

Marcos Antônio Pereira Simões

Specialist in Teaching for Professional Education – SENAC - 2012

Specialist in Information Technology Governance – SENAC – 2010

Technologist in Data Processing – Estácio de Sá - 2007

E-mail:

marcos.matriz@notacontrol.com.br



Analista e Desenvolvedor de Sistemas

Lógica de Programação

Estrutura repetitiva "para"

É uma **estrutura de controle** que **repete** um bloco de comandos **para** um certo **intervalo de valores**.

Quando usar: quando se sabe previamente a quantidade de repetições, ou o intervalo de valores.

Por exemplo:

Fazer um programa que lê um valor inteiro N e depois N números inteiros. Ao final, mostra a soma dos N números lidos

Entrada	Saída
3 5 2 4	11

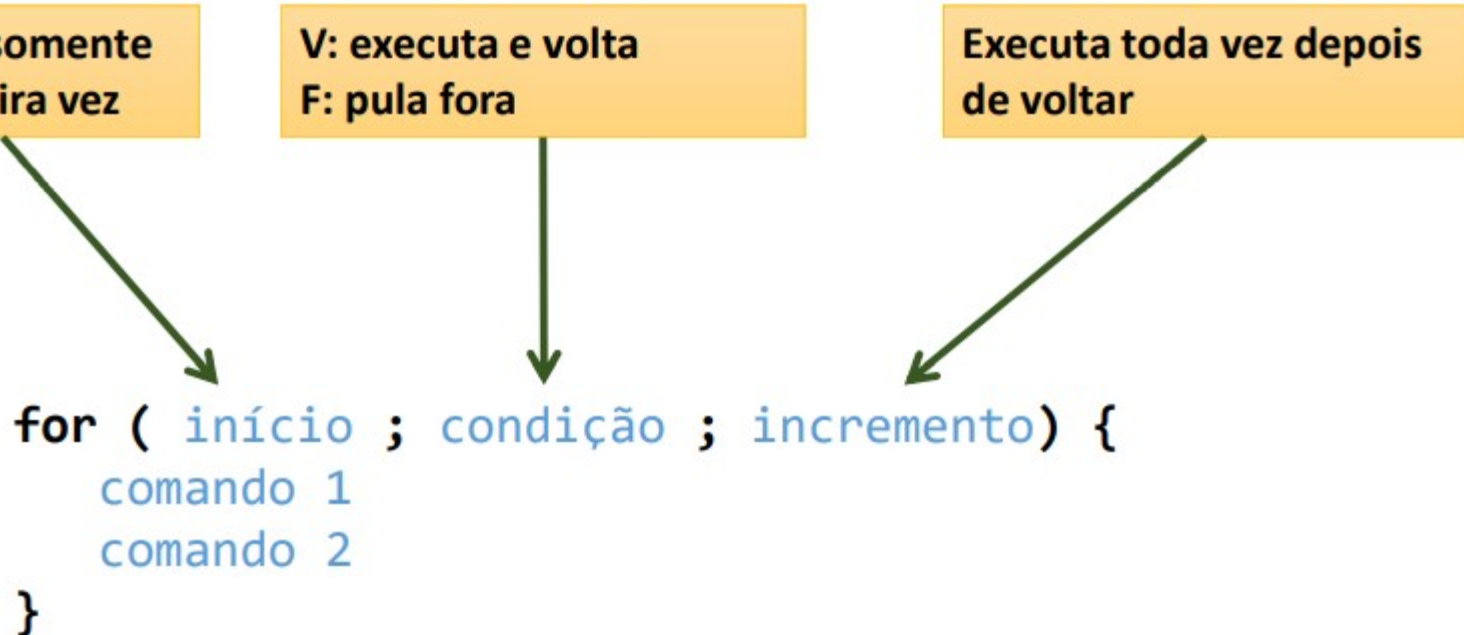
Lógica de Programação

Sintaxe / regra

Executa somente
na primeira vez

V: executa e volta
F: pula fora

Executa toda vez depois
de voltar



```
for ( início ; condição ; incremento) {  
    comando 1  
    comando 2  
}
```

Lógica de Programação

Resolução:

```
int N, x, soma;
```

```
N = int.Parse(Console.ReadLine());
```

```
soma = 0;
```

```
for (int i=1; i<=N; i=i+1){  
    x = int.Parse(Console.ReadLine());  
    soma += x;
```

```
}
```

```
Console.WriteLine(soma);
```

```
Console.ReadLine();
```

Lógica de Programação

Importante

Perceba que a estrutura "para" é ótima para se fazer uma repetição baseada em uma CONTAGEM:

```
for (int i=0; i<5; i++) {  
    Console.WriteLine("Valor de i: " + i);  
}
```

Resultado na tela:

```
Valor de i: 0  
Valor de i: 1  
Valor de i: 2  
Valor de i: 3  
Valor de i: 4
```

Lógica de Programação

Contagem regressiva

```
for (int i=4; i>=0; i--) {  
    Console.WriteLine("Valor de i: " + i);  
}
```

Resultado na tela:

```
Valor de i: 4  
Valor de i: 3  
Valor de i: 2  
Valor de i: 1  
Valor de i: 0
```

Lógica de Programação

Estrutura repetitiva "faça-enquanto"

Menos utilizada, mas em alguns casos se encaixa melhor ao problema.

O bloco de comandos executa pelo menos uma vez, pois a condição é verificada no final.

Lógica de Programação

Sintaxe / regra

```
do {  
    comando 1  
    comando 2  
} while ( condição );
```

Regra:

V: volta

F: pula fora

Lógica de Programação

Exercícios 1 ao 13

Lógica de Programação

Vetores

Um vetor corresponde a uma coleção de dados de tamanho fixo, indexada, unidimensional e homogênea

- Indexada: os elementos são acessados por meio de índices
- Unidimensional: uma dimensão
- Homogênea: todos dados são do mesmo tipo

0	Maria
1	João
2	Carlos
3	Ana
4	Joaquim

A

Lógica de Programação

Vetores

Vetores são também chamados de **arranjos** unidimensionais

Em C# a primeira posição de um vetor é a posição 0

Um arranjo deve ser alocado previamente, antes de ser utilizado. Uma vez alocado, sua quantidade de elementos é fixa

0	Maria
1	João
2	Carlos
3	Ana
4	Joaquim

A

Lógica de Programação

Como criar um vetor?

declaração



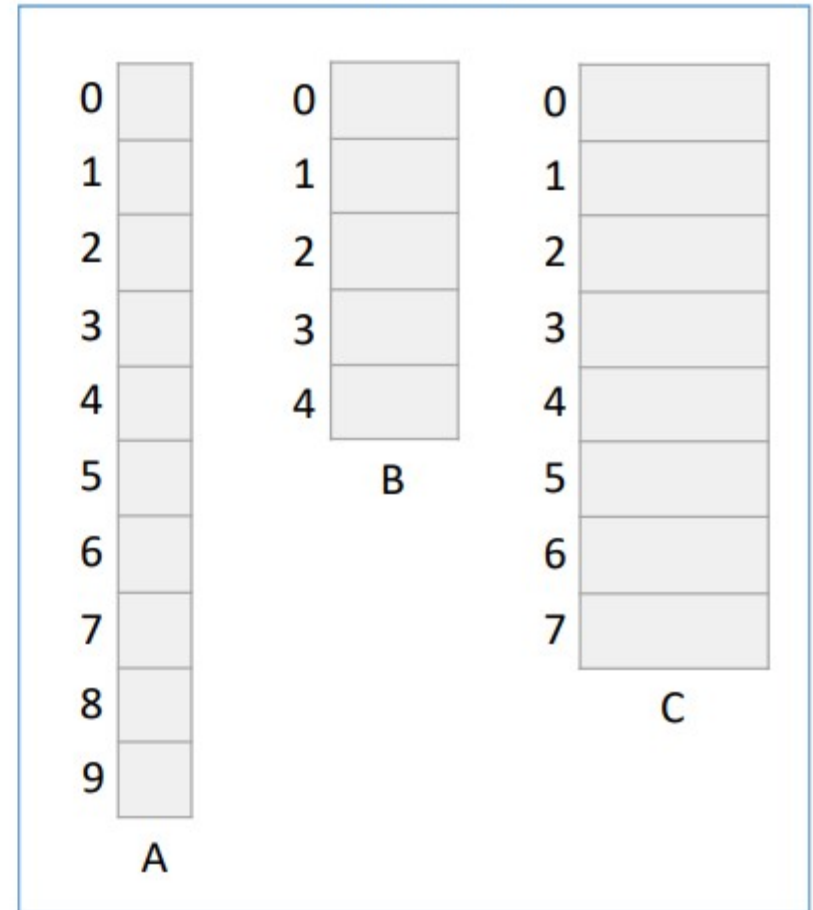
```
int[] A;  
double[] B;  
string[] C;
```

instanciação



```
A = new int[10];  
B = new double[5];  
C = new string[8];
```

Memória RAM



Lógica de Programação

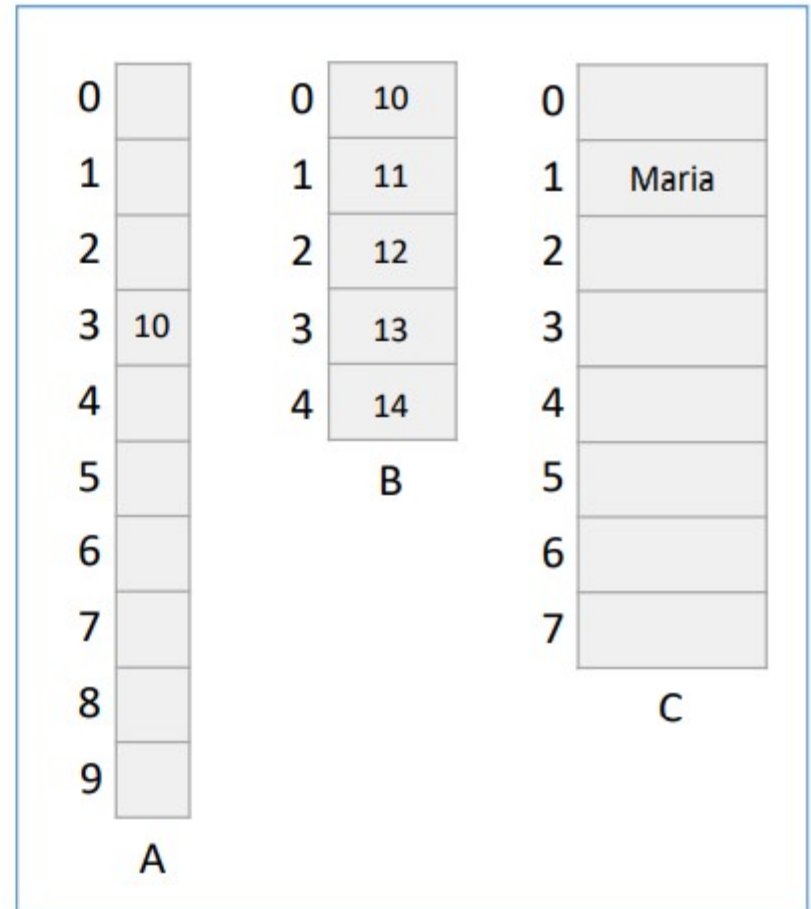
Como acessar os
elementos de um vetor?

```
A[3] = 10;
```

```
for (int i=0; i<5; i++) {  
    B[i] = i + 10;  
}
```

```
C[1] = "Maria";
```

Memória RAM



Lógica de Programação

Problema exemplo

Fazer um programa para ler um número inteiro positivo N, depois ler N números quaisquer e armazená-los em um vetor. Em seguida, mostrar na tela todos elementos do vetor.

Entrada	Saída
4	10.5
10.5	4.2
4.2	-7.1
-7.1	15.0
15.0	

Lógica de Programação

Resolução:

```
int N;  
double[] vet;  
N = int.Parse(Console.ReadLine());  
vet = new double[N];  
  
for (int i=0; i<N; i++){  
    vet[i] = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);  
}  
  
for (int i=0; i<N; i++){  
    Console.WriteLine(vet[i].ToString("F1", CultureInfo.InvariantCulture));  
}  
  
Console.ReadLine();
```

Lógica de Programação

Foreach

O foreach é um recurso do C# que possibilita executar um conjunto de comandos para cada elemento presente em uma coleção (Array, List e outras). Portanto, diferentemente do **while** e do **for**, não precisamos definir uma condição de parada. Isso é definido de forma implícita, pelo tamanho da coleção.

Sintaxe da estrutura de repetição **foreach**:

```
1  foreach (tipo elemento in coleção)
2  {
3      //bloco de código
4  }
```


Lógica de Programação

Exemplo prático:

Considere que desejamos imprimir na tela todos os nomes presentes em um array. Para isso, em vez de criar um while e nos preocuparmos com a condição de parada, podemos fazer uso do foreach.

Exemplo de uso:

```
1  string[] nomes = { "André", "Bruna", "Carla", "Daniel" };  
2  
3  foreach (string nome in nomes)  
4  {  
5      Console.WriteLine(nome);  
6  }
```

Lógica de Programação

Exercícios 14 ao 23