

# Algoritmos e Lógica de Programação

- Douglas Baptista de Godoy

# Ementa

- Projeto e representação de algoritmos.
- Estruturas de controle de fluxo de execução: sequência, seleção e repetição.
- Tipos de dados básicos e estruturados (vetores e registros).
- Rotinas. Arquivos.
- Implementação de algoritmos usando uma linguagem de programação.

# Objetivo

- Analisar problemas computacionais e projetar soluções por meio da construção de algoritmos.

# Matrizes

- **Definição de Matriz**
- Uma Matriz pode ser definida como um conjunto de variáveis de mesmo tipo e identificadas pelo mesmo nome. Essas variáveis são diferenciadas por meio da especificação de suas posições dentro dessa estrutura.
- A linguagem C/C++ permite a declaração de matrizes unidimensionais, bidimensionais e multidimensionais. O padrão ANSI prevê até 12 dimensões. Entretanto, o limite de dimensões fica por conta da quantidade de recursos computacionais disponíveis. Apesar disso, as matrizes mais utilizadas possuem duas dimensões. Para cada dimensão deve ser utilizado um índice.

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Matrizes

- **Definição de Matriz**
- Os índices usados na linguagem C/C++, para identificar as posições de uma matriz, começam sempre em 0 (zero) e vão até o tamanho da dimensão menos uma unidade. Os índices de uma matriz em C/C++ devem sempre ser representados por um dos tipos inteiros disponíveis na linguagem.

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Matrizes

- **Declaração de Matriz**

- *tipo\_dos\_dados nome\_variável [dimensão1][dimensão2][...][dimensãoN];*
- Onde:
- *tipo\_dos\_dados*: é o tipo dos dados que serão armazenados na matriz;
- *nome\_variável*: é o nome dado à variável do tipo matriz;
- [dimensão1]: representa o tamanho da 1ª dimensão da matriz;
- [dimensão2]: representa o tamanho da 2ª dimensão da matriz;
- [dimensãoN]: representa o tamanho da n-ésima dimensão da matriz;

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Matrizes

- **Declaração de Matriz**
- Em C/C++, a indicação do tamanho das dimensões de uma matriz deve ser feita por um valor inteiro fixo (representado por um literal ou uma constante). Se houver necessidade de definir o tamanho da matriz em tempo de execução, deve-se fazê-lo por meio de ponteiros.

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Matrizes

- Exemplo de Matriz

## Declaração de Matriz

Exemplo 1:  
`float X[2][6];`

Exemplo 2:  
`char MAT[4][3];`

Exemplo 3:  
`float Y[2][4][3];`

## Atribuindo valores a Matriz

Exemplo 1:  
`X[1][4] = 5;`

Exemplo 2:  
`MAT[3][2] = 'D' ;`

Exemplo 3:  
`Y[0][3][1] = 12;`

Exemplo 4:  
`for (i = 0; i < 2; i++)  
{  
 for (j = 0; j < 6; j++)  
 scanf("%f%c", &X[i][j]);  
}`

## Mostrando elementos da Matriz

Exemplo 1:  
`printf("%f", X[1][4]);`

Exemplo 2:  
`printf("%c", MAT[3][2]);`

Exemplo 3:  
`for (i = 0; i < 3; i++)  
{  
 for (j = 0; j < 6; j++)  
 printf("%f", X[i][j]);  
}`



# Matriz

- **Exemplo de Matriz**

Da mesma maneira como ocorre com os vetores, os índices das dimensões das matrizes começam sempre em 0 (zero).

A seguir, são apresentadas algumas formas de criação de matrizes.

**Exemplo 1:**

*float x[2][6];*

Na declaração do exemplo 1, criou-se uma variável chamada x contendo duas linhas (de 0 a 1) com seis colunas cada (de 0 a 5), capaz de armazenar números reais, como pode ser observado a seguir

<b>x</b>	0						
	1						
		0	1	2	3	4	5

SLIDE: PROF. ROGERIO APARECIDO PEREIRA BOLIN

# Matriz

## Exemplo 2:

*char mat [4][3];*

Na declaração do exemplo 2, criou-se uma variável chamada **mat** contendo quatro linhas (de 0 a 3) com três colunas cada (de 0 a 2), capaz de armazenar caracteres, como pode ser observado a seguir:

mat	0			
	1			
	2			
	3			
		0	1	2

SLIDE: PROF. ROGERIO APARECIDO PEREIRA BOLIN

# Matriz

- **Atribuindo valores a uma matriz**

Atribuir valores a uma matriz significa armazenar informação em seus elementos, identificados de forma única por meio de seus índices.

$$x[1][4] = 5$$

Atribui o valor 5 à posição identificada pelos índices 1 (2ª linha) e 4 (5ª coluna).

<b>x</b>	0						
	1				5		
		0	1	2	3	4	5

SLIDE: PROF. ROGERIO APARECIDO PEREIRA BOLIN

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Matriz

## Atribuindo valores a uma matriz

*char mat [3][2] = 'D'*

Atribui a letra D à posição identificada pelos índices 3 (4ª linha) e 2 (3ª coluna).

mat	0			
	1			
	2			
	3			D
		0	1	2

SLIDE: PROF. ROGERIO APARECIDO PEREIRA BOLIN

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Matriz

Nas operações de atribuição, leitura e escrita devemos utilizar o número de repetições relativo ao tamanho das dimensões. Isto é, uma matriz de duas dimensões deve ser controlada por dois laços de repetição, de três dimensões três laços e assim por diante (MANZANO; OLIVEIRA, 1997).

O exemplo que segue no próximo slide exemplifica a leitura e impressão de elementos de uma matriz:

SLIDE: PROF. ROGERIO APARECIDO PEREIRA BOLIN

# Matriz

```
1  #include <stdio.h>
2  main()
3  {
4      int matrizA [2][10];
5      int i, j;
6
7      for (i=0; i<2; i++)
8      {
9          for (j=0; j<10; j++)
10         {
11             printf ("\n Digite o %d %d elemento da matriz: ", i, j);
12             scanf ("%d", &matrizA[i] [j]);
13         }
14     }
15
16     for (i=0; i<2; i++)
17     {
18         for (j=0; j<10; j++)
19         {
20             printf("\n0 elemento da posicao %d %d eh: %d ", i, j, matrizA[i][j]);
21         }
22     }
23
24     return(0);
25 }
```

# Referencias Bibliográficas

- ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de, **Fundamentos da Programação de Computadores**, Pearson Editora, 3ª edição.