

# Strings, Vetores e Matrizes

Professora: Simone Regina da Silva  
[simone.regina@ifsc.edu.br](mailto:simone.regina@ifsc.edu.br)

# Pauta

- Strings
- Vetores;
- Matrizes;
- Exercícios;

# Introdução Vetor/Matriz

**Problema:** Cadastrar os dados pessoais de 50 alunos e suas médias

Imaginem como ficaria na declaração de variáveis, declarando uma a uma, as 50 variáveis para o nome, depois as variáveis para as médias de cada aluno...

Em casos como esse que é útil a utilização da estrutura de dados conhecida como vetor

Um vetor é uma espécie de caixa com várias divisórias para armazenar coisas (dados)  
É uma variável que pode armazenar vários valores

# Introdução Vetor/Matriz

meuVetor



Médias



Nomes



Alunos



- ❑ Variáveis compostas homogêneas unidimensionais
- ❑ Os índices começam em zero
- ❑ A quantidade de elementos que o vetor pode armazenar deve ser um valor inteiro fixo
- ❑ **Declaração:**
  - ❑ inteiro `vet[10];`
    - ❑ Inicia na posição zero (0) e termina na posição 9 (tamanho do vetor -1)
  - ❑ caractere `x[5];`

## ❑ Atribuição de valores:

❑ `x[2] = 10`

❑ `y[0] = 5.9`

## ❑ Apresentação dos dados:

❑ `escreva(x[2]);`

❑ `escreva(y[0]);`

- ❑ Variáveis homogêneas bidimensionais ou multidimensionais
- ❑ Pode ter até 12 dimensões de acordo com o padrão ANSI;
- ❑ Para cada dimensão, um índice deve ser usado.
- ❑ Declaração:
  - ❑ `real x[2][6] //matriz com 2 linhas e 6 colunas;`
  - ❑ `caractere mat[4][3] //matriz com 4 linhas e 3 colunas;`

## ❑ Atribuição:

- ❑ `x[1][4] = 5.45` //atribui o valor 5.45 ao espaço identificado pelo índice 1 (linha 2) e o índice 4 (coluna 5)
- ❑ `mat[3][2] = 'd'` //atribui o valor d à quarta linha e à terceira coluna

## ❑ Impressão:

- ❑ `escreva(x[1][4])`
- ❑ `escreva(mat[3][2])`



Uma matriz é uma coleção de **variáveis de mesmo tipo**, acessíveis com um **único nome** e armazenados contiguamente na memória.

A individualização de cada variável de um vetor é feita através do uso de **índices**.

Os **Vetores** são matrizes de uma só dimensão.

# Declaração de Matrizes

```
int Vetor[5]; // declara um vetor de 5 posições
```

```
int Matriz[5][3]; // declara uma matriz de 5 linhas e 3 colunas
```

## Exemplos com Vetores

```
int Vetor[5]; // declara um vetor de 5 posições

int Matriz[5][3]; // declara uma matriz de 5 linhas e 3 colunas

Vetor[0] = 9; // coloca 9 na primeira posição do vetor

Vetor[4] = 30 // coloca 30 na última posição do vetor

Matriz[0][1] = 15; // coloca 15 na célula que está na primeira linha
                    // e na segunda coluna da matriz
```

# Exemplos com Vetores

## Declaração em pseudocódigo

```
INTEIRO k[10];  
REAL vet[5];  
CARACTERE nome[20];
```

```
int k[10];  
float vet[5];  
char nome[20];
```

Na declaração de um vetor podemos omitir seu tamanho , desde que quando seja feita a declaração, já sejam atribuídos seus valores, como no exemplo abaixo:

```
int num[] = {1, 3, 5, 7};
```

# Exemplos com Vetores

As posições de um vetor começam em 0 (zero) e vão até o tamanho do vetor menos uma unidade.

**int vet[6]**

<b>vet</b>	<b>12</b>	<b>100</b>	<b>45</b>	<b>3000</b>	<b>7</b>	<b>432</b>
	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>

# Exemplos com Vetores

Podemos atribuir valores ao vetor na posição desejada, como por exemplo:

**Y[0] = 19;**

**Y[3] = 397;**

Y	19			397		
	0	1	2	3	4	5

ou, colocando-se um sinal de igual (=) seguido de todos os valores separados por vírgula entre colchetes.

Exemplo:

**int Y[ ] = {12, 6, 8, 3, 1, 19};**

ou

**int Y[6] = {12, 6, 8, 3, 1, 19};**

Y	12	6	8	3	1	19
	0	1	2	3	4	5

- ▶ A forma mais eficiente de trabalhar com coleções de elementos é através da construção de vetores (arrays).
- ▶ Em Java, ***arrays são objetos que armazenam múltiplas variáveis do mesmo tipo***
  - ▶ *Uma dimensão = vetor*
  - ▶ *Duas dimensões = matriz*
- ▶ Uma vez criado, um ***array*** não pode ter seu tamanho alterado.

## Vetor = Arrays

- ▶ Os vetores são definidos pelo **tipo de dados** que eles devem armazenar e a **quantidade de posições**
- ▶ **Exemplo:**
  - ▶ Vetor de 8 posições para armazenar números reais
  - ▶ Vetor de 40 posições para armazenar objetos do tipo Aluno
- ▶ Os vetores são estruturas **homogêneas**.
  - ▶ Ex: um vetor de inteiros só armazena dados do tipo inteiro  
um vetor de alunos só armazena dados do tipo Aluno



- Vetores, também chamados de Arrays:
  - São conjunto de variáveis agrupadas e indexadas;
- Uso:  
***<nome>*: vetor [*<n>*..*<m>*] de *<tipo\_dado>***
  - Onde:
    - *<nome>* = nome do vetor;
    - *<n>* = índice do primeiro elemento;
    - *<m>* = índice do último elemento;
    - *<tipo\_dado>* = tipo de dados dos elementos;

# Vetores

- **Exemplo:**

numeros: **vetor** [1..10] **de inteiro**

numeros[1] <- 5

numeros[2] <- 10

numeros[7] <- 99

numeros[10] <- 0

numeros[3] <- 4

numeros[6] <- 123

numeros[5] <- numeros[3]

numeros[4] <- 4

numeros[9] <- 11

numeros[8] <- -numeros[7]

<b>Numeros:</b>	<b>5</b>	<b>10</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>123</b>	<b>99</b>	<b>99</b>	<b>11</b>	<b>0</b>
	1	2	3	4	5	6	7	8	9	10