

Aula 8

Conceitos de Função e modularização

Seiji Isotani, Rafaela V. Rocha

sisotani@icmc.usp.br

rafaela.vilela@gmail.com

PAE: Armando M. Toda, Geiser Chalco

armando.toda@gmail.com

geiser.gcc@gmail.com

Conceitos de função e modularização



Você já percebeu que:

- À medida que aumenta o **tamanho** de um programa fica mais difícil de **gerenciar**
- As vezes existem muitas **repetições** no código e que trechos do programa podem ser **reutilizados** em diferentes lugares
- É possível **testar partes** do programa em separado
- As vezes conseguimos **resolver parte** do problema, mas não ele todo

Exemplo

Dado três inteiros crie um algoritmo para retornar o menor deles

USE APENAS (SE-ENTÃO)

Resposta

LEIA n1, n2, n3

SE (n1 <= n2) ENTÃO

SE (n1 <= n3) ENTÃO

IMPRIME n1

SENÃO

IMPRIME n3

SENÃO // n1 > n2

SE (n2 <= n3) ENTÃO

IMPRIME n2

SENÃO

IMPRIME n3

Resposta

LEIA n1, n2, n3

SE (n1 <= n2) ENTÃO

SE (n1 <= n3) ENTÃO

IMPRIME n1

SENAO

IMPRIME n3

SENAO // n1 > n2

SE (n2 <= n3) ENTÃO

IMPRIME n2

SENAO

IMPRIME n3

Resposta

LEIA n1, n2, n3

SE n1 \leq n2 ENTÃO

imprimeMenor(n1, n3)

SENAO

imprimeMenor(n2, n3)

FUNCAO imprimeMenor(x, y)

SE x \leq y ENTÃO

IMPRIME x

SENAO

IMPRIME y

Resposta

LEIA n1, n2, n3, menor

SE n1 \leq n2 ENTÃO

menor \leftarrow imprimeMenor(n1, n3)

IMPRIME menor

SENAO

menor \leftarrow imprimeMenor(n2, n3)

IMPRIME menor

FUNCAO imprimeMenor(x, y)

SE x \leq y ENTÃO

DEVOLVE x

SENAO

DEVOLVE y

Exemplo:

- Para modularizar o código utiliza-se o conceito de função (ou procedimento)
- Uma função é um conjunto de instruções com entradas e saídas bem definidas
 - Nome
 - Entradas
 - Instruções que executam a tarefa
 - Instruções que devolvem o resultado (saídas)

Definição

/****** Comentário *****/

- O que a função faz?
- Quais as entradas
- Quais as saídas

*****/

argumentos

FUNCAO nomeDaFuncao(parâmetro1, parâmetro2, ...)

DECLARE variável1, variáveln

Variáveis locais
Acessíveis somente
dentro da função

instrução1

instruçãon

.....

Devolve variáveln;

Valor de retorno

corpo

Vantagens das funções

- **Modularização**
 - Permite dividir de forma lógica partes do código
 - Facilita a reutilização
 - ajuda a lidar com a complexidade
 - facilita a divisão de trabalho
- **Encapsulamento**
 - Esconde o que não precisa ser apresentado

Pontos importantes: variáveis

Declare x

$x \leftarrow 1$ **1**

teste()

IMPRIME x **1**

FUNCAO teste()

 Declare x

$x \leftarrow 2;$

 DEVOLVE x **2**

Pontos importantes: variáveis

Declare x

$x \leftarrow 1$ **1**

$x \leftarrow \text{teste}()$ **3**

IMPRIME x **3**

FUNCAO teste()

Declare x, i

$x \leftarrow 2$

PARA $i \leftarrow 0$ ate 1 FACA

$x \leftarrow x + i$

DEVOLVE x **3**

Pontos importantes: variáveis

Declare x

$x \leftarrow 1$ **1**

teste(x)

IMPRIME x **1**

FUNCAO teste(x) **1**

$x \leftarrow x+1$ **2**

DEVOLVE x **2**

Exercício

Crie um programa que consegue adivinhar um número entre 1 a 15 em no máximo 4 tentativas

(assuma que o jogador diga se o número é maior ou menor a cada tentativa incorreta).

USE APENAS (SE-ENTAO)