



Microsoft
.NET



Residência
em Software

Estruturas de Controle de Fluxo

Prof. Hélder Almeida



INSTITUIÇÃO EXECUTORA



UESC

COORDENADORA



APOIO

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Estrutura condicional: If

Definições:

- Tradução: “SE” ou seja, “caso a condição seja atendida”
- Utilizado para tomada de decisões
- Trabalha com os operadores lógicos

Estrutura condicional: Else

Definições:

- Tradução: “Senão” ou seja, “caso a condição NÃO seja atendida”
- Complemento ao if
- Pode ser implementada:
 - else if(condição)
 - Não há limites (Cuidado!)

Exemplos

```
if(condicao){ // caso seja verdadeiro
    ...      // instruções
}
```

```
if(condicao) // caso seja verdadeiro
    ...      // caso só haja uma instrução, não
precisa de chaves
```

Exemplos

```
if(condicao){ // caso seja verdadeiro
```

```
...
```

```
} else {    // caso seja falso
```

```
...
```

```
}
```

```
if(condicao) // caso seja verdadeiro
```

```
...      // caso só haja uma instrução, não precisa de chaves
```

```
else     // caso seja falso
```

```
...
```

Exemplos

```
if(condicao){ // caso seja verdadeiro
    ...
} else if(condicao) {    // Senão, testa outra condição
    ...
} else if(condicao) {    // Senão, testa outra condição
    ...
} else if(condicao) {    // Senão, testa outra condição
    ...
} else {    // Se não acontecer nenhuma das anteriores
    ...
}
```

Estrutura condicional: Switch

Definições:

- Tradução: “Chavear” ...
- Utilizado quando temos muitas decisões
- Executado em cascata
- Devemos parar manualmente através do comando **break**
- Possui uma execução padrão chamada **default** que é executadas se as demais condições forem falsas

Exemplos

```
int valor = 1;

switch (valor){ // "valor" pode ser de qualquer tipo
    case 1:
        Console.WriteLine("Valor é 1");
        break;
    case 2:
        Console.WriteLine("Valor é 2");
        break;
    default:
        Console.WriteLine("Valor é diferente de 1 e 2");
        break;
}
```


Laço de Repetição: For

Definições:

- Tradução: “para”, ou seja, para cada item em um conjunto...
- Executa uma determinada ação várias vezes
- Definido pela palavra **for**
- Exige 3 parâmetros:
 - Contador
 - Condição
 - Incremento

Exemplos

```
for (var i=0; i<10; i++)  
{  
    Console.WriteLine($"0 valor de i é {i}");  
}
```

```
for (var i=0; i<=10; i++)  
    Console.WriteLine($"0 valor de i é {i}");
```

```
for (var i=1; i<10; i+=2)  
    Console.WriteLine($"0 valor de i é {i}");
```

```
for (var i=1; i<10; i*=i)  
    Console.WriteLine($"0 valor de i é {i}");
```

Laço de Repetição: While

Definições:

- Tradução: “enquanto”, ou seja, enquanto uma condição for verdadeira...
- Definido pela palavra **while**
- Exige apenas uma condição
- Verifica a condição **antes** de executar:

Exemplos

```
var valor = 0;

while (valor >=0 && valor <= 5){
    Console.WriteLine($"0 valor é {valor}");
    valor++; // valor = valor + 1;
}

while (valor >=0 && valor <= 5)
    Console.WriteLine($"0 valor é {valor++}");

while (valor >=0 && valor <= 5)
    Console.WriteLine($"0 valor é {++valor}");

var valor = -1;
while (valor >=0 && valor <= 5)
    Console.WriteLine($"0 valor é {++valor}");
```

Laço de Repetição: Do-While

Definições:

- Tradução: “faça”, ou seja, realize algo enquanto uma condição for verdadeira...
- Definido pelas palavras **do** e **while**
- Exige apenas uma condição no **while**
- Verifica a condição **depois** de executar:

Exemplos

```
var valor = 0;

do {
    Console.WriteLine($"0 valor é {valor}");
    valor++;
} while (valor >= 0 && valor <= 5);
```

```
var valor = -1;
do {
    Console.WriteLine($"0 valor é {valor}");
    valor++;
} while (valor >= 0 && valor <= 5);
```

Laço de Repetição: Foreach

Definições:

- Tradução: “para cada...”, ou seja, realize algo para cada item de uma **coleção**
- Definido pela palavra **foreach**
- Exige uma coleção como parâmetro
- Sempre percorre todos os itens da coleção

Exemplos

```
// aqui temos um array, assunto da próxima aula
string[] colecao= { "Item1", "Item2", "Item3", "Item4" };

foreach (string item in colecao)
{
    Console.WriteLine(item );
}
```


Exercício

1. Escreva um programa em C# que imprime todos os números que são divisíveis por 3 ou por 4 entre 0 e 30;
2. Faça um programa em C# que imprima os primeiros números da série de Fibonacci até passar de 100. A série de Fibonacci é a seguinte: 0, 1, 1, 2, 3, 5, 8, 13, 21 etc... Para calculá-la, o primeiro elemento vale 0, o segundo vale 1, daí por diante, o n -ésimo elemento vale o $(n-1)$ -ésimo elemento somado ao $(n-2)$ -ésimo elemento (ex: $8 = 5 + 3$)
3. Faça um programa que imprima a seguinte tabela até o nível 8:

```
1
2  4
3  6  9
4  8  12  16
n  n*2 n*3 ... n*n
```