

# Linguagem de Programação

Funções (passagem por referência)

**ECT2303**

helton.maia@ufrn.br

# Estudamos nas aulas anteriores

- Declaração de variáveis
- Declaração de funções
- Passagem de parâmetros por valor
- Passagem de parâmetros por referência

# Parâmetros de uma função

Variáveis locais, **são inicializadas** pelos valores passados na chamada da função.

```
int soma(int x, int y){  
    int s;  
    s = x + y;  
    return s;  
}
```

→ Variável local

São visíveis apenas dentro da função, criadas na entrada e eliminadas na saída do bloco { } de código (função).

# Passagem de parâmetros

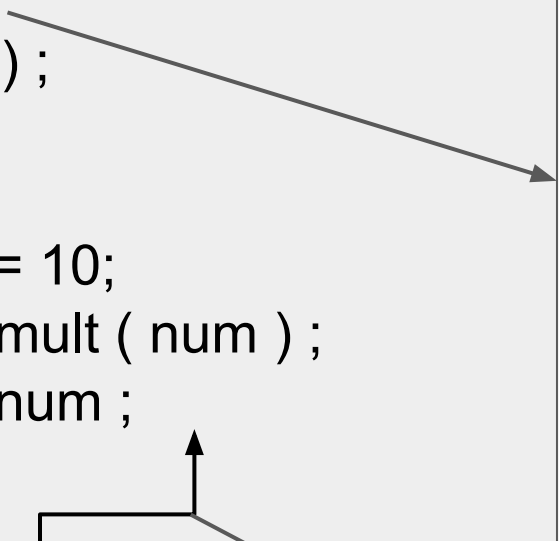
Pode-se passar parâmetros para funções de duas formas:

- **Por valor:** o valor do argumento é copiado para o parâmetro e as alterações feitas neste parâmetro não tem efeito sobre a variável utilizada na chamada da função;
- **Por referência:** o endereço de uma variável é copiado para o parâmetro, portanto, as mudanças feitas no parâmetro afetam a variável utilizada na chamada da função.

```
#include <iostream>
using namespace std ;
int var = 5;
int mult ( int ) ;
```

```
int main () {
    int num = 10;
    cout << mult ( num ) ;
    cout << num ;
    return 0;
}

int mult ( int x ) {
    x = x * x * var;
    return x ;
}
```



# Passagem de parâmetros **por valor e** **variáveis globais**

## Variável Global:

- Visíveis/válidas em todo o programa;
- Memória utilizada pela variável global é alocada durante toda a execução do programa.

**Passagem por valor**

# Variáveis Globais

## Vantagens:

- Útil quando o mesmo dado é utilizado em mais de uma função;
- Recomendado quando se deseja utilizar valores constantes.

## Desvantagens:

- Ocupam espaço de memória desnecessário mesmo sem sua utilização;
- Pode levar a erros comuns de programação.

# Passagem de parâmetros **por referência**

- Os parâmetros da função passam a referenciar o mesmo local de memória alocado para a variável em uso na chamada função;
- É passado para a função o endereço da variável utilizada na chamada da função;
- Utiliza-se o operador unário de referência (&).

# Passagem de parâmetros **por valor**

Qual seria o problema nessa troca valores?

```
1  #include <iostream>
2  using namespace std ;
3
4  void trocar (int x, int y) ;
5
6  int main () {
7      int a =1, b =2;
8      cout << " a = " << a << " b = " << b << endl ;
9      trocar (a, b ) ;
10     cout << " a = " << a << " b = " << b << endl ;
11     return 0;
12 }
13 void trocar (int x, int y) {
14     int temp = x ;
15     x = y;
16     y = temp ;
17 }
18
```



# Passagem de parâmetros **por referência**

O endereço de uma variável é copiado e mudanças no parâmetro afetam a variável utilizada.

```
1  #include <iostream>
2  using namespace std ;
3
4  void trocar (int &x, int &y) ;
5
6  int main () {
7      int a =1, b =2;
8      cout << " a = " << a << " b = " << b << endl ;
9      trocar (a, b ) ;
10     cout << " a = " << a << " b = " << b << endl ;
11     return 0;
12 }
13 void trocar (int &x, int &y) {
14     int temp = x ;
15     x = y;
16     y = temp ;
17 }
18
```

# Passagem de parâmetros **por valor e referência**

```
1  # include <iostream>
2  using namespace std;
3
4  void incremento(int x, int &y) ;
5
6  int main () {
7      int i = 5;
8      int j = 5;
9      cout << " i : " << i << " j : " << j << endl ;
10     incremento(i, j) ;
11     cout << " i : " << i << " j : " << j << endl ;
12     return 0;
13 }
14 void incremento(int x, int &y){
15     x++;
16     y++;
17 }
18
```

O que será impresso?

# Passagem de parâmetros **por referência**

```
1  # include <iostream>
2  using namespace std;
3
4  void incdec(int &x, int &y) ;
5
6  int main () {
7      int i = 5;
8      int j = 5;
9      cout << " i : " << i << " j : " << j << endl ;
10     incdec(i, j) ;
11     cout << " i : " << i << " j : " << j << endl ;
12     return 0;
13 }
14 void incdec(int &x, int &y){
15     x++;
16     y--;
17 }
18
```

O que será impresso?

# Observações

- Observe, se a função precisa realmente modificar o valor por referência, caso não precise, não utilize.
- Note que, se a função exige o retorno de mais de um valor, deve-se então utilizar parâmetros por referência.

## Exercício

Escreva uma função que recebe por referência os endereços de três variáveis, e realiza de forma circular (sentido horário) a troca dos seus valores. A função principal, deve imprimir os valores na tela, antes e depois das trocas.

**Entrada:**  $x = 1, y = 2, z = 3$

**Saída:**

**Antes:**  $x = 1, y = 2, z = 3$

**Depois:**  $x = 3, y = 1, z = 2$

?