de Modularização? Quais são os 2 tipos de Módulos básicos que existem? Como se dá o Módulo Procedimento? - reservada para a questão acima -Ilustre como podemos implementar um Procedimento... Como se dá o Módulo Função? - reservada para a questão acima -Ilustre como podemos implementar uma Função...

PROGRAMAÇÃO ESTRUTURADA (MODULAR)

Consiste na divisão de grandes tarefas de computação em partes menores, a fim de utilizar seus resultados parciais para compor o resultado final desejado. Dessa forma, há uma diminuição da extensão dos programas, de forma que qualquer alteração poderá ser feita mais rapidamente, caso seja necessária.

A modularização é uma técnica utilizada para desenvolver algoritmos, na qual se divide o problema em pequenas partes denominadas módulos, sendo estes também conhecidos também como sub-rotinas, sub-programas ou sub-algoritmos.

A principal vantagem da modularização é possibilitar o reaproveitamento de código, já que podemos utilizar um módulo várias vezes, eliminando assim a necessidade de escrever o mesmo código em situações repetitivas. As vantagens da programação estruturada são:

- Deixar o programa mais legível e fácil de entender;
- Encontrar e corrigir erros com mais facilidade;
- Facilitar os testes, diminuindo a probabilidade de erros;
- Reduzir o tempo e custo da programação;
- Possibilitar o reaproveitamento de partes do programa (módulos);

Basicamente, há 2 tipos de módulos: **Procedimentos** e **Funções**. Entre esses dois tipos de módulos existem algumas diferenças, mas o conceito é mesmo para ambas. *O importante no uso prático desses dois tipos de módulo é distinguir as diferenças entre eles e como utilizá-los no momento mais adequado.*

Um procedimento é um bloco de programa contendo início e fim, identificado por um nome, por meio do qual será referenciado em qualquer parte do programa principal ou em outro procedimento. A característica principal de um procedimento é que não há retorno de valor ou variável, pois, no momento da execução do programa, somente as instruções definidas dentro do escopo do procedimento são executadas. Quando um procedimento é chamado por um programa, ele é executado até o seu término e a execução do programa volta exatamente para a primeira linha de instrução, após a linha que fez a chamada do procedimento.

A forma geral para se definir um procedimento é a seguinte:

void nome-do-procedimento (lista de parâmetros){ corpo do procedimento (instruções)

Temos um Procedimento de Entrada para uma calculadora, que quando for chamado solicitará a entrada ao usuário das variáveis A e B... void Entrada ()

```
printf("\nDigite o primeiro numero: ");
scanf("%f",&A);
printf("\nDigite o segundo numero: ");
scanf("%f",&B);
```

Uma função também é um bloco de programa, como são os procedimentos, contendo início e fim e identificada por um nome, pelo qual também será referenciada em qualquer parte do programa principal. A principal diferença entre um procedimento e uma função está no fato de uma função retornar um determinado valor, sendo que o tipo de dado (int, float, char) de tal valor deve ser definido na criação da função.

A forma geral para se definir uma função é a seguinte:

tipo-de-dado nome-da-função (lista de parâmetros){
 corpo da função (instruções)
 return (variável/valor);
}

A seguir, apresentamos a definição do Módulo Adição citado no exemplo da calculadora. Quando esta função for chamada e executada, será feito o cálculo de adição das variáveis A e B, o qual será atribuído à variável R, que será o retorno da função.

PROGRAMAÇÃO ESTRUTURADA (MODULAR) Como seria a forma geral da utilização de Procedimentos e Funções dentro de um programa? Ilustre um programa de uma calculadora com procedimentos e funções totalmente implementados... - reservada para a questão acima -- reservada para a questão acima -

PROGRAMAÇÃO ESTRUTURADA (MODULAR)

A forma geral da utilização de um procedimento é a seguinte: nome-do-procedimento (lista de parâmetros);

Para utilizarmos uma função, normalmente, devemos declarar uma variável que receberá o retorno da função. Então, a forma geral da utilização de uma função é a seguinte:

Variável = nome-da-função (lista de parâmetros);

```
#include<stdio.h>
#include<stdlih h>
#include<conio.h>
float A, B, R;
void Entrada ()
    printf("\nDigite o primeiro numero: ");\\
    scanf("%f",&A);
   printf("\nDigite o segundo numero: ");
    scanf("%f",&B);
float Adicao ()
   return (R);
    return (R);
float Multiplicacao ()
    R = A * B
    return (R);
float Divisao ()
    R = A / B:
   return (R);
main ()
int opcao=0:
 float result:
 while (opcao!=5)
 system("cls");
 printf("\n1-adicao");
 printf("\n2-subtracao");
 printf("\n3-multiplicacao");
 printf("\n4-divisao");
 printf("\n5-sair");
 printf("\n\n-Escolha um num:");
 scanf("%d",&opcao);
 switch(opcao)
   case 1:
     Entrada ();
     result = Adicao ();
     printf ("\n A + B = %f", result);
     break;
 }
case 2:
     Entrada ();
     result = Subtração ():
     printf ("\n A - B = %f", result);
     break:
case 3:
     Entrada ():
     result = Multiplicacao ();
     printf ("\n A * B = %f", result);
     break;
   case 4:
     printf ("\n A / B = %f", result);
     break;
   default:printf("\nEsta operacao nao existe");
system("PAUSE");
```

PROGRAMAÇÃO ESTRUTURADA (MODULAR) Qual seria outra forma de implementar a Estrutura Modular em um programa? Ilustre a implementação de Procedimentos e Funções por através de Protótipos num programa... - reservada para a questão acima -- reservada para a questão acima -

PROGRAMAÇÃO ESTRUTURADA (MODULAR)

Existe, também, outra forma de estruturarmos um programa por meio de módulos, os quais podem ser definidos após o programa principal (main). Neste caso, como os módulos são criados após o programa principal, então deverá haver um protótipo para cada módulo logo após a seção include. O protótipo consiste no cabeçalho do módulo.

```
#include<stdio.h>
#include<stdlib.h>
float A, B, R;
/************
Prototipos dos modulos
float Adicao ():
float Divisao ();
/********************/
main ()
int opcao=0;
while (opcao!=5)
   printf("\n1-adicao"):
   printf("\n2-subtracao");
   printf("\n3-multiplicacao");
printf("\n4-divisao");
   printf("\n5-sair");
   printf("\n\n-Escolha um num:");
    scanf("%d",&opcao);
    switch(opcao)
        result = Adicao ();
        printf ("\n A + B = %f", result);
      case 2:
        Entrada ();
        result = Subtracao ():
        printf ("\n A - B = %f", result);
        break;
      case 3:
        Entrada ();
result = Mu
        printf ("\n A * B = %f", result);
        break:
     case 4:
        Entrada ();
        printf ("\n A / B = %f", result);
        break;
        printf("\nEsta operacao nao existe");
void Entrada ()
   printf("\nDigite o primeiro numero: ");
   scanf("%f",&A);
   printf("\nDigite o segundo numero: ");
   scanf("%f",&B);
float Adicao ()
   return (R);
float Subtracao ()
   return (R);
float Multiplicacao ()
   return (R);
   return (R);
```