

Passagem de Parâmetros

Algoritmos e Programação II
2014/2

Profa: Daniela Scherer dos Santos
daniela.santos37@ulbra.edu.br

Parâmetros


- Parâmetros são dados que os subprogramas precisam **RECEBER** para realizar suas operações;
- Exemplo:

```
static int somaInteiros(int a, int b)  
{  
    int soma;  
    soma = a + b;  
    return soma;  
}
```

A função `somaInteiros` recebe dois parâmetros do tipo inteiro: `a` e `b`

Parâmetros

Formais: são informados no cabeçalho do subprograma.




```
static double calculaPerimetro(double raio)
{
    double perimetro = 2*3,14*raio;
    return perimetro;
}
```

Parâmetros

Reais: são informados na chamada do subprograma.

```
static void Main(string[] args)
{
    double per = calculaPerimetro(2,5);
    Console.WriteLine("Perímetro = "+ per);
}
```



- Os parâmetros de um subprograma só serão válidos durante a execução do mesmo;

Quando o subprograma termina, os parâmetros deixam de existir.

Passagem de Parâmetros

- Pode ser:
 - Por VALOR;
 - Por REFERÊNCIA;

Passagem de Parâmetros

- **Por VALOR:**

- Mudança do valor do parâmetro dentro do subprograma não afeta a variável original;

```
static void Main(string[] args)
{
    int a = 4;
    int b = 8;
    testeValor(a, b);
    Console.WriteLine("valor de a: "+a);
    Console.WriteLine("valor de b: "+b);
}
```

```
static void testeValor(int a, int b){
    a = 2;
    b++;
    Console.WriteLine("valor de a:"+a);
    Console.WriteLine("valor de b:"+b);
}
```

Passagem de Parâmetros

- **Por VALOR:**

- Após a execução do exemplo o que será impresso?

```
static void Main(string[] args)
{
    int a = 4;
    int b = 8;
    testeValor(a, b);
    Console.WriteLine("valor de a: "+a);
    Console.WriteLine("valor de b: "+b);
}
```

Valor de a: 4
Valor de b: 8

Valor de a: 2
Valor de b: 9

```
static void testeValor(int a, int b){
    a = 2;
    b++;
    Console.WriteLine("valor de a:"+a);
    Console.WriteLine("valor de b:"+b);
}
```


Passagem de Parâmetros

- **Por VALOR:**

Note que os valores das variáveis “a” e “b” são alterados apenas dentro do procedimento *testeValor*.

Ao sair do procedimento, as variáveis permanecem inalteradas.

```
static void Main(string[] args)
{
    int a = 4;
    int b = 8;
    testeValor(a, b);
    Console.WriteLine("valor de a: "+a);
    Console.WriteLine("valor de b: "+b);
}
```

Valor de a: 4
Valor de b: 8

Valor de a: 2
Valor de b: 9

```
testeValor(int a, int b){
    a = 2;
    b++;
    Console.WriteLine("valor de a:"+a);
    Console.WriteLine("valor de b:"+b);
}
```

Passagem de Parâmetros

- **Por REFERÊNCIA:**
 - Estudaremos este assunto após trabalharmos o tema PONTEIROS.

Passagem de Parâmetros

- **Por REFERÊNCIA:**

- a mudança do valor de um parâmetro dentro do subprograma afeta o valor da variável original;

```
static void Main(string[] args)
{
    unsafe
    {
        int a = 4;
        testeRef(&a);
        Console.WriteLine("MAIN:
valor de a:" + a);
        Console.ReadKey();
    }
}
```

```
unsafe static void testeRef(int *a)
{
    *a = 2;
    Console.WriteLine("PROCEDIMENTO:
Valor de a:" + *a);
}
```

Passagem de Parâmetros

- **Por REFERÊNCIA:**

- O que será impresso após a execução do exemplo?

```
static void Main(string[] args)
{
    unsafe
    {
        int a = 4;
        testeRef(&a);
        Console.WriteLine("MAIN:
valor de a:" + a);
        Console.ReadKey();
    }
}
```

Valor de a: 2

Valor de a: 2

```
unsafe static void testeRef(int *a)
{
    *a = 2;
    Console.WriteLine("PROCEDIMENTO:
valor de a:" + *a);
}
```

Passagem de Parâmetros

- **Por REFERÊNCIA:**

- O que será impresso após a execução do exemplo?

```
static void Main(string[] args)
{
    unsafe
    {
        int a = 4;
        testeRef(&a);
        Console.WriteLine("MAIN:
valor de a:" + a);
        Console.ReadKey();
    }
}
```

Valor de a: 2

O conteúdo da variável “a” foi modificado pelo procedimento “testeRef”, pois a passagem de parâmetro foi feita por referência

Valor de a: 2

```
unsafe static void testeRef(int *a)
{
    *a = 2;
    Console.WriteLine("PROCEDIMENTO:
valor de a:" + *a);
}
```