

AULA DE ALGORITMO - 04

Profa. M. Sc. Valéria Maria Volpe

Variáveis Estruturas

2

□ Variáveis Estruturadas

- ▣ As variáveis usadas em nossos algoritmos/programas, até agora, são variáveis simples, uma vez que armazenam apenas **um** valor, sempre o último que foram atribuídos a elas.

▣ Exemplos:

inteiro: X;

real: Salario;

Variáveis Estruturas

3

□ Variáveis Estruturadas

- ▣ As variáveis estruturadas são variáveis que permitirão armazenar **um conjunto de valores**.
- ▣ As variáveis estruturadas são:
 - Homogêneas
 - Unidimensional – vetor
 - Multidimensional – matriz
 - Heterogênea
 - Registro

Variáveis Estruturas

4

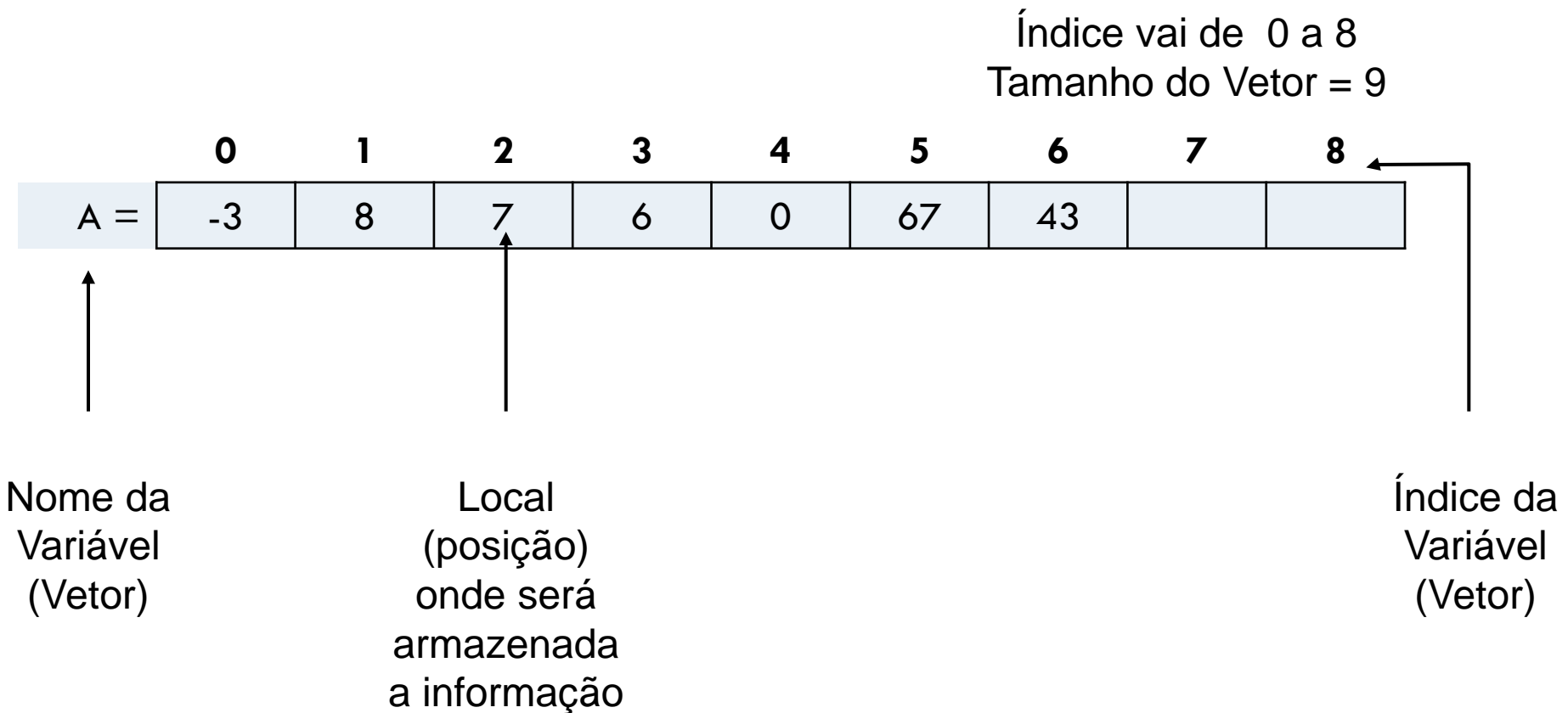
□ Variável Estruturada Homogênea

- São variáveis que armazenam **um conjunto** de informações (estrutura), todas as informações são de **mesmo tipo** (homogênea).
- As variáveis unidimensionais (Vetor) utilizam apenas **um índice** para acessar a informação, já as variáveis multidimensionais (Matriz) utilizam **mais de um índice** para acessar a informação.
- O **índice** é um número inteiro positivo que se **inicia no 0 (zero) e vai até N-1**, para um vetor de **tamanho N**.

Variáveis Estruturas

5

□ Representação de Vetor



Variáveis Estruturas

6

□ Declaração de Vetor - Sintaxe Algoritmo

NomeVetor: vetor [tam] de tipo primitivo;

- Sabendo que **tam** é o tamanho máximo do vetor, ou seja, quantidade máxima de informações que ele pode armazenar.

□ Exemplo

A: vetor [9] de inteiro; A =

0	1	2	3	4	5	6	7	8
3	7	-9	1					

X: vetor [20] de real; X =

0	1	2	3	4	5	6	7	...	19
1.7	3.0	-2.3	2.1	-9.8	0.8				

Variáveis Estruturas

7

- Declaração de Vetor
- Sintaxe – Linguagem C

tipo primitivo NomeVetor [**tam**];

- Sabendo que **tam** é o tamanho máximo do vetor, ou seja, quantidade máxima de informações que ele pode armazenar.

- **Exemplo**

int A[9]

	0	1	2	3	4	5	6	7	8
A =	3	7	-9	1					

float X[20]

	0	1	2	3	4	5	6	7	...	19
X =	1.7	3.0	-2.3	2.1	-9.8	0.8				

Variáveis Estruturas

8

- Para acessar a informação armazenada no vetor usa-se o nome do vetor mais o índice.

	0	1	2	3	4	5	6	7	8
A =	-3	8	7	6	0	67	43		

- Exemplo:

A [2] → 7

A [5] → 67

- Deve-se acessar uma informação de cada vez, nunca o vetor todo.
- Para armazenar a informação no vetor deve-se começar pela posição ZERO, depois 1, 2, 3, ..., até a posição N-1.
- Não é necessário preencher todo o vetor, porém deixe os espaços vazios no final dele.

Variáveis Estruturas

9

□ Exemplo

1) Faça um algoritmo para ler um vetor de 1000 n° inteiros. Calcule e mostre quantos n° pares pertencem ao vetor.

▣ Ações:

- Ler todos os 1000 números e armazenar em um vetor
- Contar quantos pares pertencem ao vetor
- Mostrar todos os números armazenados no vetor e os demais resultados

▣ Variáveis:

- Vetor
- ContPar
- Índice

Variáveis Estruturas

10

Algoritmo vetor;

Início

var

inteiro: ContPar, I;

A: vetor [1000] de inteiro;

para I \leftarrow 0 até 999 passo 1 faça

início

escreva ("Digite um n° inteiro: ");

leia (A [I]);

fim para;

// Continua no próximo slide

Variáveis Estruturas

11

```
ContPar ← 0;  
para I ← 0 até 999 passo 1 faça  
  início  
    se A [ I ] MOD 2 = 0 então  
      início  
        ContPar ← ContPar + 1 ;  
      fim se;  
  fim para;
```

// Continua no próximo slide

Variáveis Estruturas

12

```
para I ← 0 até 999 passo 1 faça  
  início
```

```
    escreva (A [ I ] );
```

```
  fim para;
```

```
  escreva ("O vetor armazena ", ContPar, " números pares");
```

```
Fim.
```

Variáveis Estruturas

13

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    int  ContPar = 0, A [1000],  I;
    for (I = 0; I <= 999; I = I + 1)
    {
        printf("\n Digite um n° inteiro: ");
        scanf("%d", & A [ I ]);
    }
    // Continua no próximo slide
}
```

Variáveis Estruturas

14

```
for( I = 0; I <= 999; I = I + 1)
{
    if ( A [ I ] % 2 == 0)
    {
        ContPar = ContPar + 1 ;
    }
}
```

// Continua no próximo slide

Variáveis Estruturas

15

```
for (I = 0; I <= 999; I = I + 1)
{
    printf("    %d    ", A [ I ] );
}
printf("\n\n O vetor armazena %d    números pares \n", ContPar);
system("pause");
}
```

Variáveis Estruturas

16

- 2) Faça um algoritmo que receba 50 números e armazene-os em um vetor. Calcule e mostre a soma dos números desse vetor.
- 3) Faça um algoritmo que leia dois vetores de 150 posições e faça a multiplicação dos elementos de mesmo índice, colocando o resultado em um terceiro vetor.
- 4) Faça um algoritmo que receba o nome de 80 clientes e armazene-os em um vetor. Em um segundo vetor armazene a quantidade de DVD's locados em 2009 por cada um dos 80 clientes. Sabe-se que para cada dez locações o cliente tem direito a uma locação grátis. Faça um algoritmo que mostre o nome de todos os clientes com a quantidade de locações grátis a que ele tem direito.
- 5) Faça um algoritmo que carregue três vetores com dez elementos cada um. O primeiro vetor com os nomes de dez produtos. O segundo vetor com os códigos dos dez produtos e o terceiro vetor com os preços dos dez produtos. Mostre um relatório apenas com o nome, o código, o preço e o novo preço dos produtos que sofrerão aumento. Sabe-se que os produtos que sofrerão aumento são aqueles que possuem código par ou preço superior a R\$ 1.000,00. Sabe-se ainda que se o produto satisfaz as duas condições (código par e preço superior a R\$ 1.000,00), o aumento de preço será de 20%; se satisfaz apenas a condição do código par o aumento será de 15%; se satisfaz apenas a condição de preço, o aumento será 10%.

Solução Exercício 2

17

Algoritmo soma elementos vetor;

Início

var

inteiro: soma, I;

A: vetor [50] de real;

para I \leftarrow 0 até 49 passo 1 faça

início

escreva ("Digite um n° inteiro: ");

leia (A [I]);

fim para;

// Continua no próximo slide

Solução Exercício 2

18

```
soma ← 0;  
para I ← 0 até 49 passo 1 faça  
  início  
    soma ← soma + A[ I ];  
fim para;
```

// Continua no próximo slide

Solução Exercício 2

19

```
para I ← 0 até 49 passo 1 faça  
  início
```

```
    escreva (A [ I ] );
```

```
  fim para;
```

```
  escreva ("Soma dos elementos do vetor = ", soma);
```

```
Fim.
```

Solução Exercício 2

20

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    int  soma = 0, A [50],  I;
    for (I = 0; I <= 49; I = I + 1)
    {
        printf("\n Digite um n° inteiro: ");
        scanf("%d", & A [ I ]);
    }
    // Continua no próximo slide
}
```

Solução Exercício 2

21

```
for( I = 0; I <= 49; I = I + 1)
{
    soma = soma + A [ I ];
}
```

// Continua no próximo slide

Solução Exercício 2

22

```
for (I = 0; I <= 49; I = I + 1)
{
    printf("    %d    ", A [ I ] );
}
printf("\n\n Soma dos elementos do vetor: %d \n", soma);
system("pause");
}
```

Solução Exercício 3

23

Algoritmo multiplica vetor;

Início

var

inteiro: I;

A, B, C: vetor [150] de real;

para I \leftarrow 0 até 149 passo 1 faça

início

escreva ("Digite um n° inteiro: ");

leia (A [I]);

fim para;

// Continua no próximo slide

Solução Exercício 3

24

```
para I ← 0 até 149 passo 1 faça  
  início  
    escreva ("Digite um n° inteiro: ");  
    leia (B [ I ] );  
  fim para;
```

```
para I ← 0 até 149 passo 1 faça  
  início  
    C [ I ] ← A [ I ] * B [ I ] ;  
  fim para;  
// Continua no próximo slide
```


Solução Exercício 3

25

```
para I ← 0 até 149 passo 1 faça  
  início  
    escreva (A[ I ], " * ", B[ I ], " = ", C[ I ]);  
  fim para;
```

Fim.

Solução Exercício 4

26

Algoritmo Locadora;

Início

var

inteiro: I;

locacoes, locgratis: vetor [80] de inteiro;

nomecli: vetor [80] de string;

para I \leftarrow 0 até 79 passo 1 faça

início

leia(nomecli[I]);

leia(locacoes [I]);

fim para;

// Continua no próximo slide

Solução Exercício 4

27

```
para I ← 0 até 79 passo 1 faça  
  início  
    locgratis [ I ] ← locacoes [ I ] / 10;  
  fim para;
```

```
para I ← 0 até 79 passo 1 faça  
  início  
    escreva(nomecli[ I ], locacoes[ I ], locgratis [ I ]);  
  fim para;
```

Fim.

Solução Exercício 5

28

Algoritmo Produtos;

Início

var

inteiro: I;

preco, preconovo: vetor [10] de real;

produto: vetor [10] de string;

codigo: vetor [10] de inteiro;

para I \leftarrow 0 até 9 passo 1 faça

início

leia(produto[I]);

leia(codigo[I]);

leia(preco[I]);

fim para;

// Continua no próximo slide

Solução Exercício 5

29

```
para I ← 0 até 9 passo 1 faça
  início
    se preco[I] > 1000.00 E codigo[I] mod 2 =0 então
      início
        preconovo[I] ← preco[I] + (preco[I] * 0.20);
      fim
    senão
      início
        // Continua no próximo slide
```

Solução Exercício 5

30

```
se preco[I] > 1000.00 então
  início
    preconovo[I] ← preco[I] + (preco[I] * 0.10);
  fim
senão
  início
    // Continua no próximo slide
```

Solução Exercício 5

31

```
se codigo[I] mod 2 =0 então
  início
    preconovo[I] ← preco[I]+(preco[I]*0.15);
  fim
senão
  início
    preconovo[I] ← preco[I];
  fim se;
fim se;
fim se;
fim para;
// Continua no próximo slide
```

Solução Exercício 5

32

```
para I ← 0 até 9 passo 1 faça
```

```
  início
```

```
    escreva(produto[ I ], codigo[ I ], preco[ I ],  
    preconovo[I]);
```

```
  fim para;
```

```
Fim.
```


Exercício para entregar

33

- Uma cidade do interior de São Paulo tem 1.000 residências cadastradas na prefeitura. Nesta cidade foi feita uma pesquisa de audiência de canal de TV em várias de suas casas, em um determinado dia. Para cada casa consultada foi fornecido o número do canal (4, 5, 7, 12) e o número de pessoas que estavam assistindo àquele canal. Se a televisão estivesse desligada, nada era anotado, ou seja, essa casa não entrava na pesquisa. Faça um algoritmo que:
 - ▣ leia número do canal e o número de pessoas que estavam assistindo de um conjunto de pessoas;
 - ▣ Encontre a residência que tem mais moradores;
 - ▣ Encontre o canal mais assistido;
 - ▣ calcule e mostre a percentagem de audiência de cada canal.
- Para finalizar o algoritmo digite o número do canal ZERO.

Exercício

34

- Uma empresa fez uma pesquisa de mercado para saber se as pessoas gostaram ou não de um novo produto lançado. Para isso cadastrou o sexo do entrevistado e sua resposta (S - Sim ou N - Não). Sabe-se que foram entrevistadas 1000 pessoas. Faça um algoritmo que calcule e mostre:
 - ▣ o número de pessoas que respondeu sim;
 - ▣ o número de pessoas que respondeu não;
 - ▣ o número de mulheres que respondeu sim;
 - ▣ a percentagem de homens que respondeu não entre todos os homens analisados.

Bibliografia

35

□ **Básica**

ASCENCIO, A. F. G, CAMPOS, E. A. V. **Fundamentos da Programação de Computadores**: algoritmos, Pascal e C/C++ e Java. Longman, 2007.

FORBELLONE, L. V., EBERSPACHER, H. F. **Lógica de Programação**: a construção de algoritmos e estruturas de dados. Prentice Hall, 2005.

ZIVIANI, Nivio. **Projeto de Algoritmos com Implementações em Pascal e C**. 2.ed. Thomson Pioneira, 2004.

□ **Complementar**

FARRER, H et al. **Algoritmos estruturados**. 3 ed. Rio de Janeiro: LTC, 1999. 284 p.

MANZANO, J. A. N. G.; **Estudo dirigido de algoritmos**. 9. ed. São Paulo: Érica, 2004.

LOUNDON, L. **Algoritmos em C**. São Paulo: Ciência Moderna, 2000.

ASCENCIO, A. F. G.; CAMPOS, E.A.V. **Fundamentos da programação de computadores**: algoritmo, pascal e C++. São Paulo: Pearson Prentice Hall, 2002. 355 p