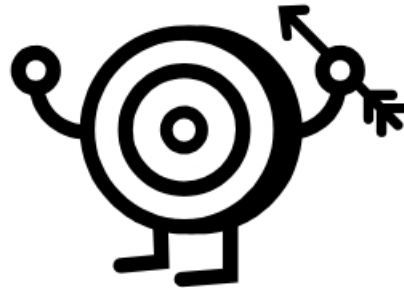




Objetivo da Aula



Construir programas capazes de repetir uma ou mais
instruções utilizando estruturas de repetição

Agenda



- ✓ Estrutura de Repetição Indefinida
- ✓ Estrutura de Repetição Definida
- ✓ Contador
- ✓ Acumulador

Estrutura de Repetição Indefinida

A repetição ocorre enquanto a condição resultar em ^{true} true.

Sintaxe:

```
while (<expressão lógica>)  
{  
    instrução_1  
    instrução_2  
    ...  
    instrução_n  
}
```

Estrutura de Repetição Indefinida - Exemplo

Escreva um programa para receber o tipo do curso (T/S).
Crítico para só aceitar “T” ou “S”.
Mostrar “Técnico” ou “Superior”, conforme a letra recebida.

```
string tipoCurso = "";
bool Console.WriteLine("Tipo de curso [T/S]: ");
tipoCurso = Console.ReadLine();
tipoCurso = tipoCurso.ToUpper();
while (erro == true)
{
    if (tipoCurso=="T" || tipoCurso=="S")
    {
        erro = false;
    }
    else
    {
        Console.WriteLine("Tipo de curso incorreto");
    }
}

if (tipoCurso ==
"T") Console.WriteLine("Técnico");
}
else
{
    Console.WriteLine("Superior");
}
```

Estrutura de Repetição Definida

A repetição ocorre uma quantidade determinada de vezes.

Sintaxe:

```
for (int var = vl_inicial; condição; incremento)
{
    instrução_1
    instrução_2
    ...
    instrução_n
}
```

Estrutura de Repetição Definida - Exemplo

Escreva um programa para receber 5 números e mostrar o maior deles.

```
int numero, maiorNumero=0;
```

```
for (int i = 1; i <= 5; i++)  
{
```

```
    Console.Write("Digite o {0}º número: ", i);
```

```
    numero = Convert.ToInt32(Console.ReadLine());
```

```
    if (numero > maior)
```

```
    {
```

```
        maiorNumero = numero;
```

```
    }
```

```
}
```

```
Console.WriteLine("O maior número digitado foi {0}", maiorNumero);
```

Contador

A estrutura de repetição possibilita contar a quantidade de repetição ou quando uma determinada condição for satisfeita.

O contador é uma variável que, a partir de um valor inicial, tem seu valor incrementado de um valor fixo, por atribuição, sempre que for demandado.

Sintaxe:

```
CONTADOR = CONTADOR + 1;
```


Contador - Exemplo

```
string tipoCurso = "";
int contador = 0;
bool erro = true;

while (erro == true)
{
    Console.Write("Tipo de curso [T/S]: ");
    tipoCurso = Console.ReadLine();
    tipoCurso = tipoCurso.ToUpper();
    contador = contador + 1;
    if (tipoCurso=="T" || tipoCurso=="S")
    {
        Erro = false;
    }
    else
    {
        Console.WriteLine("Tipo de curso incorreto");
    }
}

Console.WriteLine("Você acertou na {0}ª tentativa", contador);
```

Acumulador

A estrutura de repetição possibilita acumular valores processados durante a repetição.

O acumulador é uma variável que, a partir de um valor inicial, tem seu valor incrementado de um valor variável, por atribuição, sempre que for demandado.

Sintaxe:


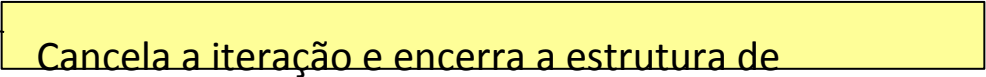
```
ACUMULADOR = ACUMULADOR + VALOR;
```

Acumulador Exemplo

```
int numero = 0, total = 0;

for (int i = 1; i <= 3; i++)
{
    Console.Write("Digite o {0}º número: ", i);
    numero = Convert.ToInt32(Console.ReadLine());
    total = total + numero;
}
Console.WriteLine("Total: {0}", total);
```

continue / break

```
double numero;  
int nrArredondado = 0;  
  
while (true)  
{  
    try  
    {  
        Console.Write("Digite um número fracionado:  
");  
        numero = Convert.ToDouble(Console.ReadLine());  
    }  
    catch  
    {  
        continue;    
    }  
    break;   
}  
  
nrArredondado=Convert.ToInt32(Math.Truncate(numero + 0.5));  
Console.WriteLine("Número arredondado: {0:F1}", nrArredondado);
```