

# Introdução à Ciência da Computação

## Aula de Exercícios

Prof. Ricardo J. G. B. Campello  
Estagiário PAE: Bilzã M. de Araújo

## Créditos

- ◆ Os exercícios dos slides a seguir foram adaptados dos originais de A. L. V. Forbellone e H. F. Eberspächer.

## Aula de Hoje

- ◆ Iremos exercitar a elaboração de algoritmos
  - Estrutura sequencial
  - Estruturas de seleção
  - Estruturas de repetição
- ◆ Teste de mesa
- ◆ Tirar dúvidas sobre outros tópicos
  - 10 minutos finais

## Estrutura sequencial

- ◆ Um índio precisa atravessar um rio com uma canoa que possui capacidade para carregar apenas ele mesmo e mais uma carga: um lobo, um bode ou um maço de alfafa.
- ◆ O quê o índio deve fazer para conseguir atravessar o rio sem prejuízos? Escreva um algoritmo mostrando cada uma das ações do índio.

## Estrutura sequencial



A



B

## Estrutura sequencial

Solução:

**início**

- leva bode de A para B;
- vai de B para A;
- leva lobo de A para B;
- leva bode de B para A;
- leva alfafa de A para B;
- vai de B para A;
- leva bode de A para B;

**fim.**

## Estrutura sequencial

Solução Alternativa:

**início**

leva bode de A para B;  
vai de B para A;  
leva alfafa de A para B;  
leva bode de B para A;  
leva lobo de A para B;  
vai de B para A;  
leva bode de A para B;

**fim.**

## Estruturas de seleção

- ❖ O IMC é um critério da OMS para dar uma indicação sobre a condição de peso de uma pessoa adulta.

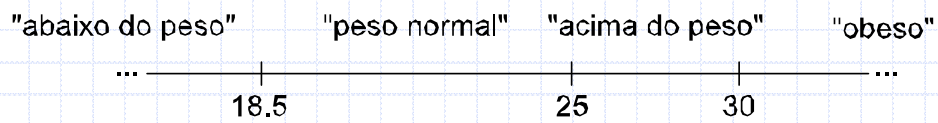
$$IMC = \frac{\textit{peso}}{\textit{altura}^2}$$

- ❖ Elabore um algoritmo que leia o peso e a altura de um adulto e mostra a sua condição.

IMC em adultos	Condição
abaixo de 18,5	abaixo do peso
entre 18,5 e 24,9	peso normal
entre 25 e 29,9	acima do peso
a partir de 30	obeso

## Estruturas de seleção

IMC em adultos	Condição
abaixo de 18,5	abaixo do peso
entre 18,5 e 24,9	peso normal
entre 25 e 29,9	acima do peso
a partir de 30	obeso



## Estruturas de seleção

Solução:

```
início
  real: peso, altura, IMC;
  leia(peso, altura);
  IMC <- peso/pot(altura, 2);
  se IMC < 18.5 então
    escreva("abaixo do peso");
  fimse;
  se IMC >= 18.5 e IMC < 25 então
    escreva("peso normal");
  fimse;
  se IMC >= 25 e IMC < 30 então
    escreva("acima do peso");
  fimse;
  se IMC >= 30 então
    escreva("obeso");
  fimse;
fim.
```

## Estruturas de seleção

Solução Melhorada:

```
início
  real: peso, altura, IMC;
  leia(peso, altura);
  IMC <- peso/pot(altura, 2);
  se IMC < 18.5 então
    escreva("abaixo do peso");
  senão se IMC < 25 então
    escreva("peso normal");
  senão se IMC <= 30 então
    escreva("acima do peso");
  senão
    escreva("obeso");
  fimse;
fimse;
fimse;
fim.
```

## Estruturas de repetição

- ◆ Elabore um algoritmo que, dado um número como entrada, verifique se este número é ou não um número primo. Um número primo é divisível apenas por 1 e por ele mesmo. Ex: 1, 2, 3, 5, 7, 11, 13, 17, 23, ...
- ◆ Dica: Se um número for divisível por qualquer número que não seja 1 ou ele mesmo, então este número não é primo. Ex: 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 19, 20, ...
- ◆ Estratégia: Podemos dividir o número por todos os números entre 2 e número div 2. Se alguma divisão tiver resto = 0, então o número não é primo.

## Estruturas de repetição

É necessário utilizar uma estrutura de repetição:

```
enquanto <condição> faça
  C1;

  Cn;
fimenquanto;

repita
  C1;

  Cn;
até <condição>

para V de vi até vf passo p faça
  C1;

  Cn;
fimpara;
```

## Estruturas de repetição

Solução usando **enquanto**:

```
início
  inteiro: numero, divisor;
  lógico: eh_primo;
  leia(numero);
  eh_primo <- V;
  divisor <- 2;
  enquanto divisor <= numero div 2 e eh_primo faça
    se numero mod divisor = 0 então
      eh_primo <- F;
    fimse;
    divisor <- divisor + 1;
  fimenquanto;
  se eh_primo então
    escreva("é primo")
  senão
    escreva("não é primo")
  fimse;
fim.
```

## Estruturas de repetição

Solução usando **repita**:

```
início
  inteiro: numero, divisor;
  lógico: eh_primo;
  leia(numero);
  eh_primo <- V;
  divisor <- 2;
  repita
    se numero mod divisor = 0 então
      eh_primo <- F;
    fimse;
    divisor <- divisor + 1;
  até divisor > numero div 2 ou não eh_primo;
  se eh_primo então
    escreva("é primo")
  senão
    escreva("não é primo")
  fimse;
fim.
```

## Estruturas de repetição

Solução usando **para**:

```
1. início
2.   inteiro: numero, divisor;
3.   lógico: eh_primo;
4.   leia(numero);
5.   eh_primo <- V;
6.   para divisor de 2 até numero div 2 passo 1 faça
7.     se numero mod divisor = 0 então
8.       início
9.         eh_primo <- F;
10.        divisor <- numero; //a execução sairá do para,
11.        fim.              //pois V será maior que vf
12.      fimse;
13.    fimpara;
14.    se eh_primo então
15.      escreva("é primo")
16.    senão
17.      escreva("não é primo")
18.    fimse;
19. fim.
```



## Teste de mesa

- ◆ <http://www.brasilacademico.com/ed/testemesa.htm>

- ◆ Exemplo no quadro.

Dúvidas?

## Outros Exercícios

◆ Realize a conversão Decimal -> Binário -> Decimal

- Caractere 'M' (ASCII: 77)

◆ Calcule o resultado das seguintes expressões:

- não  $Y * 7 \text{div} 2 \leq 47 - \text{pot}(2, 3) \text{mod} 3$  e  $\text{rad}(13 - 4) - 3 * 2 > Z$

- não  $(8 < 24 / 3$  ou  $\text{rad}(25) = 5$  e  $15 - 5 > 5 * 2$  ou  $V$ )