

Operadores lógicos e estruturas de decisão

Introdução

- Os operadores são símbolos ou palavras que representam ações a serem realizadas em um programa;
- Em C#, os operadores podem ser usados para realizar operações matemáticas, comparações, atribuições, entre outras coisas.
- Cada operador tem sua própria sintaxe e regras de uso, mas juntos eles fornecem uma ampla gama de funcionalidades que permitem controlar e manipular dados em um programa.

Operadores aritméticos

Operador	Descrição
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Resto

Operadores lógicos

- Os operadores lógicos são usados para combinar valores booleanos (verdadeiro ou falso) e produzir um resultado lógico;
- Em C#, existem três operadores lógicos principais: && (E), || (OU) e ! (NÃO);
- Estes operadores lógicos são usados frequentemente em estruturas de controle de fluxo, como if e while, para tomar decisões baseadas em condições lógicas. Além disso, eles podem ser usados em expressões para determinar o valor lógico de uma variável booleana.

Operador && (E)

- Este operador lógico combina dois valores booleanos e retorna verdadeiro se ambos os valores forem verdadeiros. Caso contrário, retorna falso.

Operador || (OU)

- Este operador lógico combina dois valores booleanos e retorna verdadeiro se pelo menos um dos valores for verdadeiro. Caso contrário, retorna falso.

Operador ! (NÃO)

- Este operador lógico inverte o valor booleano. Se o valor for verdadeiro, retorna falso. Se o valor for falso, retorna verdadeiro.

Prática

- Operadores matemáticos: declare duas variáveis e teste as operações matemáticas: + - * /, mostre os resultados no console;
- Operadores booleanos: declare duas variáveis booleanas e teste os operadores &&, || e !, mostre os resultados no console;

Estruturas de decisão

- As estruturas de decisão em C# são mecanismos que permitem a execução condicional de código;
- Isso significa que o código só será executado se uma determinada condição for atendida;
- As estruturas de decisão mais comuns em C# são: if, if...else, switch;

Operadores de comparação

Operador	Descrição
==	Igualdade
>	Maior
<	Menor
>=	Maior igual
<=	Menor igual
!=	Diferente

If

- Esta é a estrutura de decisão básica em C#.
- Ela permite que você verifique se uma determinada condição é verdadeira ou falsa e execute um bloco de código apenas se a condição for verdadeira.
- Sintaxe: A palavra chave If é seguida por uma condição entre parênteses. A condição é avaliada como verdadeira ou falsa e, se for verdadeira, o código dentro do bloco delimitado por chaves será executado.

Sintaxe do If

```
if (condição)
{
    // código a ser executado se a condição for verdadeira
}
```

If...else

- Esta estrutura de decisão é uma extensão da estrutura if;
- Além de verificar se uma condição é verdadeira ou falsa, ela também permite que você especifique o código a ser executado se a condição for falsa.
- Sintaxe: A palavra chave If é seguida por uma condição entre parênteses. A condição é avaliada como verdadeira ou falsa. Se a condição for verdadeira, o código dentro do primeiro bloco delimitado por chaves será executado. Se a condição for falsa, o código dentro do segundo bloco delimitado por chaves será executado.

Sintaxe do If...else

```
if (condição)
{
    // código a ser executado se a condição for verdadeira
}
else
{
    // código a ser executado se a condição for falsa
}
```

If...else...if

- É uma estrutura de decisão em C# que permite verificar várias condições e executar o código correspondente a cada uma delas.
- Sintaxe: A palavra chave If é seguida por uma condição entre parênteses. A condição é avaliada como verdadeira ou falsa.
- Se a condição for verdadeira, o código dentro do primeiro bloco delimitado por chaves será executado e as demais condições serão ignoradas. Se a primeira condição for falsa, a estrutura verifica a próxima condição (condição2) usando a palavra-chave **else if**.
- Este processo é repetido até que uma condição verdadeira seja encontrada ou todas as condições sejam falsas.
- Neste último caso, o código dentro do bloco delimitado por chaves após a palavra-chave **else** será executado.

Sintaxe do If...else...if

```
if (condição1)
{
    // código a ser executado se a condição 1 for verdadeira
}
else if (condição2)
{
    // código a ser executado se a condição 1 for falsa e a condição 2 for verdadeira
}
else if (condição3)
{
    // código a ser executado se as condições 1 e 2 forem falsas e a condição 3 for verdadeira
}
...
else
{
    // código a ser executado se nenhuma das condições anteriores for verdadeira
}
```


Switch

- É uma estrutura de decisão em C# que permite verificar vários valores de uma única variável e executar o código correspondente a cada valor.
- É útil quando você precisa verificar vários valores de uma única variável e executar o código diferente para cada valor.
- Em comparação a estrutura *if...else...if*, a estrutura *switch* é mais fácil de ler e manter quando há muitos valores diferentes para serem verificados.

Switch

- Sintaxe: A palavra chave `switch` é seguida por uma expressão entre parênteses, cujo valor será comparado com os valores especificados em cada cláusula (`case`);
- Se o valor da expressão corresponder ao valor especificado em uma cláusula (`case`) o código dentro dessa cláusula será executado.
- A palavra chave *break* é usada para interromper a execução do código dentro do bloco `switch`;
- Se você não usar a palavra-chave `break` o código continuará sendo executado a partir do próximo `case` até encontrar a instrução `break` ou o fim do bloco `switch` quando retorna `default`

Sintaxe switch

```
switch (expressão)
{
    case valor1:
        // código a ser executado se o valor da expressão for igual a value1
        break;
    case valor2:
        // código a ser executado se o valor da expressão for igual a value2
        break;
    ...
    case valorN:
        // código a ser executado se o valor da expressão for igual a valueN
        break;
    default:
        // código a ser executado se nenhum dos valores acima for igual ao valor da expressão
        break;
}
```