C# - Passando arrays como parâmetro para métodos

Para passar um array como parâmetro a algum método, basta passá-lo **sem** os parêntesis retos, como abaixo:

```
int[] array = new int[5];

teste.dobrarArray(array);
```

IMPORTANTE: O array é uma variável que "guarda" um objeto, mas não guarda o objeto em si.

Ao invés disso, o array guarda uma referência a esse objeto (o local na memória onde esse objeto está).

No código acima, todos os elementos desse array são passados como parâmetro na linha . Por questões de performance, por default, o que é passado é apenas uma **referência do array**, ou seja, o espaço em memória onde o array está armazenado. **Isso significa que as mudanças feitas a qualquer elemento desse array dentro do método implicarão em mudanças na variável do chamador desse método**, **pois o parâmetro foi passado por referência.**

Conclusão: Por padrão, qualquer array ou objeto que é passado como parâmetro a um método, é passado por referência.

No caso de precisar passar um elemento de um array como parâmetro a um método, esse elemento (ex.: array[1]) é passado por valor ou por referência? Isso vai depender se foi usada a palavras-chave ref, ou seja, é como se trata-se de um parâmetro que não é array.

Passar uma referência (array) por referência

Apesar do array já ser uma variável que guarda uma referência e ser passado por padrão como referência a um método, existe uma diferença entre **passar essa referência por referência** ou por valor (utilizando a palavra-chave ref ou não). Quando se passa uma referência por referência, existe a possibilidade de recriar essa referência dentro do método, caso contrário não. O exemplo a seguir ilustra essa diferença.

```
class Program
{
        static void Main(string[] args)
        {
            int[] vetor = new int[4];

            // chamada de um Procedimento que coloca 1 em todos os elementos do vetor
            ColocaUMs(vetor);

            // mostra valores para verificar que houve mudança
            Console.WriteLine("Verificação 1");
            for (int i = 0; i < vetor.Length; i++)
            {
                Console.Write("{0} ", vetor[i]);
            }
            Console.WriteLine();
```

```
            // chamada de um Procedimento que não tendo passagem por referencia da referencia
            // tenta recria-lo
            ColocaDados(vetor);

            // mostra valores para verificar que ...
            Console.WriteLine("Verificação 2");
            for (int i = 0; i < vetor.Length; i++)
            {
                Console.Write("{0} ", vetor[i]);
            }
            Console.WriteLine();

            // chamada de um Procedimento que  tendo passagem por referencia afeta inclusive o
            // tamaho do vetor original
            ColocaDadosR(ref vetor);

            // mostra valores para verificar que ...
            Console.WriteLine("Verificação 3");
            for (int i = 0; i < vetor.Length; i++)
            {
                Console.Write("{0} ", vetor[i]);
            }
            Console.WriteLine();
            Console.ReadKey();

        }

        static void ColocaUMs(int[] v)
        {

            for (int i = 0; i < v.Length; i++)
            {
                v[i] = 1;
            }
        }



        static void ColocaDados(int[] v)
        {
            int Tam;

            Console.WriteLine("Digite o número de elementos do vetor:");
            Tam = Convert.ToInt32(Console.ReadLine());

            v = new int[Tam];

            for (int i = 0; i < Tam; i++)
            {
                Console.Write("Entre com {0}º de {1} Números :",i+1,Tam);
                v[i] = Convert.ToInt32(Console.ReadLine());
            }
        }


        static void ColocaDadosR(ref int[] v)
        {
            int Tam;

            Console.WriteLine("Digite o número de elementos do vetor:");
            Tam = Convert.ToInt32(Console.ReadLine());

            v = new int[Tam];

            for (int i = 0; i < Tam; i++)
            {
                Console.Write("Entre com {0}º de {1} Numeros :", i + 1, Tam);
                v[i] = Convert.ToInt32(Console.ReadLine());
            }
        }
    }
```

Tarefa: **Altere o último programa da ficha 6 – de forma a não utilizar nenhuma declaração global.**