

Bacharelado em Sistemas de Informação

Algoritmos - Aula 11

Prof. Dr. Sérgio Luis Antonello

Prof. Me. Antero Sewaybricker Todesco

Bibliografia básica desta aula

Medina & Fertig (2006). Capítulo 07

Forbellone & Eberspacher (2000). Capítulo 04

Ascencio & Campos (2002). Capítulo 05

19/05/2020

Plano de Ensino: agenda

15	19/05/2020	Variáveis indexadas: declaração e manipulação de vetor.
16	26/05/2020	Variáveis indexadas: declaração e manipulação de matriz.
17	02/06/2020	Semana Científica do Curso.
18	09/06/2020	AVALIAÇÃO: Prova 2.
19	16/06/2020	Devolutiva da Prova 2. TRABALHO: Entrega do trabalho A2.
20	23/06/2020	AVALIAÇÃO: SUB
21	30/06/2020	Devolutiva da SUB.

OBS: Trabalho em grupo

- Arquivo txt com nomes e RAs (19/05)
- deadline 16/06/2020 (19h)

Plano de ensino: avaliação

$$\text{Nota Final} = (A1 + (2 \times A2)) / 3$$

Composição da nota A2

- ❑ 70% por uma prova individual (P2) +
 - ❑ 10% Atividades práticas (tarefas e atividades) +
 - ❑ 20% Atividade prática em grupo (trabalho A2)
-

Plano de ensino: conteúdo

1. Unidade I – Introdução a algoritmos (objetivos a, b, c)
 - 1.1. Conceitos de abstração de dados
 - 1.2. Lógica de programação
 - 1.3. Algoritmos
 - 1.4. Formas de representação de algoritmos: pseudocódigo e fluxograma.
 - 1.5. Teste de mesa
 - 1.6. Tipos de dados
 - 1.7. Constantes e variáveis
 - 1.8. Atribuição
 - 1.9. Operadores e precedência
 - 1.10. Expressões aritméticas, relacionais e lógicas.
2. Unidade II – Estruturas básicas de controle (objetivos c, d, e)
 - 2.1. Blocos de comando
 - 2.2. Estruturas de decisão
 - 2.3. Estruturas de repetição
 - 2.4. Aninhamento
3. Unidade III – Modularização (objetivos c, d, e)
 - 3.1. Dividir para conquistar
 - 3.2. Procedimentos e funções
 - 3.3. Escopo de variáveis
 - 3.4. Parâmetros e argumentos
 - 3.5. Passagem de parâmetros por valor e por referência
4. Unidade IV – Estruturas de dados homogêneas (objetivos d, e)
 - 4.1. Vetor
 - 4.2. Matriz

Sumário

Primeiro momento: revisão

- ✓ Passagem de parâmetros por valor
- ✓ Passagem de parâmetros por referência
- ✓ Correção de exercícios

Segundo momento

- ✓ Variáveis indexadas
 - ✓ Vetor
 - ✓ Matriz

Terceiro momento: síntese

1. Revisão: Passagem de Parâmetros

Funcao SeraQueMudaValor(A: inteiro; **Var** B: inteiro) : inteiro

Var

valorcalculado: inteiro

Inicio

Escreval("A subrotina está sendo executada nesse momento")

Escreval("Valores recebidos")

Escreval("Valor de A: ", A)

Escreval("Valor de B: ", B)

A <- 10

B <- 20

valorcalculado <- A + B

Escreval("Novo valor de A: ", A)

Escreval("Novo valor de B: ", B)

Escreval("A subrotina está sendo finalizada")

Escreval()

Retorne valorcalculado

FimFuncao

ALGORITMO "Exemplo"

Var

V1, V2, R: inteiro

Inicio

Leia(V1)

Leia(V2)

Escreval("Valores digitados ", V1, V2)

R <- **SeraQueMudaValor**(V1, V2)

Escreval("Valores após a execução da subrotina")

Escreval("Valor de V1: ", V1)

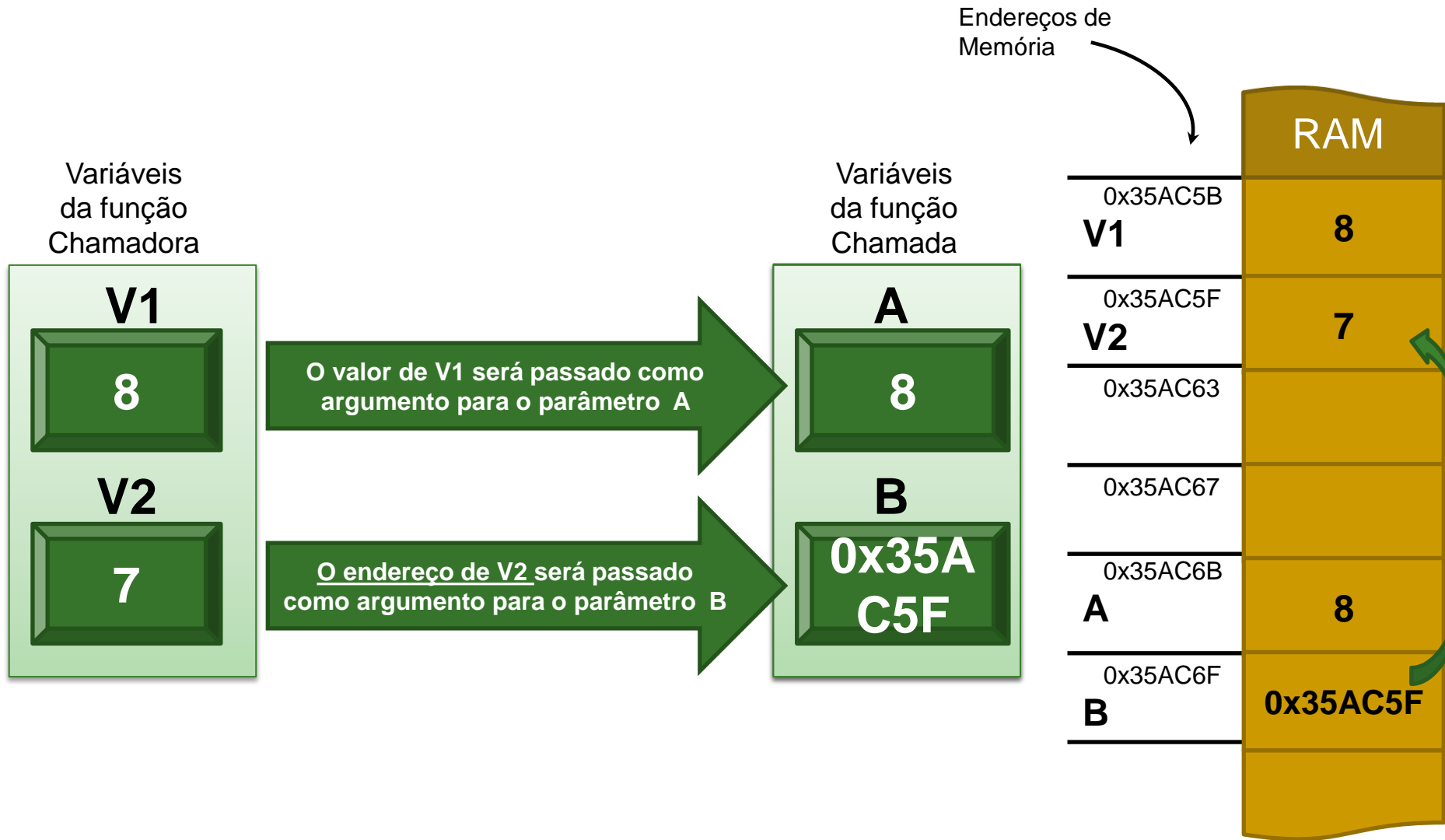
Escreval("Valor de V2: ", V2)

Escreval("Valor de R: ", R)

FimAlgoritmo

Programa principal

1. Revisão: Passagem de Parâmetros por Referência



1. Primeiro momento:

Correção de exercícios

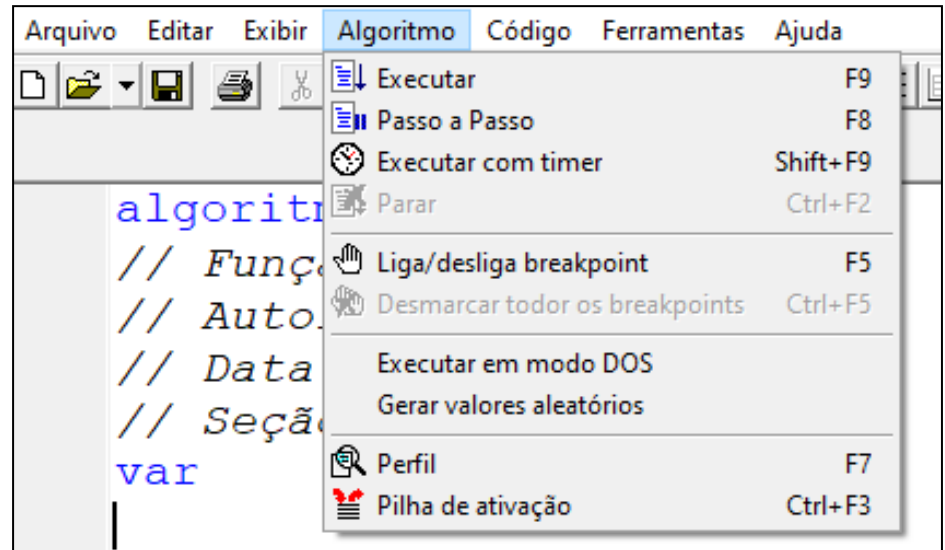
2. Segundo momento: Motivação

- ❑ Uma variável declarada como inteiro constitui-se de um espaço na memória quem permite armazenamento de exatamente um valor do tipo inteiro. O mesmo ocorre com os demais tipos. Assim, elas são chamadas de variáveis simples.
 - ❑ Se um algoritmo necessita manipular a nota de um aluno, sabe-se que a nota é do tipo real, então, é possível utilizar uma **variável simples**.
 - ❑ Mas, se o algoritmo precisa manipular **várias notas** de um mesmo aluno?
 - ❑ Ou ainda, se o algoritmo precisa manipular **várias notas de vários alunos**?
-

2. Segundo momento: Motivação



designed by freepik



100, 2, 1, 3, 100

41

42

43

44

45

46

47

48

49

50

2. Segundo momento

❑ Variáveis indexadas

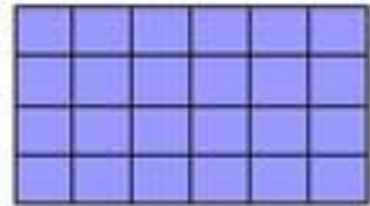
❑ Vetor

❑ Matriz

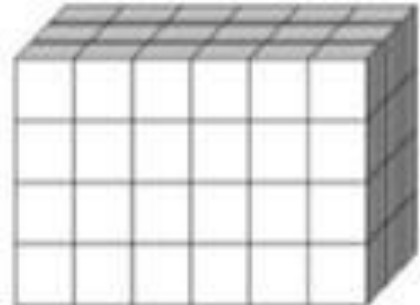
■ Vetor



■ Matriz 2D



■ Matriz 3D



2. Variável indexada

- ❑ Se o algoritmo precisa manipular várias notas do aluno, em vez de declarar muitas variáveis, uma para cada nota, é possível declarar uma variável indexada.
- ❑ Também é conhecida como arranjo ou array.
- ❑ Se constitui de um conjunto de espaços de memória.
- ❑ Cada espaço deste conjunto possibilita armazenar um valor de cada vez.
- ❑ Todos os valores armazenados em uma variável indexada devem ser de mesmo tipo.

3. Variável indexada: vetor

Vetor (array unidimensional) é uma variável que armazena várias variáveis do mesmo tipo.

Vetor de nomes dos alunos

1	2	3	...	49	50
João	Pedro	Carlos	...	José	Maria

3. Vetor: declaração

- ❑ A declaração deve conter o nome da variável, o tipo de dados e a quantidade de valores a serem armazenados.
- ❑ Nota: Vetor[1..5] de Real

Nome da variável	Nota				
Índice (posição)	1	2	3	4	5
Conteúdo	8.5	2.0	9.5	10.0	6.5

3. Vetor: manipulação

- ❑ A atribuição de uma determinada nota no respectivo espaço de memória deve ser realizada com uso do índice, como exemplo:
 - ✓ `Nota[1] <- 8.5`
 - ✓ `Nota[2] <- 2.0`
 - ✓ `Nota[3] <- 7.8`
 - ✓ `Leia (Nota[4])`
 - ✓ `Leia (Nota[5])`
- ❑ Porém, se a quantidade de elementos de uma variável indexada for muito grande, torna-se inviável a manipulação dos elementos.

3. Vetor: manipulação

- ❑ Para facilitar a manipulação de variáveis indexadas com grande quantidade de elementos utiliza-se estruturas de repetição.
- ❑ Usa-se uma variável como índice.

Exemplo:

Leia (Nota[1])

Leia (Nota[2])

Leia (Nota[3])

Leia (Nota[4])

Leia (Nota[5])

Para **ind** de 1 Ate 5 Passo 1 Faca

 Leia (Nota[**ind**])

FimPara

Vamos Programar!



5. Exercícios com vetor



- 1) Desenvolver um algoritmo para ler 5 valores numéricos inteiros e armazene em um vetor.

Em seguida deve ser preenchido um segundo vetor, com os mesmo valores do primeiro, só que em ordem inversa. O primeiro elemento de um vetor será o último do outro vetor. O segundo elemento de um será o penúltimo do outro e assim por diante.

- 2) Desenvolver um algoritmo para ler 10 valores numéricos do tipo real. Como resultado do processamento exiba quantos e liste quais elementos estão acima da média calculada entre eles.
-

5. Exercícios com vetor



- 3) Declarar três variáveis indexadas A, B e C com capacidade para armazenar, cada uma, cinco elementos do tipo inteiro. Solicitar a entrada de dados para os elementos de A e B. Fazer cada elemento de C igual aos respectivos elementos de A multiplicado por B. Exibir todos os valores de A, B e C.

 - 4) Desenvolver um algoritmo que possibilite armazenar em uma variável indexada sete valores numéricos do tipo inteiro. Determinar qual é o menor valor armazenado e indicar qual posição da variável ele ocupa.
-