

IFRN

PROGRAMAÇÃO BÁSICA EM C#

Repetição

Prof. Gilbert Azevedo

Objetivos

- Utilizar as estruturas de repetição: while, for, do – while
- Utilizar os operadores de atribuição compostos

Estruturas de Repetição

- Controlam a repetição de um comando ou de um bloco
- Enquanto (while)
 - Repetição com teste no início
 - Repete comandos enquanto o teste é verdadeiro
- Para (for)
 - Repetição com variação fixa (em C#, enquanto um teste é verdadeiro)
 - Repete comandos um número determinado de vezes, gerenciado por uma variável de controle
- Repita (do-while)
 - Repetição com teste no fim
 - Repete comandos enquanto o teste é falso (verdadeiro para o do-while)

Operadores de Atribuição Compostos

- Combinam operadores aritméticos com uma atribuição
 - `int x = 10;`
 - `x = x + 10; // 20`
 - `x += 5; // 25`

Operação	Operador composto
<code>x = x * y</code>	<code>x *= y</code>
<code>x = x / y</code>	<code>x /= y</code>
<code>x = x % y</code>	<code>x %= y</code>
<code>x = x + y</code>	<code>x += y</code>
<code>x = x - y</code>	<code>x -= y</code>

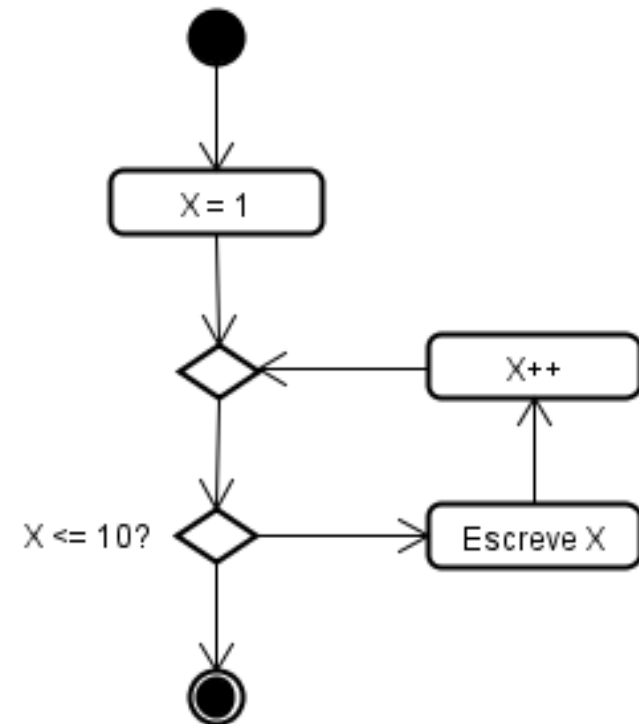
Estrutura While

- Controla a repetição de comandos enquanto uma expressão booleana (teste) é verdadeira

Algoritmo	C#
enquanto teste faça comando;	while (teste) comando;
enquanto teste faça início comandos; fim;	while (teste) { comandos; }

Diagrama do While

- O primeiro teste num *while* normalmente é verdadeiro
- O teste deve ser falso em um tempo finito
- Algoritmo: Escrever de 1 até 10
 - `int x = 1;`
 - `while (x <= 10) {`
 - `Console.WriteLine(x);`
 - `x++;`
 - `}`



Exemplo de While em C#

- Algoritmo para mostrar e somar os ímpares menores que 20
- `int s = 0;` // Somatório inicia com zero
- `int n = 1;` // Primeiro elemento
- `while (n < 20) {` // Teste
- `Console.WriteLine(n);` // Mostra o valor
- `s += n;` // Soma o valor no total
- `n += 2;` // Atualiza o valor
- `}`
- `Console.WriteLine($"Soma = {s}");` // Resultado

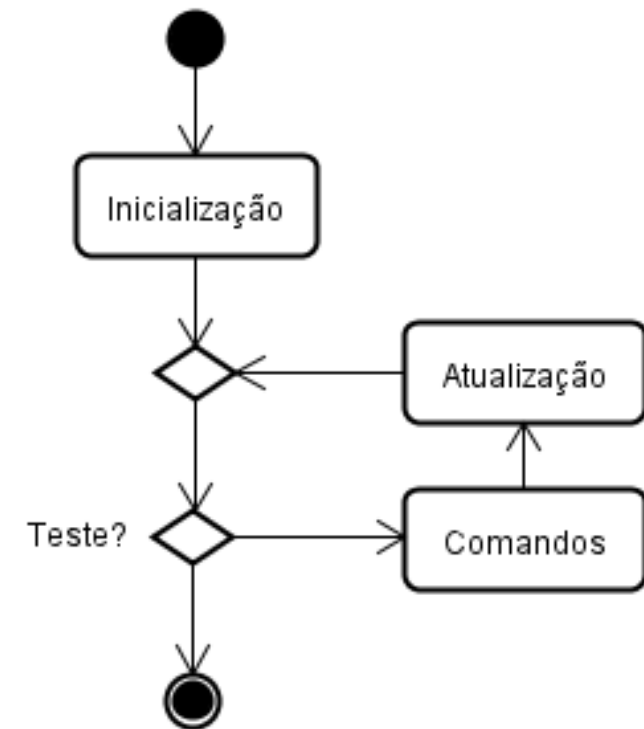
Estrutura For

- Repete um comando ou bloco de comandos um número determinado de vezes, controlado por uma variável de controle
- No C#, o for é semelhante ao while, controlando a repetição de comandos enquanto o teste é verdadeiro

Algoritmo	C#
para var de início até fim faça comando;	for(var=início; var<=fim; var++) comando;
para var de início até fim faça início comandos; fim;	for(var=início; var<=fim; var++) { comandos; }

Comparação entre For e While

- inicialização;
- while (teste) {
- comandos;
- atualização;
- }
- for (inicialização; teste; atualização) {
- comandos;
- }



Comparação entre For e While

- Repetição de 1 a 10 com while
 - `int x = 1;`
 - `while (x <= 10) {`
 - `Console.WriteLine(x);`
 - `x++;`
 - `}`
- Repetição de 1 a 10 com for
 - `for(int x = 1; x <= 10; x++) {`
 - `Console.WriteLine(x);`
 - `}`

Exemplo de For em C#

- Algoritmo para calcular o fatorial de um número: $n! = 1*2*3*...*n$
- `int f = 1;` // Produtório inicia em um
- `Console.WriteLine("Digite um número");` // Mensagem
- `int n = int.Parse(Console.ReadLine());` // Entrada de dados
- `for (int i = 1; i <= n; i++)` // Repete até chegar em n
- `f = f * i;` // Multiplica cada fator
- `Console.WriteLine($"Fatorial = {f}");` // Resultado

Estrutura Do-While

- Repete um comando ou bloco de comandos enquanto o teste é falso (verdadeiro em C#)

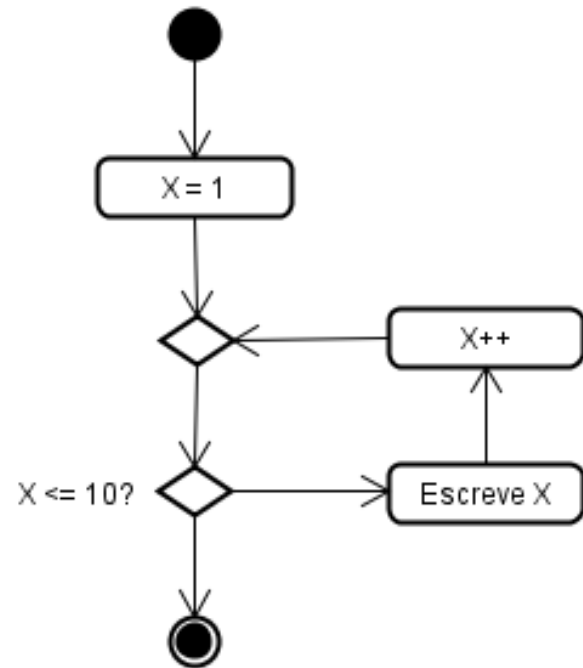
Algoritmo	C#
repita comando; até que teste;	do comando; while (teste);
repita comandos; até que teste;	do { comandos; } while (teste);

Comparação entre While e Do-While

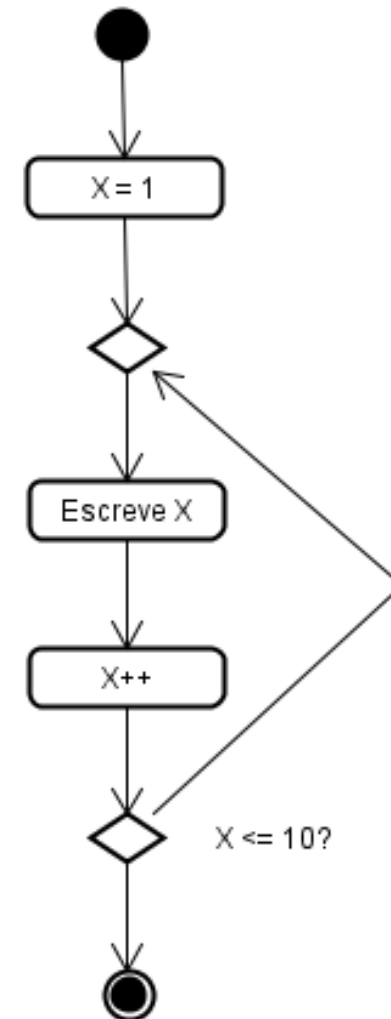
- As estruturas do-while e while são semelhantes e repetem comandos enquanto um teste for verdadeiro
- No do-while, entretanto, os comandos são realizados pelo menos uma vez, pois o teste ocorre no final
 - while (teste) { // Repete se verdadeiro – Teste no início
 - comandos;
 - }
 - do {
 - comandos;
 - } while (teste); // Repete se verdadeiro – Teste no fim

Comparação entre While e Do-While

- While



Do-While



Do-While: Código C#

- Algoritmo para calcular a média de números n números, utilizando o zero como finalizador

```
• int n = 0; // Quantidade de números
• double s = 0; // Somatório inicia com zero
• double x; // Cada valor informado
• do { // Entrada
    x = double.Parse(Console.ReadLine());
    s += x; // Soma o valor no total
    n++; // Incrementa o número de parcelas
} while (x != 0); // Continua se não foi informado 0
• double m = s / (n - 1); // Cálculo da média
• Console.WriteLine($"Media = {m}"); // Resultado
```

Referências

- Microsoft Visual C# 2010 – Passo a passo, John Sharp, Bookman, 2010
- Operadores em C#
 - <https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/>
- Instruções (Guia de Programação em C#)
 - <https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/statements-expressions-operators/statements>

Fim

- Tarefa
 - Questionário
- Próxima Aula
 - Métodos