

Operadores Lógicos

Todos os programas até agora consideraram if com condições de teste simples.

Alguns exemplos de testes simples:

- `b != 0;`
- `contador <= 5;`

Estas expressões testam **uma** condição. Portanto, quando mais de uma condição precisa ser testada, precisamos usar sentenças if e if-else.

A linguagem C#, assim como a maioria das linguagens de programação de alto nível suportam operadores lógicos que podem ser usados para criar operações **lógicas mais complexas**, combinando condições simples. O valor

Operadores Lógicos

Operadores lógicos são utilizados para que possamos colocar mais de um critério em nossa condição e assim tomar uma decisão mais direta.

Operador	Descrição
	Operador “ou”.
&&	Operador “e”.
!	Operador “não”

Operadores Lógicos

- Tabela

Expr 1	Expr 2	AND
True	True	True
True	False	False
False	True	False
False	False	False

Expr1	Expr2	OR
True	True	True
True	False	True
False	True	True
False	False	False

Expr1	NOT
False	True
True	False

Operadores Lógicos

- Exemplo:

```
a = 15  
b = 5  
c = 3  
d = 15  
((a > b) && (c > a))  
((a > b) || (c > a))  
((c > a) && (d <= c))  
((c > a) || (d <= c))  
((a == d) && (a > b))  
((a == d) || (a > b))  
((a == d) && !(a > b))
```

Operadores Lógicos

```
//inicio
Console.Title = "Operadores Lógicos";
double num;
Console.WriteLine("Digite um número par positivo: ");
num = Convert.ToDouble(Console.ReadLine());
if (num > 0)
{
    if (num % 2 == 0)
    {
        Console.WriteLine("Numero Par Positivo");
    }
}
Console.ReadKey();
} //fim
```

```
//inicio
Console.Title = "Operadores Lógicos";
double num;
Console.WriteLine("Digite um número par positivo: ");
num = Convert.ToDouble(Console.ReadLine());
if ((num > 0) && (num % 2 == 0))
{
    Console.WriteLine("Numero Par Positivo");
}
Console.ReadKey();
} //fim
```

Operadores Lógicos

```
double n;  
Console.WriteLine("Digite um numero entre 20 e 90:");  
n = Convert.ToDouble(Console.ReadLine());  
if ((n>=20) && (n<=90))  
{  
    Console.WriteLine("Muito bem!");  
}  
else  
{  
    Console.WriteLine("Errado");  
}  
Console.ReadKey();
```

Operadores Lógicos

Programa deve permitir a entrada de quatro notas de um aluno, calcular a sua média aritmética e imprimir a mensagem correta de acordo com os critérios abaixo:

- Média maior ou igual a 5 – “APROVADO”.
- Média maior ou igual a 3 e inferior a 5 “RECUPERAÇÃO”
- Média inferior a 3 – “REPROVADO”.

Exemplo:

```
double n1, n2, n3, n4, med;
Console.WriteLine("digite a primeira nota:");
n1 = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("digite a segunda nota:");
n2 = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("digite a terceira nota:");
n3 = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("digite a quarta nota:");
n4 = Convert.ToDouble(Console.ReadLine());
med = (n1 + n2 + n3 + n4) / 4;
Console.WriteLine("Media:" + med);
if (med < 3)
{
    Console.WriteLine("Reprovado");
}
if ((med >= 3) && (med < 5))
{
    Console.WriteLine("Recuperação");
}
if (med >= 5)
{
    Console.WriteLine("Aprovado");
}
Console.ReadKey();
```


Exercícios

1 – Crie um programa que permita que o usuário digite o valor das vendas que executou neste mês, imprimir na tela o prêmio que ganhou por estas vendas de acordo com os critérios abaixo:

- Menor que R\$ 1000,00 (Sem prêmio).
- Maior igual a R\$ 1000,00 e menor que R\$ 3000,00 (prêmio de R\$ 150,00).
- Maior igual a R\$ 3000,00 (prêmio de R\$ 300,00).

2 – Crie um programa que permite que o usuário digite o valor de um número qualquer, verificar se o número é múltiplo de 3 e de 7 ao mesmo tempo, caso seja imprimir na tela “Múltiplo de 3 e 7”, caso não imprimir “Não é múltiplo”.

Exercícios

3 - Faça um programa para ler: a descrição do produto (nome), a quantidade adquirida e o preço unitário. Calcular e escrever o total (total = quantidade adquirida * preço unitário), o desconto e o total a pagar (total a pagar = total - desconto), sabendo-se que:

- Se quantidade ≤ 5 o desconto será de 2%
- Se quantidade > 5 e quantidade ≤ 10 o desconto será de 3%
- Se quantidade > 10 o desconto será de 5%

Exercícios

4 - Faça um programa para ler as 3 notas. Calcular a média de aproveitamento, usando a fórmula abaixo e escrever o conceito do aluno de acordo com a tabela de conceitos mais abaixo:

AP = 25%

TP = 25 %

P = 50%

A atribuição de conceitos obedece a tabela abaixo:

Média de Aproveitamento	Conceito
$\geq 9,0$	A
$\geq 7,5$ e $< 9,0$	B
$\geq 6,0$ e $< 7,5$	C
$< 6,0$	D

Estrutura SWITCH CASE

Esta estrutura é redundante com a estrutura “if”, Mesmo sendo redundantes, cada uma tem seu método de funcionamento diferenciado e são recomendadas para usos diferentes, facilitando as boas praticas na programação.

```

1 switch (variável ou valor) // Coloca-se o valor a ser analisado
2 {
3
4 case valor1:           // Bloco da primeira possibilidade
5     // código 1
6     break;             // Fim do bloco
7 case valor2:           // Bloco da segunda possibilidade
8     // código 2
9     break;             // Fim do bloco
10 }
    
```

Estrutura SWITCH CASE

Podemos fazer uma analogia entre o **switch** e a estrutura **if-else-if** apresentada anteriormente. A diferença fundamental é que a estrutura switch não aceita expressões. Aceita apenas constantes. O switch testa a variável e executa a declaração cujo case corresponda ao valor atual da variável.

A declaração **default é opcional** e será executada apenas se a variável, que está sendo testada, não for igual a nenhuma das constantes.

O comando **break**, faz com que o switch seja interrompido assim que uma das declarações seja executada. Mas ele não é essencial ao comando switch. Se após a execução da declaração não houver um break, o programa continuará executando. Isto pode ser útil em algumas situações, mas eu recomendo cuidado. Veremos agora um exemplo do comando switch.

Estrutura SWITCH CASE

```
double n, resto;
Console.WriteLine("Digite um numero:");
n = Convert.ToDouble(Console.ReadLine());
resto = n % 2;
if (resto == 0)
{
    Console.WriteLine("PAR");
}
else
{
    Console.WriteLine("IMPAR");
}
Console.ReadKey();
```

```
int num, resto;
Console.WriteLine("Entre com um numero");
num = Convert.ToInt32(Console.ReadLine());
resto = num % 2;
switch (resto)
{
    case 0:
        Console.WriteLine("Par");
        break;
    default:
        Console.WriteLine("Impar");
        break;
}
Console.ReadKey();
```

Estrutura SWITCH CASE

```
//inicio
int op;
Console.WriteLine("1 - PRIMEIRA");
Console.WriteLine("2 - SEGUNDA");
Console.WriteLine("3 - TERCEIRA");
op = Convert.ToInt32(Console.ReadLine());
Console.Clear();
switch (op) {
    case 1:
        Console.WriteLine("opção 1");
        break;
    case 2:
        Console.WriteLine("opção 2");
        break;
```

```
        case 3:
            Console.WriteLine("opção 3");
            break;
        default:
            Console.WriteLine("nenhuma");
            break;
    }
    Console.ReadKey();
} //fim
```

Exercícios

1 – Elabore um programa que crie o seguinte menu:

Formulas:

1 – Baskara.

2 – Força. ($F = m * a$)

3 – Velocidade. ($V = V_0 + a * t$)

2 – Desenvolva uma calculadora, onde será necessário entrar com a operação, primeiro e segundo valor, exiba o resultado na tela.

Melhore adicionando recursos visuais como cores, centralizações, molduras etc.

Exercícios

3 - Uma empresa concederá um aumento de salário aos seus funcionários, variável de acordo com o cargo, conforme a tabela abaixo.

Programa que lê o salário e o código do cargo de um funcionário e calcula o novo salário.

Se o cargo do funcionário não estiver na tabela, ele recebe 40% de aumento.

Código	Cargo	Aumento
1	Gerente	10%
2	Analista	20%
3	Técnico	30%

- No final é mostrado o salário antigo, o novo salário e a diferença.