

Encadeamento de condições

- Outros nomes:
 - Desvio condicional composto;
 - if aninhado;

Encadeamento de condições

- É possível encadear desvios condicionais fazendo com que após um teste em que a condição testada for verdadeira, outro teste seja realizado. Este encadeamento gera estruturas como:

```
if (condição1) {  
  
    if (condição2) {  
        <bloco de instruções 1>  
    }  
    else {  
        <bloco de instruções 2>  
    }  
    <bloco de instruções 3>  
  
}  
  
else{  
    <bloco de instruções 4>  
}
```

Encadeamento de condições

- A execução deste algoritmo anterior ocorre da seguinte forma:
 - se a condicao1 for verdadeira, a condição2 determina qual dentre os <bloco de instruções 1> e <bloco de instruções 2> será executado;
 - o <bloco de instruções 3> é executado independentemente da condição2;
 - se a condicao1 for falsa, apenas o <bloco de instruções 4> será executado.

Encadeamento de condições

- É possível também encadear desvios condicionais fazendo com que após um teste em que a condição testada era falsa, outro teste seja realizado.
- Este encadeamento gera estruturas semelhantes à mostrada a seguir:

```
if (condição1) {  
<bloco de instruções 1>  
}  
else if (condição2) {  
    <bloco de instruções 2>  
}  
else {  
    <bloco de instruções 3>  
}  
<bloco de instruções 4>
```

Encadeamento de condições

- A execução do algoritmo anterior ocorre da seguinte forma:
 - se a condição1 for verdadeira, apenas o <bloco de instruções 1> será executado;
 - se a condição1 for falsa, a condição2 determina qual dos <bloco de instruções 2> e <bloco de instruções 3> será executado:
 - se a condição2 for verdadeira, o <bloco de instruções 2> será executado;
 - se for falsa, o <bloco de instruções 3>
- O <bloco de instruções 4> é executado independentemente das condicao1 e condicao2.

Exemplo 14

```
static void Main(string[] args)
{
    int num;
    Console.WriteLine("Digite um numero: ");
    num= int.Parse(Console.ReadLine());
    if (num == 10)
    {
        Console.WriteLine("\n\nVoce acertou!\n");
        Console.WriteLine("O numero eh igual a 10.\n");
    }
    else
    {
        if (num > 10)
        {
            Console.WriteLine("O numero eh maior que 10.");
        }
        else
        {
            Console.WriteLine("O numero eh menor que 10.");
        }
    }
}
```

Operadores lógicos

- São operadores que geram resultados lógicos (Verdadeiro ou Falso) pela avaliação de expressões lógicas:
 - Operador E (and): com o uso do operador E, todos os termos da expressão devem ser verdadeiros para que a expressão seja verdadeira.
 - Operador OU (or): com o uso do operador OU, pelo menos um dos termos da expressão deve ser verdadeiro para que a expressão seja verdadeira.
 - Operador NÃO (not): inverte ou nega valor lógico de uma expressão ou elemento.

Operadores lógicos

Operador	Ação
&&	And (E)
	Or (OU)
!	Not (Não)

Tabela dos operadores:

NÃO (not)

p	! p
1	0
0	1

OU (or)

p	q	p q
1	1	1
1	0	1
0	1	1
0	0	0

E (and)

p	q	p && q
1	1	1
1	0	0
0	1	0
0	0	0

1 – Verdadeiro
0 - Falso

Exercício:

- Avalie o resultado das expressões a seguir, considerando as variáveis e seus valores:
 - $a=4, b=5, c=9, d=1$

1. $((a \neq 0) \ \&\& \ b == 5)$

2. $(a > 3 \ || \ d == 1)$

3. $!((a > 3 \ || \ d < 3) \ \&\& \ (b < c \ \&\& \ b > c))$

4. $(b \neq 0 \ || \ a > 4) \ || \ (d < 3 \ \&\& \ c \geq 8)$

5. $((a > d \ || \ a < d) \ || \ a == 4) \ || \ (!(b == 5 \ \&\& \ b > c))$

Exercício:

- Respostas: Exemplo 15

Exemplo 16 – número positivo e par

```
static void Main(string[] args)
{
    int val;
    Console.WriteLine("Digite um número: ");
    val = int.Parse(Console.ReadLine());

    if (val > 0 && val % 2 == 0)
    {
        Console.WriteLine("O número é par e positivo");
    }
}
```

Exemplo 17 – número positivo e ímpar

```
static void Main(string[] args)
{
    int val;
    Console.WriteLine("Digite um número: ");
    val = int.Parse(Console.ReadLine());

    if (val > 0 && val % 2 != 0)
    {
        Console.WriteLine("O número é ímpar e positivo");
    }
}
```

Exemplo 17v2 – número positivo e ímpar

```
static void Main(string[] args)
{
    int val;
    Console.WriteLine("Digite um número: ");
    val = int.Parse(Console.ReadLine());

    if (val > 0 && !(val % 2 == 0))
    {
        Console.WriteLine("O número é ímpar e positivo");
    }
}
```

Exemplo 17v2 – número positivo e ímpar

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
main () {
```

```
    int val;
```

```
    printf("Digite um número: ");
```

```
    scanf("%d", &val);
```

val é positivo E ! (o resto da divisão de val por 2 é 0)?

```
    if (val > 0 && !(val%2==0) ) {
```

```
        printf("O número é ímpar e positivo");
```

```
    }
```

```
    getch();
```

```
}
```

Exemplo 17v2 – número positivo e ímpar

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
main () {
```

```
    int val;
```

```
    printf("Digite um número: ");
```

```
    scanf("%d", &val);
```

val é positivo E o resto da divisão de val por 2 NÃO é 0?

```
    if (val > 0 && !(val%2==0)) {
```

```
        printf("O número é ímpar e positivo");
```

```
    }
```

```
    getch();
```

```
}
```

Exemplo 18

- Considere que será concedido um bônus aos trabalhadores de uma empresa, conforme o salário atual do vendedor, descrito abaixo. Desenvolva o algoritmo para ler o salário e calcular o valor em reais do aumento. Exibir na tela, o salário atual, o aumento (% e R\$) e o novo salário.

- | Salários | % aumento |
|--|------------------|
| Até R\$ 900,00 (inclusive)..... | 5% |
| De R\$ 900,00 a R\$ 1.400,00..... | 8% |
| Acima de R\$ 1.400,00 (inclusive)..... | 10% |

Exemplo 18

```
static void Main(string[] args)
{
    double salario, aumento=0;
    Console.WriteLine("Digite o salario: ");
    salario = double.Parse(Console.ReadLine());
    if (salario < 900)
    {
        aumento = 5;
    }
    else if (salario >= 900 && salario < 1400)
    {
        aumento = 8;
    }
    else if (salario >= 1400)
    {
        aumento = 10;
    }
    salario = salario + salario * (aumento / 100);
    Console.WriteLine("O aumento eh de "+aumento+" por cento e o salario final de "+salario);
}
```

Exemplo 19

- Dados três números, verificar se eles podem representar as medidas dos lados de um triângulo e, se puderem, classificar o triângulo em **equilátero**, **isósceles** ou **escaleno**.
- Para que três números representem os **lados** de um **triângulo** é necessário que **cada um deles seja menor que a soma dos outros dois**.
- Um triângulo é **equilátero** se tem os **três lados iguais**, **isósceles** se tem apenas **dois lados iguais** e **escaleno** se tem **todos os lados distintos**.

Exemplo 19

```
static void Main(string[] args)
{
    double a, b, c;
    Console.WriteLine("\nInforme tres numeros: ");
    a = double.Parse(Console.ReadLine());
    b = double.Parse(Console.ReadLine());
    c = double.Parse(Console.ReadLine());
    if (a < (b + c) && b < (a + c) && c < (a + b))
    {
        Console.WriteLine("\nEh um triangulo: ");
        if (a == b && b == c)
        {
            Console.WriteLine("equilatero");
        }
        else if (a == b || a == c || b == c)
        {
            Console.WriteLine("isosceles");
        }
        else
        {
            Console.WriteLine("escaleno");
        }
    }
    else
    {
        Console.WriteLine("\nNao eh um triangulo");
    }
}
```