

PASSAGEM DE PARÂMETROS

Conforme visto em aulas anteriores, os parâmetros são variáveis através das quais podemos fornecer alguma informação para uma determinada função.

Quando passamos uma variável como argumento para uma função, é feita uma cópia do seu valor para o parâmetro da função, ou seja, apesar de fornecermos um valor para a função processar, o valor da variável original permanece inalterado.

No exemplo mostrado na Figura 1, a variável **numero** é passada como argumento para a função **dobro()**, porém, seu valor permanece o mesmo.

```
void dobro (int n) {  
    n = n * 2;  
    cout << n;  
}  
int main () {  
    int numero = 5;  
    dobro(numero); //imprime 10  
    cout << numero; //imprime 5  
}
```

Figura 1

Existe uma forma de se trabalhar com parâmetros de funções onde a variável passada como argumento pode ser manipulada dentro da função. Para que isto seja possível, em vez de passar a variável em si como argumento, é preciso fornecer seu endereço de memória, desta forma, qualquer alteração realizada na variável pela função terá validade quando a função terminar de ser executada.

A Figura 2 mostra um exemplo onde uma função recebe um endereço de memória e realiza uma alteração no endereço recebido.

```
void dobro (int *n) {  
    *n = *n * 2;  
    cout << *n;  
}  
int main () {  
    int numero = 5;  
    dobro(&numero); //imprime 10  
    cout << numero; //imprime 10, pois o valor no endereço foi  
    alterado  
}
```

Figura 2



REFERÊNCIAS

ASCENCIO, A. F. G.; CAMPOS, E. A. V. Fundamentos da Programação de Computadores. 3. ed. São Paulo: Pearson, 2012.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. Lógica de Programação: A Construção de Algoritmos e Estruturas de Dados. 3. ed. São Paulo: Pearson, 2005.

C++ Pointers, W3Schools, 2020. Disponível em https://www.w3schools.com/cpp/cpp_pointers.asp. Acesso em: 17 de novembro de 2020.