



LÓGICA DE PROGRAMAÇÃO AULA 5

+55 (21) 99461-8818

@explicadoresnet

www.explicadores.net.br



Estruturas de Dados Homogêneas

+55 (21) 99461-8818

@explicadoresnet

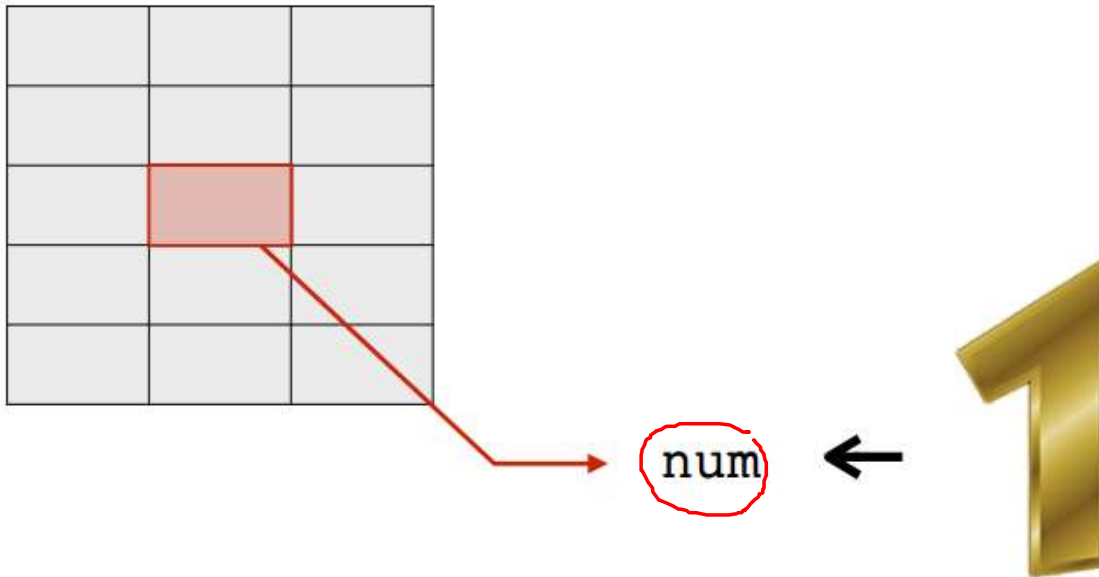
www.explicadores.net.br



Variáveis Compostas Homogêneas

VARIÁVEIS SIMPLES
ESCALARES

As **variáveis simples** permitem guardar **apenas um valor** por vez.



$$A[3+1] = 8$$

Variáveis Compostas Homogêneas

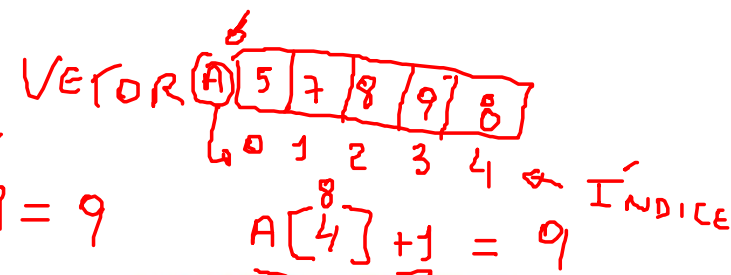
As **variáveis compostas** permitem guardar **vários valores** por vez, um em cada espaço.

[0]	[1]	[2]

- ✓ **variáveis indexadas**
- ✓ **variáveis subscritas**
- ✓ **arranjos**
- ✓ **vetores**
- ✓ **matrizes**
- ✓ **tabelas em memória**
- ✓ **arrays**

num [1] ←

4



	0	1	2	3
0	1	15	17	19
1	2	11	7	13
2	4	6	9	9
3	3	2	1	0

MATRIZES

$$B[1, 2] = 11$$

AB	5	6	7	1	0	2	3
	0	1	2	3	4	5	6

$$AB[] ? \textcircled{7}$$

$$* AB[2] + AB[4] = \textcircled{7}$$

$$* AB[\overset{2}{\underset{5}{3+2}}] + AB[\overset{7}{\underset{2}{4-2}}] = \textcircled{9}$$

$$* AB[5] + 6 = \textcircled{8}$$

	0	1	2
CA 0	5	4	2
1	4	3	1
2	3	1	2

$$CA[][] = ? \textcircled{9}$$

$$3 \times 3 =$$

$$\overset{2}{\underset{LC}{CA}}[0,2] + \overset{1}{CA}[1,2] = 3$$

$$CA[\underset{1}{1}, \underset{2}{2}] + CA[2,2] = \textcircled{3}$$

$$CA[\underset{2}{1+1}, \underset{2}{0+2}] + 10 = \textcircled{12}$$

$$AB[5] * \overset{CA}{\underset{2}{\cancel{AB}}}[1,1] = \textcircled{6}$$



Fundamentos

```
Algoritmo Composto
var
  N: vetor[0..3] de Inteiro
início
  Escreva("Informe um valor")
  Leia(N[0])
  Escreva("Informe um valor")
  Leia(N[1])
  Escreva("Informe um valor")
  Leia(N[2])
  Escreva("Informe um valor")
  Leia(N[3])
FimAlgoritmo
```

Diagram annotations: A blue arrow points from the word "dimensão" to the range "0..3" in the array declaration. Red checkmarks are placed above the "var" and "N" declarations. Red circles highlight the indices "0" and "3" in the array declaration and the "4" in the final array visualization. A red bracket is placed to the right of the loop body.

N

5	7	8	4
0	1	2	3

Diagram annotations: Red checkmarks are placed below the indices 0, 1, 2, and 3, and to the right of the array.

Fundamentos

variável simples

→ $\frac{I}{0 \ 4 \ 8 \ 8}$
1 1 1 1

composta

→ $\begin{matrix} N[0] \\ N[1] \\ N[2] \\ N[3] \end{matrix}$

Algoritmo Composto

var

$\textcircled{N}$: vetor[0..3] de Inteiro

i: Inteiro

inicio

Para i ← 0 **ate** 3 **faca**

Escreva("Informe um valor")

Leia(N[i])

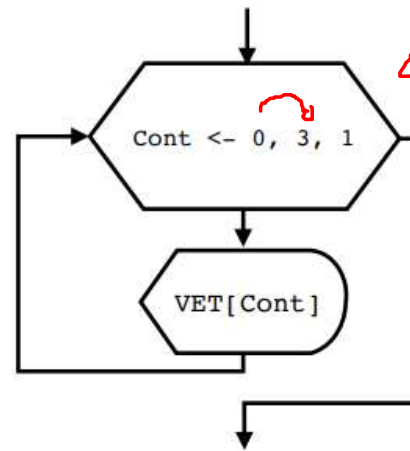
FimPara

FimAlgoritmo

→

N			
5	7	8	9
0	1	2	3

Exibição da Estrutura



VET

6	4	5	7
0	1	2	3

Variáveis Compostas Homogêneas Multidimensionais



num [1 , 2] ←

4



EXPLICADORES.NET

Fundamentos

L x C

dimensão? 12

```
Algoritmo Composto
var
  M: vetor[0..2, 0..3] de Inteiro
  L, C: Inteiro
inicio
  Para L <- 0 ate 2 faca
    Para C <- 0 ate 3 faca
      Escreva("Informe um valor")
      Leia(M[L,C])
    FimPara
  FimPara
FimAlgoritmo
```

L | C
0 | 0
1 | 1
2 | 2
3 | 3
4 | 4
5 | 5
6 | 6
7 | 7
8 | 8
9 | 9
10 | 10
11 | 11
12 | 12

M	0	1	2	3
0	8	7	5	7
1	5	8	3	5
2	3	5	6	8

Isolando uma Linha

```
Algoritmo Composto
var
  M: vetor[0..2, 0..3] de Inteiro
  L, C: Inteiro
inicio
  Para C <- 0 ate 3 faca
    Escreva(M[1,C])
  FimPara
FimAlgoritmo
```

M

0				
1				
2				
	0	1	2	3

Isolando uma Coluna

Algoritmo Composto

var

M: vetor[0..2, 0..3] de Inteiro

L, C: Inteiro

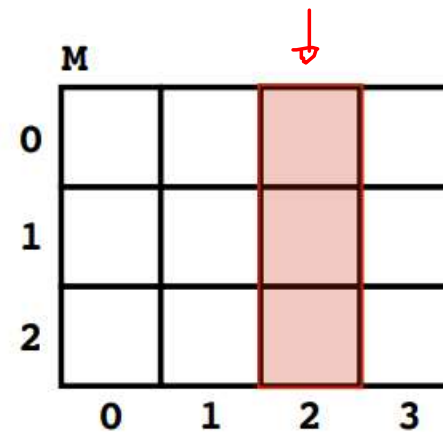
inicio

Para L <- 0 ate 2 faça

Escreva(M[L,2])

FimPara

FimAlgoritmo



Isolando a Diagonal Principal

Algoritmo Composto

var

M: vetor[0..2, 0..2] de Inteiro

L, C: Inteiro

inicio

Para C <- 0 ate 2 faca

Escreva(M[C,C])

FimPara

FimAlgoritmo

0,0
1,1
2,2

M

0	1	2
0	1	2
1	1	2
2	1	2



– Considerando a matriz bidimensional TABELA [1..8,1..5], assinale a afirmativa correta.

L C
8, 5
x

= 40
ELEMENTOS

- a) A matriz tem 5 ~~linhas~~, 8 colunas e 40 elementos.
- b) A matriz tem 8 linhas, 5 colunas e no ~~mínimo~~ 40 elementos.
- c) A matriz tem 5 ~~linhas~~, 8 colunas e podem ser armazenados até 40 elementos.
- ~~d)~~ A matriz tem 8 linhas, 5 colunas e podem ser armazenados até 40 elementos.

EAGS

– Dado o seguinte trecho de programa em Português Estruturado,

$J \leftarrow 2$

$A[1] \leftarrow 5$

enquanto ($J < 8$) **faça**

$A[J] \leftarrow A[J - 1] + 3$

$J \leftarrow J + 1$

fim_enquanto

assinale a alternativa que contém o valor correto de $A[6]$.

- a) 5
- b) 8
- c) 17
- ~~d) 20~~



Handwritten calculation: $5 + 3 \times 5 = 20$

- No algoritmo do programa abaixo, em português estruturado, temos as seguintes entradas para os valores da variável X quando requisitados: $X[1] = 2$, $X[2] = 3$, $X[3] = 8$, $X[4] = 3$. Identifique nas respostas abaixo qual será a saída do programa.

programa CONTA

var

RESULTADO : real

X : conjunto[1..4] de real

T : inteiro

Y : inteiro

início

Y $\leftarrow$ 4

RESULTADO $\leftarrow$ 0

para T de 1 até 4 passo 1 faça

leia X[T]

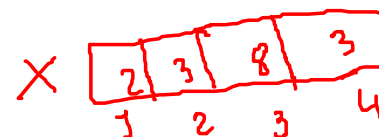
RESULTADO $\leftarrow$ RESULTADO + X[T]

fim_para

RESULTADO $\leftarrow$ RESULTADO / Y

escreva RESULTADO

fim



$$\frac{Y}{4}$$

RESULTADO
2 3 8 3 16

a) 2

b) 8

c) 3

~~d) 4~~



EXPLICADORES.NET

Variáveis Compostas Heterogêneas

As **variáveis compostas heterogêneas** ou **registros** permitem definir o **tipo primitivo** de cada posição.

```
Algoritmo Composto
tipo
  Ficha = Registro
    cod: Inteiro
    nome: Caractere
    peso: Real
  FimRegistro
var
  Pessoa: Ficha
início
  Leia (Pessoa.cod)
  Leia (Pessoa.nome)
  Leia (Pessoa.peso)
FimAlgoritmo
```

Handwritten annotations: Red arrows point from "tipo" to "Ficha = Registro" and from "Registro" to "FimRegistro". A red checkmark is next to "Ficha". The word "Tipo" is written in red above "Registro".

Pessoa

cod	01
nome	ALEX
peso	65 kg

Registros de Vetores

```
Algoritmo Composto
tipo
  Bimestre = vetor[1..4] de Real
  Aluno = Registro
    mat: Inteiro
    nome: Caractere
    notas: Bimestre
  FimRegistro
var
  MeuAluno: Aluno
inicio
  Leia (MeuAluno.nome)
  Leia (MeuAluno.notas[1])
  Leia (MeuAluno.notas[3])
FimAlgoritmo
```

MeuAluno

Mat									
nome	—								
<u>notas</u>	<table><tr><td>8.0</td><td>1</td><td>7.0</td><td>1</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	8.0	1	7.0	1	1	2	3	4
8.0	1	7.0	1						
1	2	3	4						

Vetores de Registros

Algoritmo Composto

tipo

Ficha = Registro

{
 cod: Inteiro
 nome: Caractere
 peso: Real

FimRegistro

var

Cadastro: vetor[1..3] de Ficha

inicio

 Leia (Cadastro[1].nome)

 Leia (Cadastro[2].peso)

FimAlgoritmo

Cadastro

cod		cod		cod	
nome	—	nome		nome	
peso		peso	—	peso	
1		2		3	

Pesquisa Sequencial

Consiste em efetuar a busca da informação desejada **a partir do primeiro elemento** sequencialmente **até o último**.

Localizando a informação no caminho, ela é apresentada. Este método de pesquisa é **lento**, porém **eficiente** caso o vetor encontra-se com seus elementos **desordenados**.

(A)	1	5	4	15	25	35	3	18	20	21
-----	---	---	---	----	----	----	---	----	----	----

Pesquisa Binária

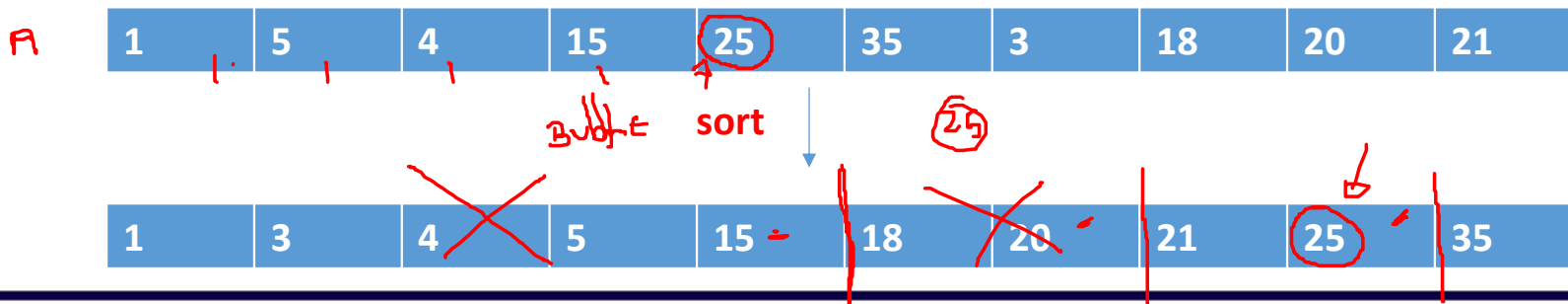
Divide o vetor em **duas partes** e procura saber se a informação a ser pesquisada está **acima ou abaixo** da linha de divisão.

Se estiver acima, toda a metade abaixo é **desprezada**.

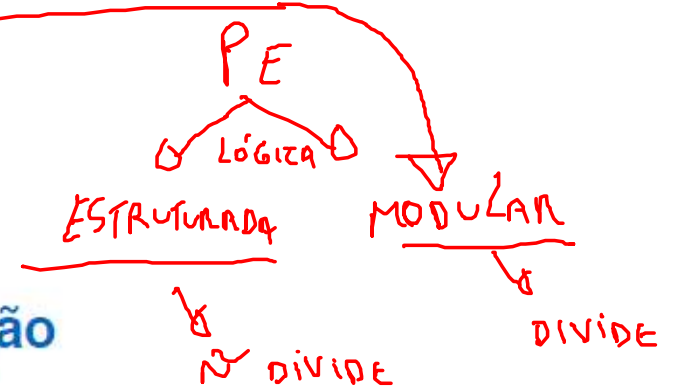
E assim vai sendo executada até **encontrar ou não** a informação pesquisada.

Esse método é em média mais **rápido** que o sequencial, porém exige que a matriz esteja **previamente classificada**.

ORGANIZADA
SORT



Método Top-Down



O método mais adequado para a programação estruturada.



Antes de iniciar o programa, deverá ter em mente as **tarefas principais** que este deverá executar.

Conhecidas todas as tarefas, imaginar como deverá ser o programa principal, que vai controlar todas as outras tarefas.

Tendo definido o programa principal, é iniciado o **processo de detalhamento**. Desta forma são definidos vários algoritmos, um para cada rotina separada.



Método Top-Down

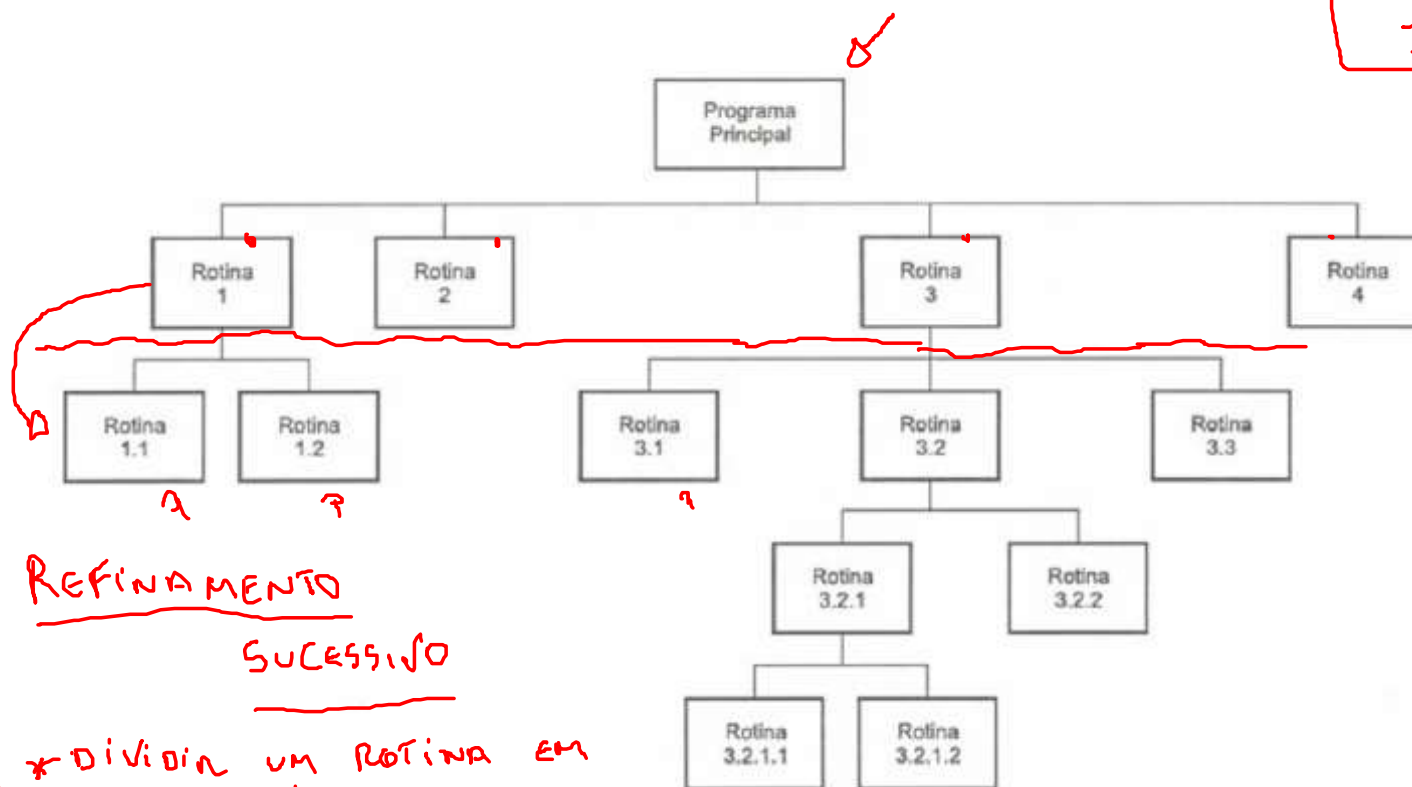
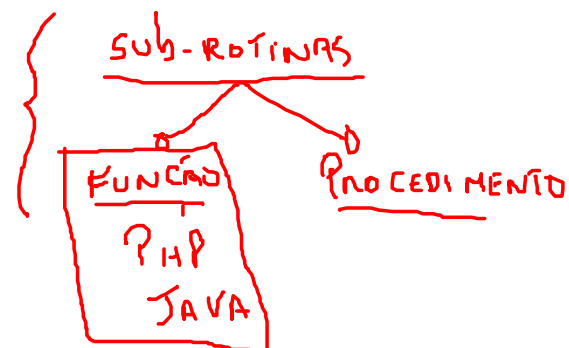
Rotinas
Sub-Rotinas
Módulos

O método **Top-Down** faz com que o programa tenha uma estrutura semelhante a um organograma.

A utilização do método “**de cima para baixo**” permite que seja efetuado cada módulo do programa em separado.

Outro detalhe a ser considerado é que muitas vezes existem em um programa trechos de códigos que são repetidos várias vezes. Esses trechos poderão ser utilizados como sub-rotinas, proporcionando um programa menor e mais fácil de ser alterado num futuro próximo.

Método Top-Down



REFINAMENTO

SUCCESSIVO

* Dividir um rotina em sub rotinas