

# **Aula 6**

## **Conceitos de Função e modularização**

**Responsável**

**Prof. Armando Toda (armando.toda@usp.br)**

# Conceitos de função e modularização



## Você já percebeu que:

- À medida que aumenta o tamanho de um programa fica mais difícil de gerenciar
- Às 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
- Às vezes conseguimos resolver parte do problemas, 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

SENAO

IMPRIME n3

SENAO // n1 > n2

SE (n2 <= n3) ENTÃO

IMPRIME n2

SENAO

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

SENÃO

menor  $\leftarrow$  imprimeMenor(n2, n3)

IMPRIME menor

---

FUNÇÃO imprimeMenor(x, y)

SE x  $\leq$  y ENTÃO

DEVOLVE x

SENÃO

DEVOLVE y

# Definição

```
/****** Comentário *****/
```

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

```
*****/
```

```
FUNCAO nomeDaFuncao(parâmetro1, parâmetro2, ...)
```

```
DECLARE variável1, variáveln
```

```
    instrução1
```

```
    instruçãon
```

```
    ...
```

```
Devolve variáveln;
```

argumentos

Variáveis locais  
Acessíveis somente  
dentro da função

corpo

Valor de retorno

## 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)

# 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 int x

$x \leftarrow 1$

teste()

IMPRIME x

FUNCAO teste()

Declare int x

$x \leftarrow 2;$

DEVOLVE x

## Pontos importantes: variáveis

Declare int x

$x \leftarrow 1$

teste()

IMPRIME x

FUNCAO teste()

Declare int x

$x \leftarrow 2;$

IMPRIME x

## Pontos importantes: variáveis

Declare int x

$x \leftarrow 1$

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

IMPRIME x

FUNCAO teste()

Declare int x

$x \leftarrow 2$

PARA  $i \leftarrow 0$  ate 1 FACA

$x \leftarrow x + i$

DEVOLVE x

## Pontos importantes: variáveis

Declare int x

$x \leftarrow 1$

teste(x)

IMPRIME x

FUNCAO teste(x)

$x \leftarrow x+1$

## Exercício 1

- a) Faça um algoritmo que contenha uma função para calcular a média de dois valores
- b) Faça um algoritmo que contenha uma função para ler um vetor
- c) Faça uma função que ordene um vetor de tamanho 10 (EXTRA 0.1)