Funções (modularização)

Patrícia de Siqueira Ramos

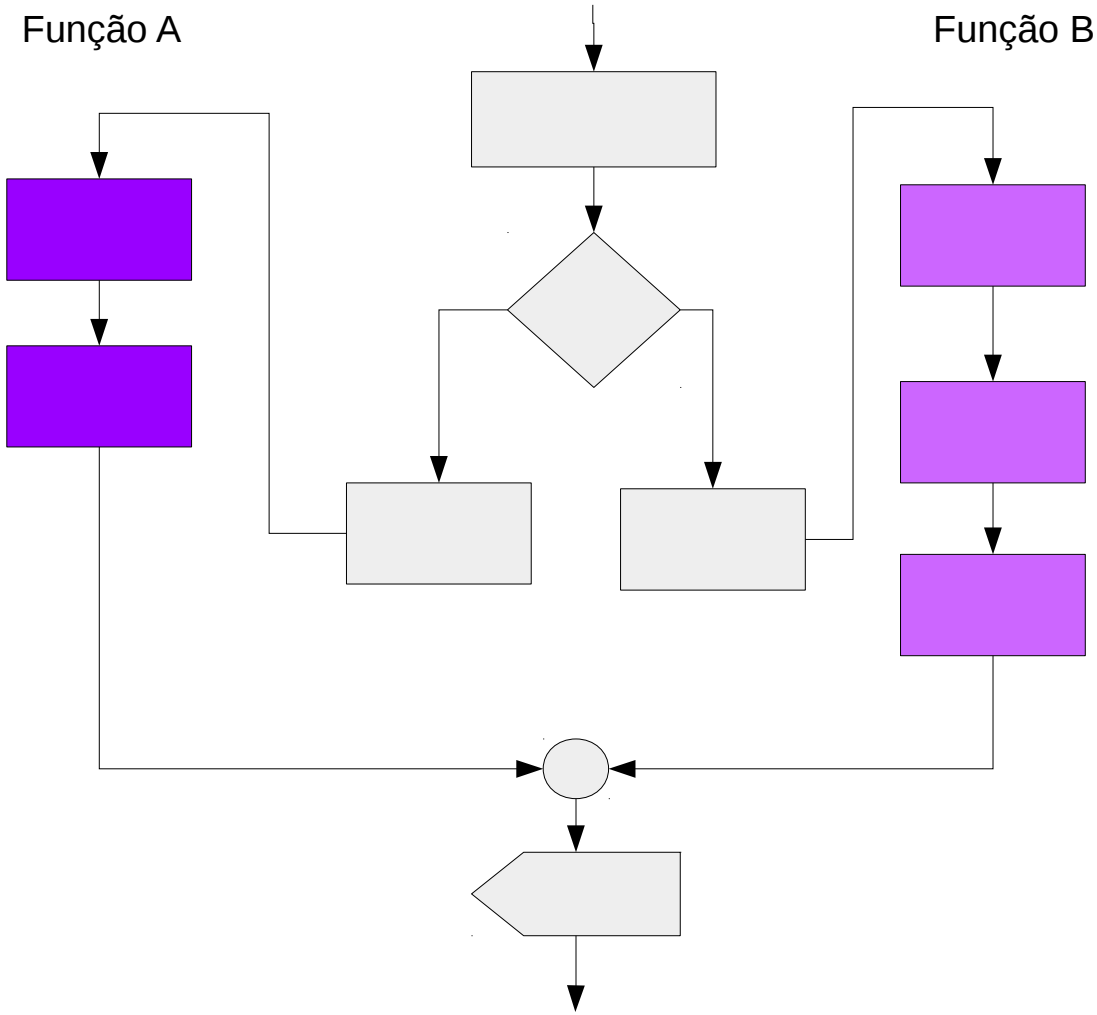
UNIFAL-MG, *campus* Varginha

1 de Outubro de 2019

Funções

- A ideia de uma função é decompor o algoritmo em módulos, de modo que tarefas mais complicadas sejam divididas em tarefas mais simples

Divisão de um fluxograma em funções



Funções

- Vantagens de se usar função:
  - um problema complexo pode ser dividido em problemas menores
  - a correção fica mais simples

Funções

- Vantagens de se usar função:
  - um problema complexo pode ser dividido em problemas menores
  - a correção fica mais simples
- Uma função computacional é como uma função matemática, que recebe uma entrada e retorna uma saída. Ex.:
  $f(x, y) = x + 3y$
  $f(2, 3) = 11$

Sintaxe de uma função - pseudocódigo

```
Função nome(argumentos formais)
    <declarações de variáveis>
    <comandos da função>
    <valor de retorno>
Fim
```

Para usar uma função:

```
nome(argumentos atuais)
```

Exemplo 1 - função que soma dois números

```
Função soma(a, b)
    Real: a, b, s
    s = a + b
    Retorne(s)
Fim
```

Exemplo 1 - função que soma dois números

```
Função soma(a, b)
    Real: a, b, s
    s = a + b
    Retorne(s)
Fim
```

Algoritmo que lê dois números e utiliza a função soma:

```
Início
    Real: x, y, w
    Leia(x, y)
    w = soma(x, y)
    Escreva('Soma:', w)
Fim
```

Observações

- Uma função pode não receber nenhum parâmetro
- Exemplo:

```
Função nada()
    Inteiro: x
    x = 3 * 4
    Retorne(x)
Fim
```

Observações

- Uma função pode não receber nenhum parâmetro
- Exemplo:

```
Função nada()
    Inteiro: x
    x = 3 * 4
    Retorne(x)
Fim
```

Uso:

```
Início
     Inteiro: num
     num = nada()
     Escreva(num)
Fim
```

Observações

- Uma função pode servir apenas para mostrar algo na tela
  (mas devemos evitar o uso de mensagens e leituras a partir do teclado dentro de funções)
- Exemplo:

```
Função mostra(nome)
    String: nome
    Escreva('Oi,', nome)
Fim
```

Observações

- Uma função pode servir apenas para mostrar algo na tela
  (mas devemos evitar o uso de mensagens e leituras a partir do teclado dentro de funções)
- Exemplo:

```
Função mostra(nome)
    String: nome
    Escreva('Oi,', nome)
Fim
```

Uso:

```
Início
    String: aluno
    Escreva('Insira seu nome:')
    Leia(aluno)
    mostra(aluno)
Fim
```

Exemplo 2 - função que retorna a raiz cúbica de um número

```
Função raiz3(num)
    Real: num, r
    r = num ** (1 / 3)
    Retorne(r)
Fim
```

Algoritmo que utiliza a função raiz cúbica:

```
Início
    Real: z, raiz
    Leia(z)
    raiz = raiz3(z)
    Escreva('Raiz cúbica é', raiz)
Fim
```

Exercício 1

Elabore uma função que recebe um número `num` e o grau `n` da raiz. Tal função deve se chamar `raizn(num, n)`.
Por exemplo:

- `raizn(9, 2)` deve retornar 3
- `raizn(27, 3)` deve retornar 3
- `raizn(256, 4)` deve retornar 4

Também faça um algoritmo que use tal função.

Exemplo 3 - função que retorna o maior de dois números

```
Função maior2(a, b)
    Real: a, b
    Se a > b Então
        Retorne(a)
    Senão
        Retorne(b)
    FimSe
Fim
```

Exemplo 3 - função que retorna o maior de dois números

```
Função maior2(a, b)
    Real: a, b
    Se a > b Então
        Retorne(a)
    Senão
        Retorne(b)
    FimSe
Fim
```

Algoritmo que utiliza a função maior2:

```
Início
    Real: n1, n2, m
    Leia(n1, n2)
    m = maior2(n1, n2)
    Escreva('Maior:', m)
Fim
```

Exercício 2

Faça uma função que retorna o maior de 3 números passados como parâmetros.

Exercício 3

Faça uma função chamada `divi` que receba um número e retorne quantos divisores esse número possui. Exemplo:
`divi(5)` deve retornar 2
`divi(10)` deve retornar 4

Exercício 4

Faça uma função chamada `primo` que receba um número e retorne se o número é primo ou não. Utilize a função `divi` para calcular o número de divisores do número. Exemplo:
`primo(5)` deve retornar `primo`
`primo(10)` deve retornar `não primo`