Programação para Engenharia I - A

Modularidade

Procedimentos e Funções

Alexandre Agustini

versão original desenvolvida pelo
Prof.Daniel Callegari

Modularidade

- Muitos problemas grandes (ou de complexa solução) podem ser divididos, sucessivamente, em problemas menores, com lógica mais simples e de compreensão mais fácil.
- Aos trechos de algoritmo que efetuam um ou mais cálculos determinados dá-se o nome de subalgoritmos.

Material adaptado de [Orth 2001]

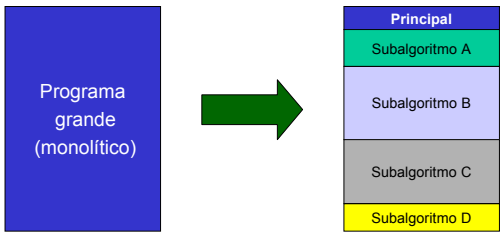
Vantagens da Modularização

- Dividir problemas grandes em vários problemas menores, de baixa complexidade.
  - Número pequeno de variáveis
  - Poucos caminhos de controle (caminhos do início ao fim)
- Possibilidade de utilizar-se soluções gerais para classes de problemas em lugar de soluções específicas para problemas particulares.
  - Reusabilidade
  - Solucionar uma única vez o problema

Vantagens da Modularização

- Permite delimitar o escopo (nível de abrangência) de variáveis.
  - Variáveis locais
- Evita a repetição, dentro de um mesmo algoritmo, de uma seqüência de ações em diferentes pontos.

Modularização



Subalgoritmos

- Um subalgoritmo (ou um módulo) pode ser:
  - Um Procedimento; ou
  - Uma Função

Subalgotimo do tipo Procedimento

```
Procedimento Calcular_e_ExibirMediaVendas
   soma : Inteiro

   Para i = 1 até 12
      soma = soma + vetor(i)
   Fim_Para

   MsgBox "Média do ano = " + soma/12

Fim Procedimento
```


Subalgotimo do tipo Função

```
Função VerificaInterface : Booleano

   Se  txtNome <> ""  e  CInt(txtIdade.Text) > 0  Então
      retornar Verdadeiro
   Senão
      retornar Falso

Fim Função
```

Conceito novo:

Uma função retorna um valor

Subalgoritmos

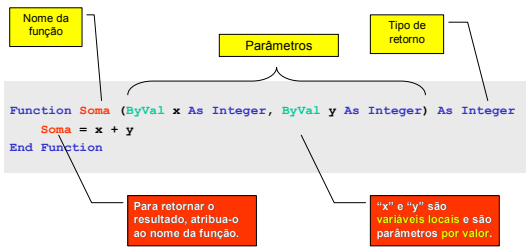
- Os subalgoritmos podem funcionar com base em informações que passamos a eles.
- A estas informações damos o nome de argumentos ou parâmetros.

Exemplo de Função em VB

```
Function Soma (ByVal x As Integer, ByVal y As Integer) As Integer
    Soma = x + y
End Function
```

```
Private Sub cmdCalcular_Click()
    Dim a As Integer
    Dim b As Integer
    Dim result As Integer
    a = 3
    b = 7
    result = Soma(a, b)
    MsgBox result
End Sub
```


Olhando mais de perto...



Outro Exemplo de Função em VB

```
Function SomaMenoresQue (ByVal v As Integer) As Integer
    Dim soma As Integer
    Dim i As Integer

    soma = 0
    For i = 1 To 100
        If vetorGlobal(i) < v Then soma = soma + vetorGlobal(i)
    Next

    SomaMenoresQue = soma

End Function
```

Resumo e Observações

- Subalgoritmos permitem a modularização, ou seja, dividir um problema em problemas menores.
- Subalgoritmos permitem a reutilização, evitando duplicação de código e facilitando a manutenção.
- Subalgoritmos podem ser do tipo Procedimento ou Função.
- Funções geralmente executam cálculos ou pesquisas e retornam um único valor, do tipo especificado em sua declaração.
- Procedimentos executam ações e não retornam nenhum valor (mas podem mudar a interface ou exibir um MsgBox).
- Procedimentos e funções podem receber parâmetros de entrada.
- A chamada de um subalgoritmo (procedimento ou função) deve obedecer, na ordem e em mesmo número, aos parâmetros definidos em sua declaração.