

# Escola *SENAI* Suíço-Brasileira “ Paulo E. Tolle ”

FPOO – Fundamentos de Programação Orientada a Objeto



Profº Atila Olivi  
[prof.atila@gmail.com](mailto:prof.atila@gmail.com)

```

class Program
{
    0 references
    static void Main(string[] args)
    {
        int num1, num2, soma;

        Console.Write("Digite o número 1: ");
        num1 = Convert.ToInt32(Console.ReadLine());

        Console.Write("Digite o número 2: ");
        num2 = Convert.ToInt32(Console.ReadLine());

        soma = num1 + num2;

        Console.WriteLine($"{num1} + {num2} = {soma}");
    }
}

```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```

Digite o número 2: 2
1 + 2 = 3
PS C:\Users\PC\Documents\00000 25 2020\CFP115\FP00\Aula23_08_12_2020>

```

Até o momento as  
variáveis só  
armazenavam um e  
somente um valor

```

num1 = 1
num2 = 2
Soma = 3

```

Agora uma variável poderá armazenar mais de um valor ou seja, o agrupamento de várias informações dentro de uma variável com mesmo nome.

Exemplo:

```
numeros [0] = 5;  
numeros [1] = 3;  
numeros [2] = 7;  
numeros [3] = 1;  
numeros [4] = 0;  
numeros [5] = 4;
```

Quantos valores a variável  
números está armazenando?

Como conseguimos isso?



# Usando arrays ...

- Utilizada para a criação de estruturas;
- Caracteriza-se por ser definida uma única variável dimensionada (**com constantes inteiras e positivas**) com um determinado tamanho;
- Exemplo: próximo slide.

- Um array será representado por:
  - Seu nome;
  - E seu tamanho (dimensão) entre os colchetes [ ];
- Desta forma seria um array **mg** [0..7];
- Isso significa que no array **mg** poderão ser armazenados até oito elementos;
- Vantagem: Pode-se manipular uma quantidade de informações com pouco trabalho de processamento;
- Observem que o nome é o mesmo o que muda é a informação dentro dos [ ] - **colchetes**;
- Essa informação chamamos de **ÍNDICE**. Exemplo:
- **mg** [0] = 4,5      0 é o **ÍNDICE**, e 4,5 é o **ELEMENTO**.

mg [0] = 4,5

mg [1] = 6,5

mg [2] = 8,0

mg [3] = 3,5

mg [4] = 6,0

mg [5] = 7,0

mg [6] = 6,5

mg [7] = 6,0



## Array de uma dimensão ou vetor

Aluno	Nota1	Nota2	Nota3	Nota4	Média
1	4	6	5	3	4,5
2	6	7	5	8	6,5
3	9	8	9	6	8,0
4	3	5	4	2	3,5
5	4	6	6	8	6,0
6	7	7	7	7	7,0
7	8	7	6	5	6,5
8	6	7	2	9	6,0

Para pensar ...

Como você  
calcularia a  
média geral da  
turma?





# Na linguagem C#

- ✓ Para **declarar** um array, defina o **tipo de variável** com colchetes:

**string**[] carros; //declaramos uma variável que contém um array de strings

- ✓ Para **inserir** valores nele, coloque os valores em uma lista separada por **vírgulas**, entre **chaves**:

**string**[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

- ✓ Para **criar** uma matriz de **inteiros**, você pode escrever:

**int**[] numeros = {10, 20, 30, 40};



# Na linguagem C#

- ✓ Para **acessar** um elemento do array referindo-se ao número do **índice**;
- ✓ Esta declaração acessa o valor do primeiro elemento em carros:

```
string[] carros = {"Volvo", "BMW", "Ford", "Mazda"};
```

```
Console.WriteLine(carros[0]);
```

```
// Saída: Volvo
```

**Nota:** Os **índices** de matriz começam com **ZERO**:



```
carros [0] = "Volvo";
```

```
carros [1] = "BMW";
```

```
carros [2] = "Ford";
```

```
carros [3] = "Mazda";
```

```
for(int i = 0; i <= 3; i++)
```

```
{
```

```
    Console.WriteLine(carros[i]);
```

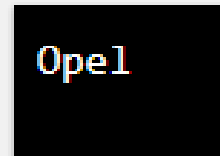
```
}
```

```
Volvo  
BMW  
Ford  
Mazda
```

# Na linguagem C#

- ✓ Para **alterar** o valor de um elemento específico, consulte o número do índice:

```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
cars[0] = "Opel";  
Console.WriteLine(cars[0]);
```



Opel

- ✓ Para **descobrir** quantos elementos uma matriz possui, use a propriedade **Length**:

```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
Console.WriteLine(cars.Length);
```



4

**Length = comprimento**



# Na linguagem C#

- ✓ Você pode **percorrer** os elementos do array com o loop for e usar a propriedade **Length** para especificar quantas vezes o loop deve ser executado.

```
string[] carros = {"Volvo", "BMW", "Ford", "Mazda"};  
for (int i = 0; i < carros.Length; i++)  
{  
    Console.WriteLine(carros[i]);  
}
```

Porque no for - loop  
i < carros.**Length** ?????

Lembrando:

valor inicial: int i = 0;

valor final : i < carros.**Length**;

passo : i++



# Estrutura de repetição foreach

Também existe um loop foreach, que é usado exclusivamente para percorrer os elementos de um array.

## **SINTAXE:**

```
foreach (tipo nomeVariável in nomeArray)
{
    // Bloco de instruções a serem executados
}
```

foreach = para cada

Onde:

tipo: pode ser string, int, float, etc.

nomeVariável: nome da variável que armazena/guarda os elementos do array;

nomeArray: nome do array que contém os elementos a serem percorridos.

Usando foreach() { }

```
string[] carros = {"Volvo", "BMW", "Ford", "Mazda"}; //array carros
```

```
//Exibindo os elementos do array carros
```

```
foreach (string i in carros)
```

```
{
```

```
    Console.WriteLine(i);
```

```
}
```

O exemplo acima pode ser lido assim:

Para cada elemento do tipo **string** (chamado **i**) em carros , imprimir o valor de **i**

# Na linguagem C#

- ✓ Existem outras formas para criar um array. Um deles, é usando a palavra-chave **new**

```
// Cria um array com quatro elementos e adiciona valores depois  
string[] carros = new string[4];
```

```
// Cria um array com quatro elementos e adicione valores imediatamente  
string[] carros = new string[4] { "Volvo", "BMW", "Ford", "Mazda"};
```

```
// Cria um array com quatro elementos sem especificar o tamanho  
string[] carros = new string[] { "Volvo", "BMW", "Ford", "Mazda"};
```

```
// Cria um array com quatro elementos, omitindo a palavra-chave new e sem especificar o tamanho  
string[] carros = { "Volvo", "BMW", "Ford", "Mazda"};
```



string[] **car** = new string[4];

Elemento	“Volvo”	“BMW”	“Ford”	“Mazda”
Índice	<b>car</b> [0]	<b>car</b> [1]	<b>car</b> [2]	<b>car</b> [3]

int[] **num** = new int[4];

Elemento	15	42	60	50
Índice	<b>num</b> [0]	<b>num</b> [1]	<b>num</b> [2]	<b>num</b> [3]

## Lendo e exibindo arrays ...

```
static void Main(string[] args)
{
    //declarar variável
    string[] carros = new string[4];

    //lendo os elementos/valores para o array
    for(int i = 0; i <= 3; i++)
    {
        Console.Write("Digite o nome de um carro: ");
        carros[i] = Console.ReadLine();
    }

    //exibindo os elementos do array
    foreach(string i in carros)
    {
        Console.WriteLine(i);
    }
}
```

i	carros[i]
0	"Volvo"
1	"BMW"
2	"Ford"
3	"Mazda"
4	-----

Volvo  
BMW  
Ford  
Mazda

O exemplo acima pode ser lido assim:

Para cada elemento do tipo **string** (chamado **i**) em carros , imprimir o valor de **i**

## Usando for(){ }

```
string[] carros = {"Volvo", "BMW", "Ford", "Mazda"};  
for (int i = 0; i < carros.Length; i++)  
{  
    Console.WriteLine(carros[i]);  
}
```

## Usando foreach(){ }

```
string[] carros = {"Volvo", "BMW", "Ford", "Mazda"};  
foreach (string i in carros)  
{  
    Console.WriteLine(i);  
}
```



# Na linguagem C#

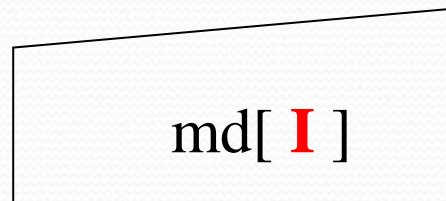
- ✓ Existem muitos métodos de array disponíveis, por exemplo `Sort()`, que classifica um array em ordem alfabética ou em ordem crescente:

```
string[] carros = {"Volvo", "BMW", "Ford", "Mazda"};
Array.Sort(carros);
foreach (string i in carros)
{
    Console.WriteLine(i);
}
```

# Leitura dos Dados de um array

- A leitura de um array é processada passo a passo, um elemento por vez;
- A instrução de leitura é o LEIA seguida da variável mais o

**I**ndice.



fluxograma





Aluno	Nota1	Nota2	Nota3	Nota4	Média
1	4	6	5	3	4,5
2	6	7	5	8	6,5
3	9	8	9	6	8,0
4	3	5	4	2	3,5
5	4	6	6	8	6,0
6	7	7	7	7	7,0
7	8	7	6	5	6,5
8	6	7	2	9	6,0

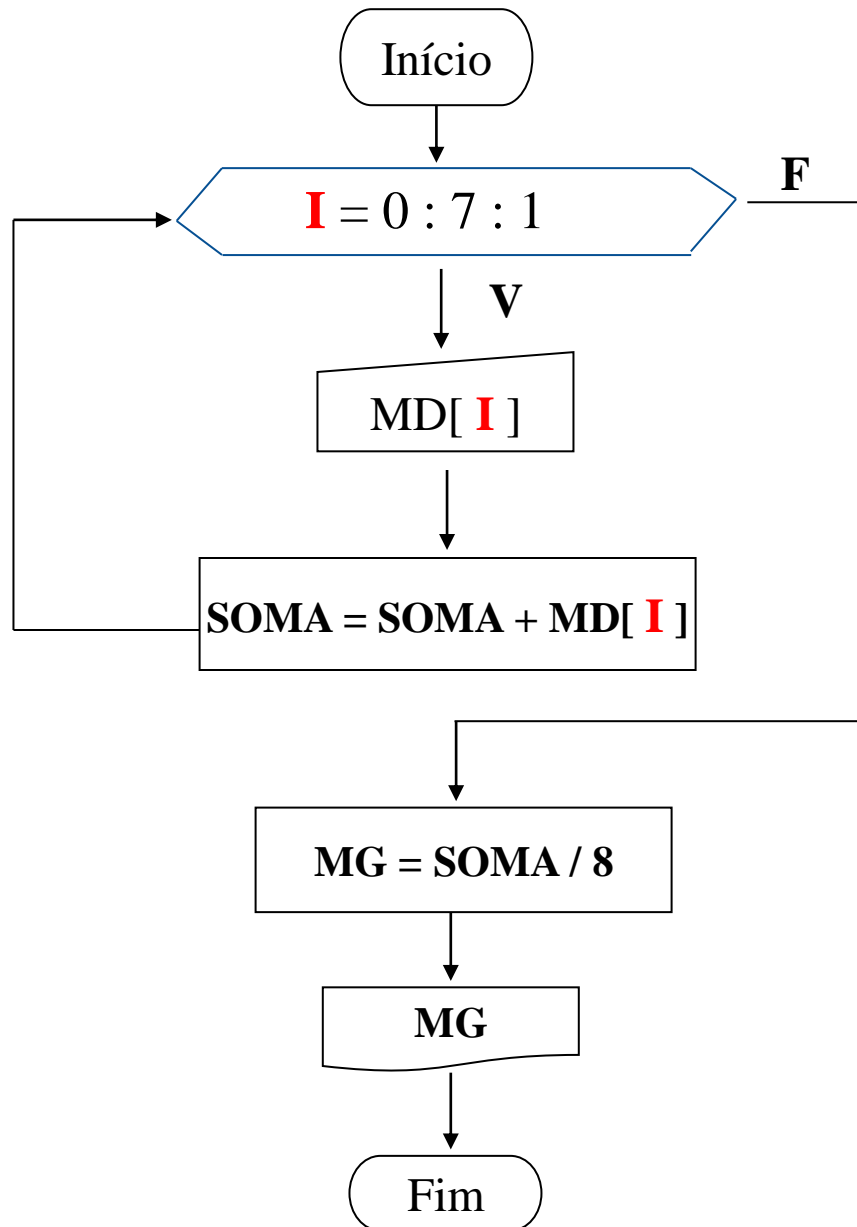
Dada a tabela ao lado elaborar um fluxograma que calcule a **média geral** da turma.

$$\text{Média Geral} = (4,5 + 6,5 + 8,0 + 3,5 + 6,0 + 7,0 + 6,5 + 6,0) / 8$$

$$\text{Média Geral} = 6,0$$



## Teste de Mesa



I	MD[ I ]	SOMA	MG
0	0	0	0

### Só para lembrar!

1) **Índice** é o endereço de alocação de uma unidade no Vetor;

2) **Elemento** é o conteúdo armazenado em um determinado endereço. EX: MD[4]

## Array de duas Dimensões ou Matriz

Aluno	Nota1	Nota2	Nota3	Nota4	Média
L1	L1C1	L1C2	L1C3	L1C4	L1C5
L2	L2C1	L2C2	L2C3	L2C4	L2C5
L3	L3C1	L3C2	L3C3	L3C4	L3C5
L4	L4C1	L4C2	L4C3	L4C4	L4C5
L5	L5C1	L5C2	L5C3	L5C4	L5C5
L6	L6C1	L6C2	L6C3	L6C4	L6C5
L7	L7C1	L7C2	L7C3	L7C4	L7C5
L8	L8C1	L8C2	L8C3	L8C4	L8C5

Legenda:

**L** – Linha

**C** - Coluna

Quantas estruturas de repetição você identifica neste problema?

# Conceitos de Matrizes

- Uma matriz na verdade é uma tabela contendo elementos. Exemplos:

5	3
6	9

Matriz 2 x 2

5	3	1
0	9	5

Matriz 2 x 3

5	3	1	8
0	9	5	6
2	3	6	9
1	2	0	2

Matriz 4 x 4

**Quando o número de linhas for igual ao número de colunas temos uma matriz QUADRADA.**





		Coluna	
		0	1
Linha	0	5	3
	1	6	9

Matriz 2 x 2

## Endereços

Linha0, Coluna0	Linha0, Coluna1
Linha1, Coluna0	Linha1, Coluna1

Endereço	Elemento
Linha 0, Coluna 0	5
Linha 0, Coluna 1	3
Linha 1, Coluna 0	6
Linha 1, Coluna 1	9

		Coluna		
		0	1	2
Linha	0	5	3	1
	1	0	9	5

Matriz 2 x 3

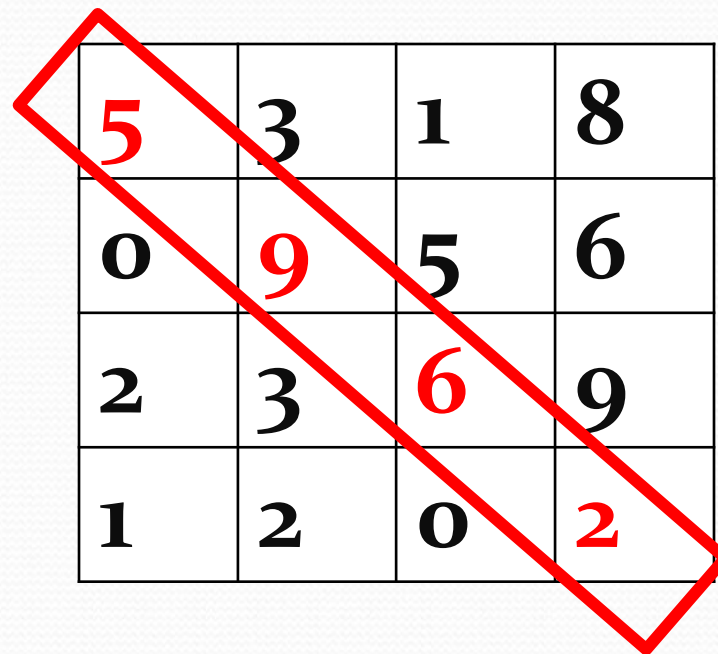
Linha0, Coluna0	Linha0, Coluna1	Linha0, Coluna2
Linha1, Coluna0	Linha1, Coluna1	Linha1, Coluna2

Endereço	Elemento
Linha 0, Coluna 0	5
Linha 0, Coluna 1	3
Linha 0, Coluna 2	1
Linha 1, Coluna 0	0
Linha 1, Coluna 1	9
Linha 1, Coluna 2	5

# Diagonal Principal

Os elementos 5, 9, 6 e 2  
formam a Diagonal Principal

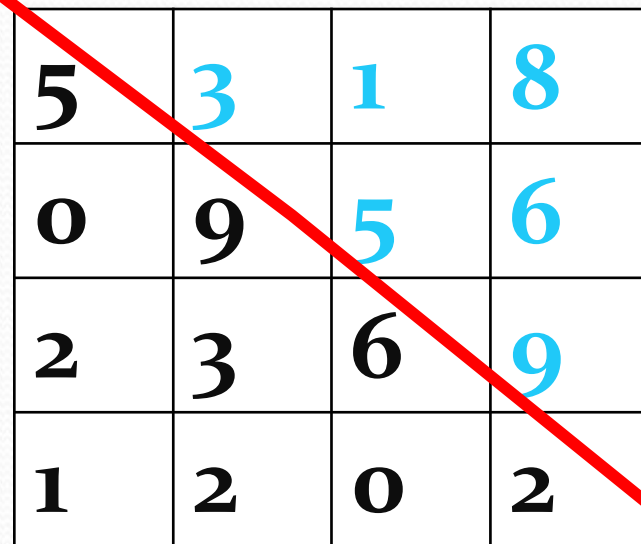
Os elementos 8, 5, 3 e 1  
formam a Diagonal  
Secundária



<b>5</b>	<b>3</b>	<b>1</b>	<b>8</b>
<b>0</b>	<b>9</b>	<b>5</b>	<b>6</b>
<b>2</b>	<b>3</b>	<b>6</b>	<b>9</b>
<b>1</b>	<b>2</b>	<b>0</b>	<b>2</b>

# Triângulo Superior

Os elementos 3, 1, 8, 5, 6 e 9 formam o Triângulo Superior da Diagonal Principal

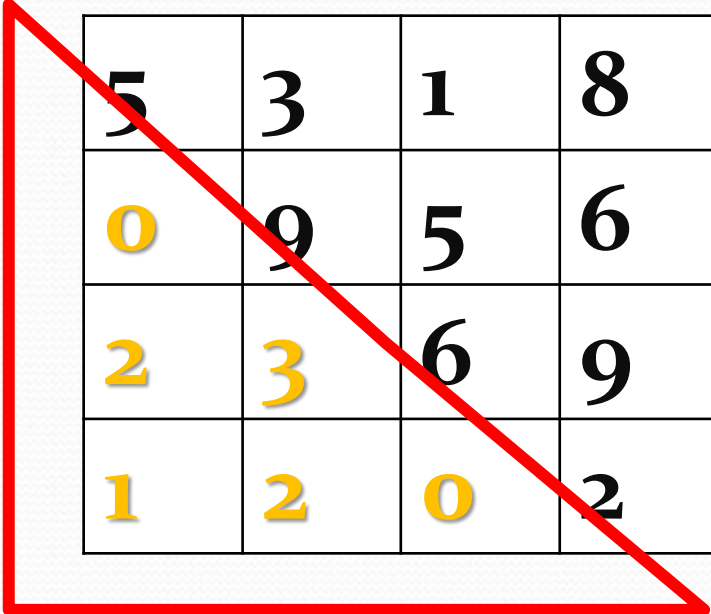


<b>5</b>	<b>3</b>	<b>1</b>	<b>8</b>
<b>0</b>	<b>9</b>	<b>5</b>	<b>6</b>
<b>2</b>	<b>3</b>	<b>6</b>	<b>9</b>
<b>1</b>	<b>2</b>	<b>0</b>	<b>2</b>



# Triângulo Inferior

Os elementos 0, 2, 3, 1, 2, 0 formam o Triângulo Inferior da Diagonal Principal



5	3	1	8
0	9	5	6
2	3	6	9
1	2	0	2

# Matrizes Bidimensionais

- Como fazer a leitura de uma Matriz Bidimensional?
- Por exemplo, uma Matriz de 4 linhas por 4 colunas;

Linha1, Coluna1	Linha1, Coluna2	Linha1, Coluna3	Linha1, Coluna4
Linha2, Coluna1	Linha2, Coluna2	Linha2, Coluna3	Linha2, Coluna4
Linha3, Coluna1	Linha3, Coluna2	Linha3, Coluna3	Linha3, Coluna4
Linha4, Coluna1	Linha4, Coluna2	Linha4, Coluna3	Linha4, Coluna4

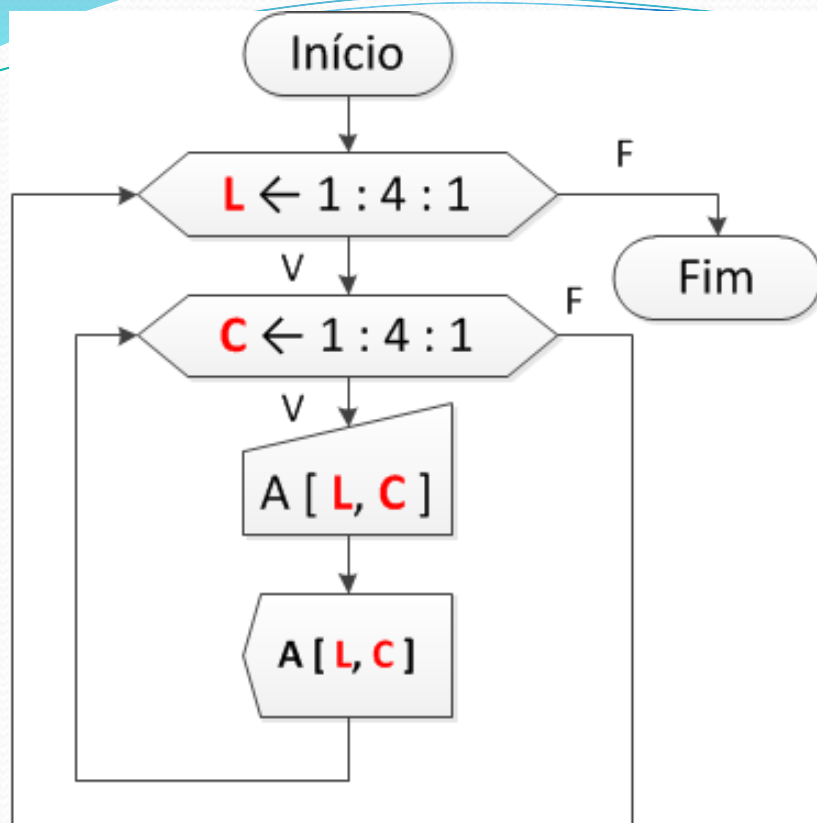
## Leitura dos Dados de uma Matriz

- A leitura de uma Matriz é processada passo a passo, um elemento por vez;
- A instrução de leitura é o LEIA seguida da variável mais os índices.

MD[ **I**, **J** ]

De onde  
veio o **I** e o **J** ?





5	3	1	8
0	9	5	6
2	3	6	9
1	2	0	2

L	C	A[L, C]
1	1	5
	2	3
	3	1
	4	8
	5	-
2	1	0
	2	9
	3	5
	4	6
	5	-
3	1	2
	2	3
	3	6
	4	9
	5	-

L	C	A[L, C]
4	1	1
	2	2
	3	0
	4	2
	5	-
5	-	-



# Exercícios ...

Codificar os exercícios abaixo em:

- ✓ C#;
- ✓ JS e;
- ✓ Python.

Página 105 do material de apoio;

Itens a, b, c e d.



# Desafio ...

		0	1	2	3	4
	Nome	Nota1	Nota2	Nota3	Nota4	Média Final
0						
1						
2						
3						
4						
5						
6						
7						