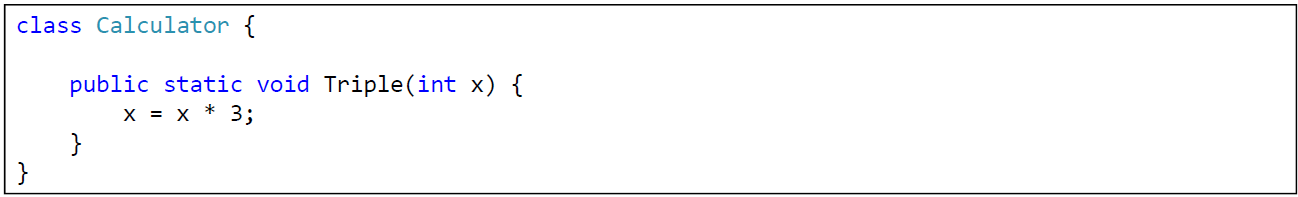
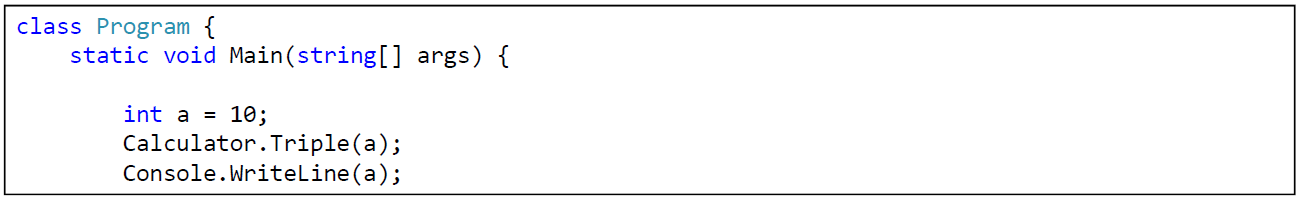
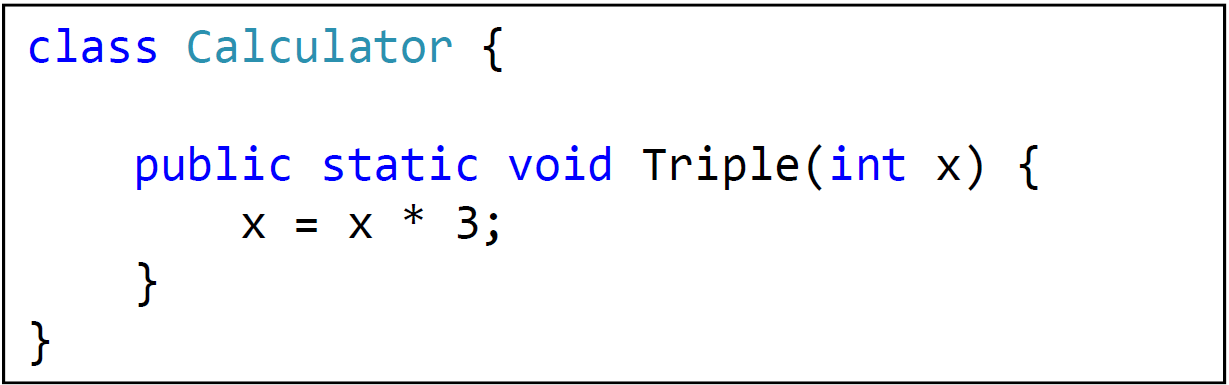
# Modificador de parâmetros: ref e out – 30/11/2020

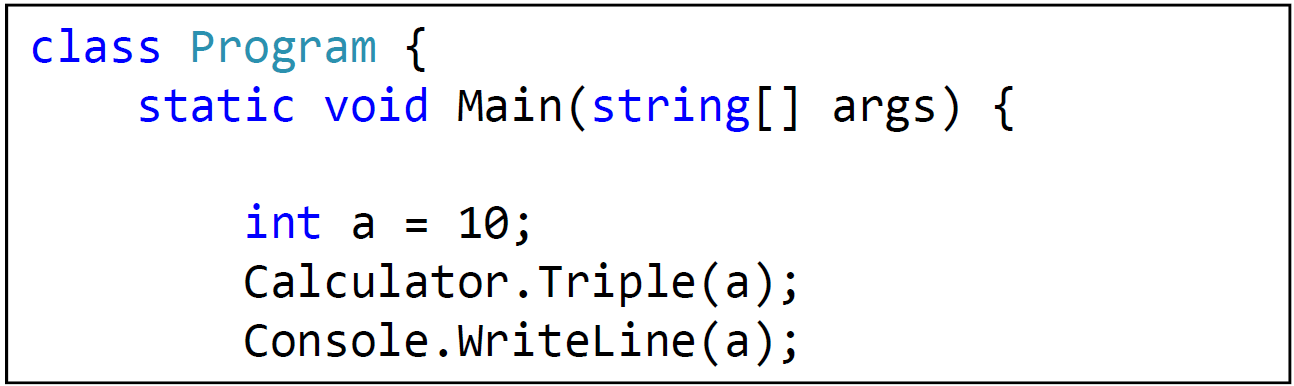
# Modificador ref

Suponha que se queira uma calculadora com uma operação para triplicar o valor de um número passado como parâmetro. A seguir uma solução que **não** funciona:

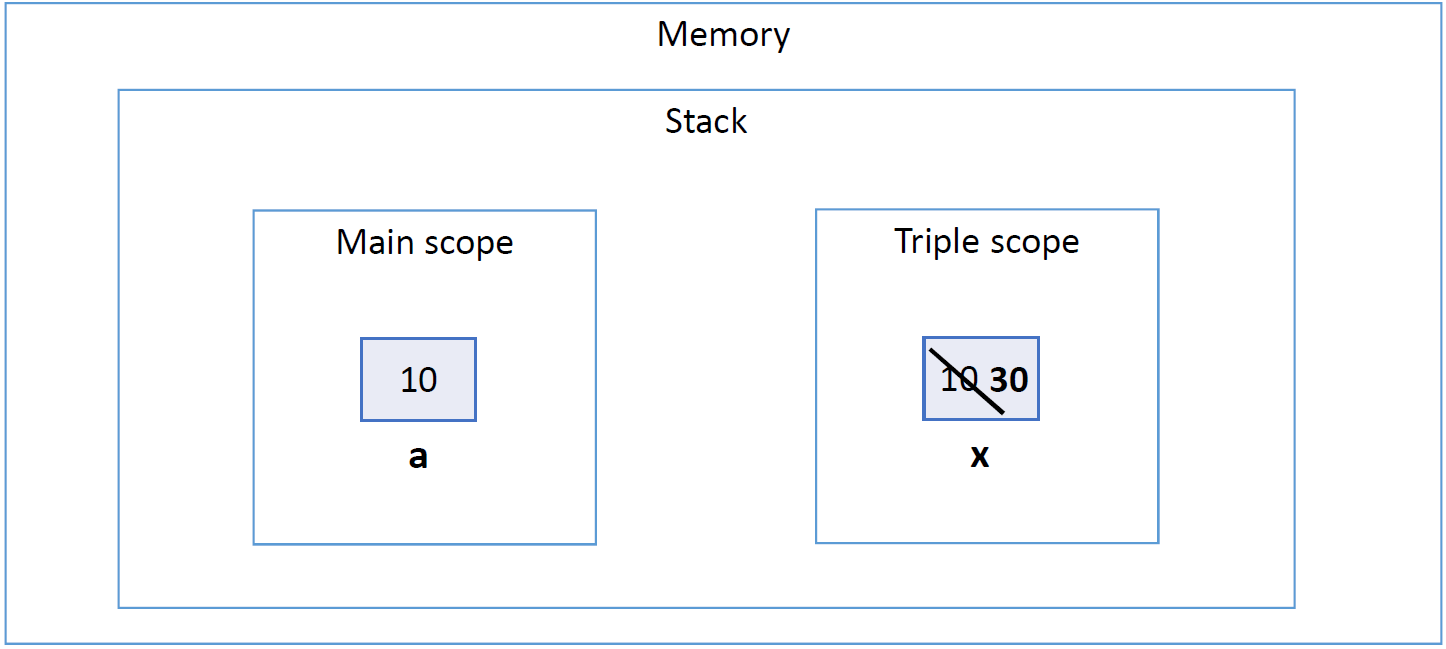




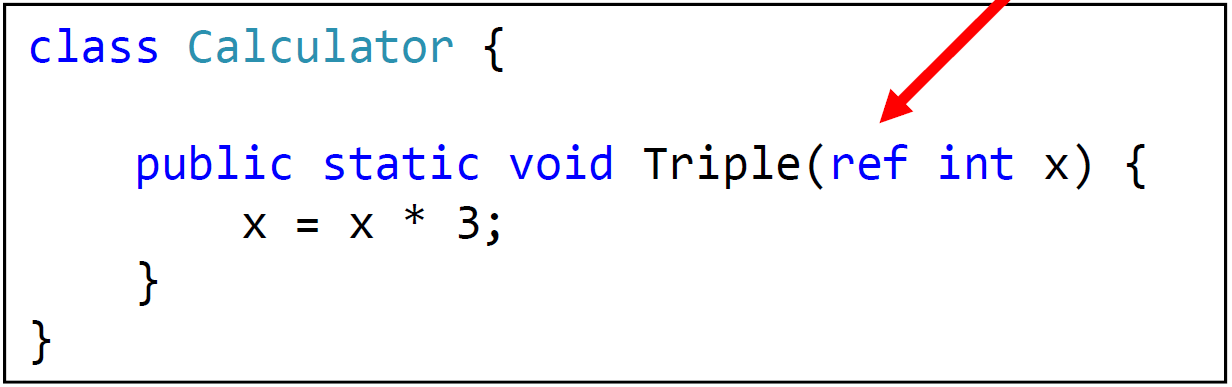


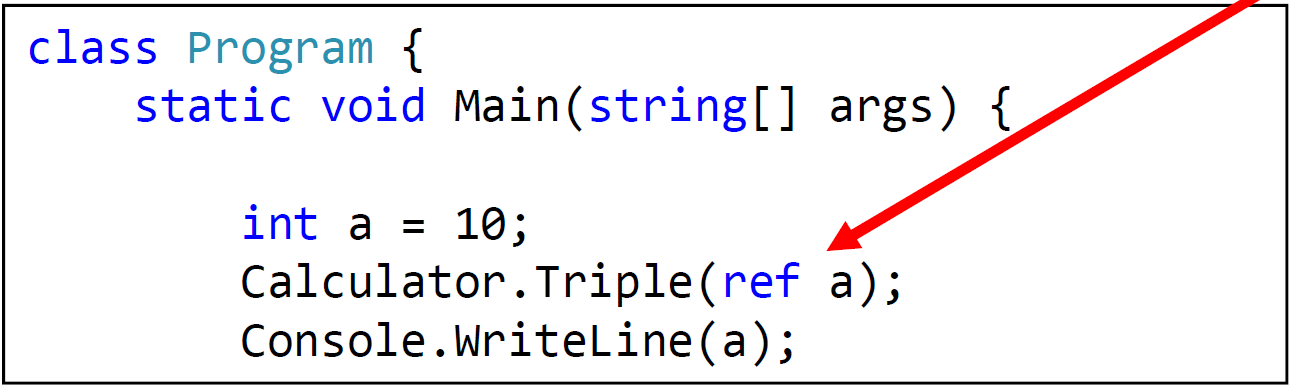


Não funciona, porque o que foi atualizado, foi o escopo da função. Não o escopo principal.

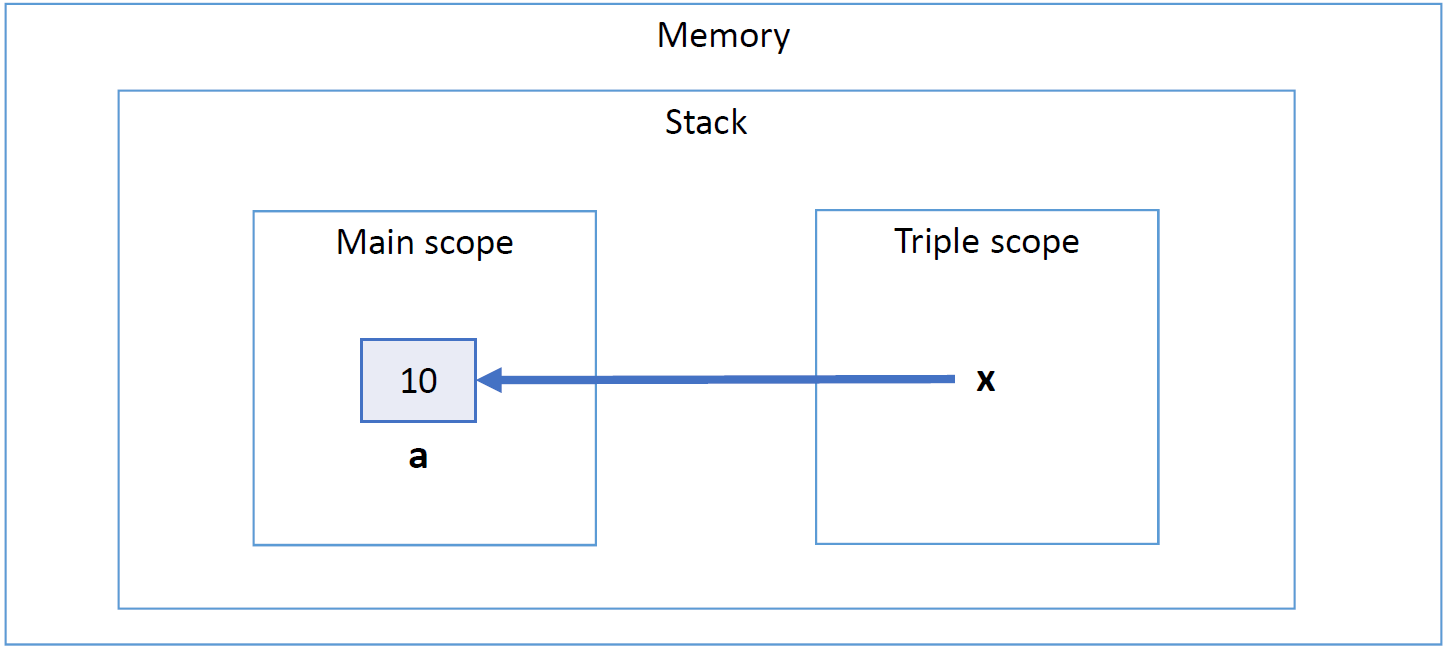


Para funcionar, precisamos de colocar a palavra reservada ref, para fazer uma referência ao escopo principal.





O ponteiro agora possui uma referência.



Código da classe Calculadora:

namespace ModificadorRef

{

class Calculadora

{

public static void Triplicar(ref int x)

{

x = x \* 3;

}

}

}

Código do programa principal:

using System;

namespace ModificadorRef

{

class Program

{

static void Main(string[] args)

{

int x = 5;

Calculadora.Triplicar( ref x);

Console.WriteLine(x);

}

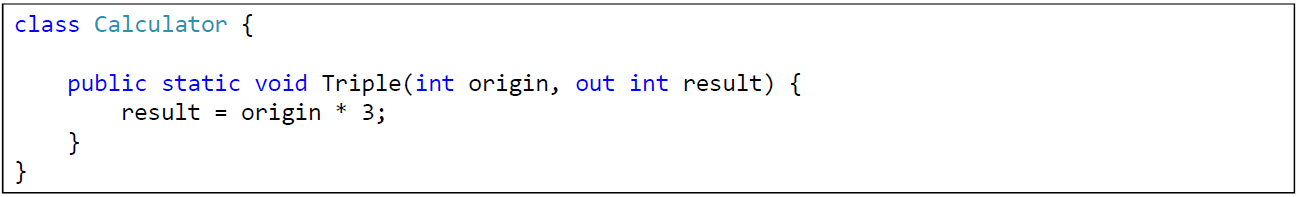
}

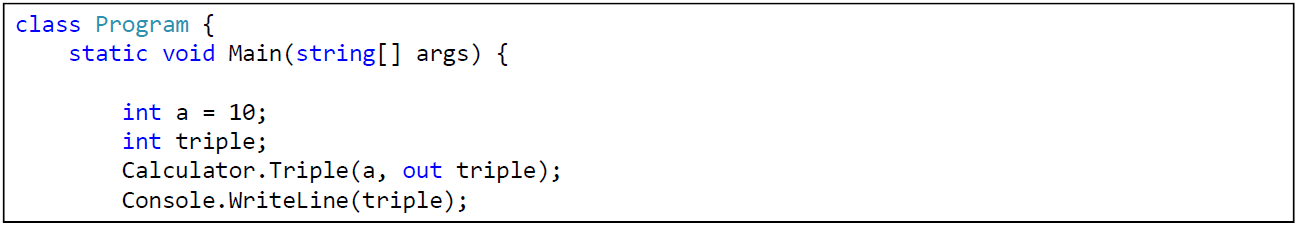
}

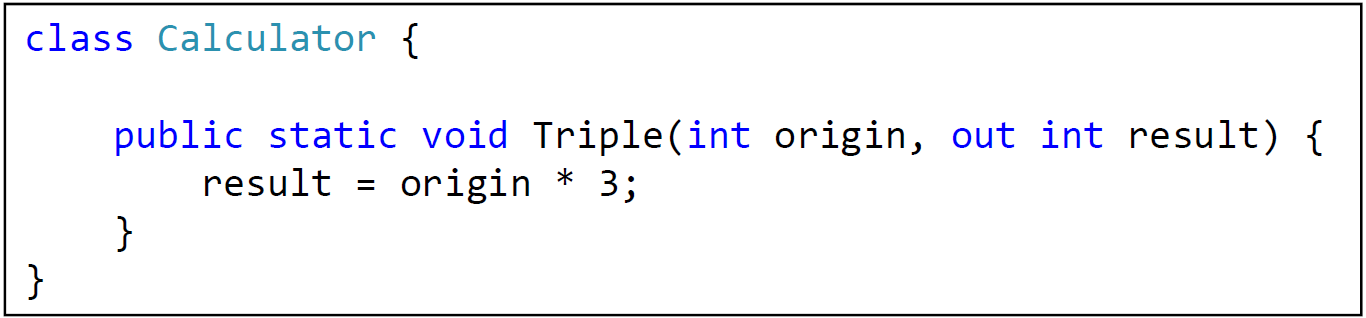
Exercício de fixação:

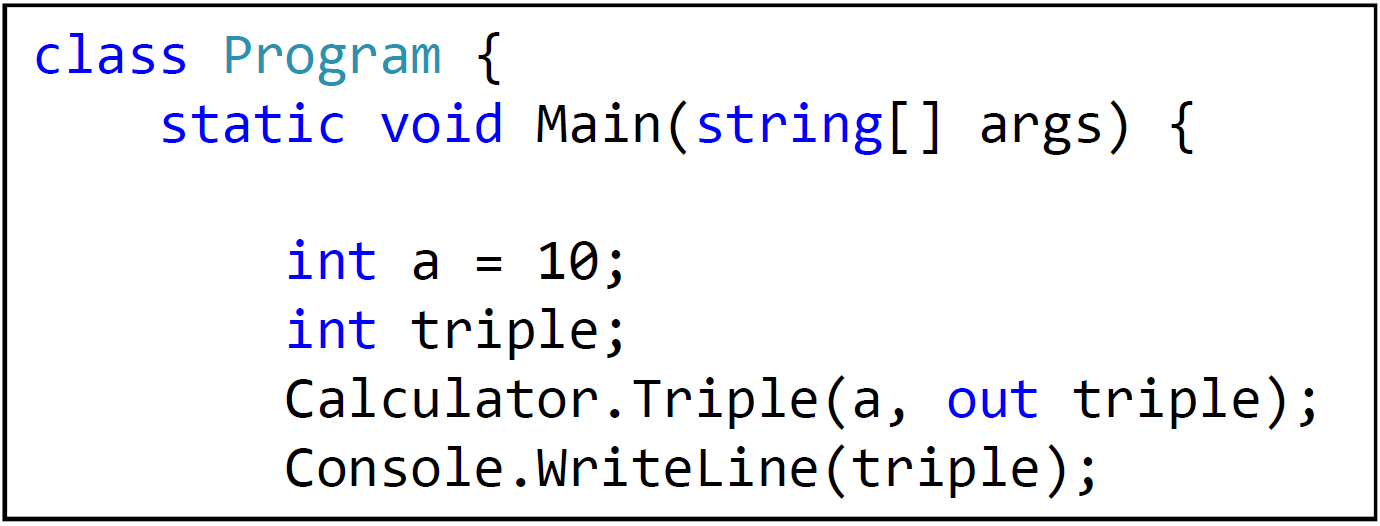
# Modificador out

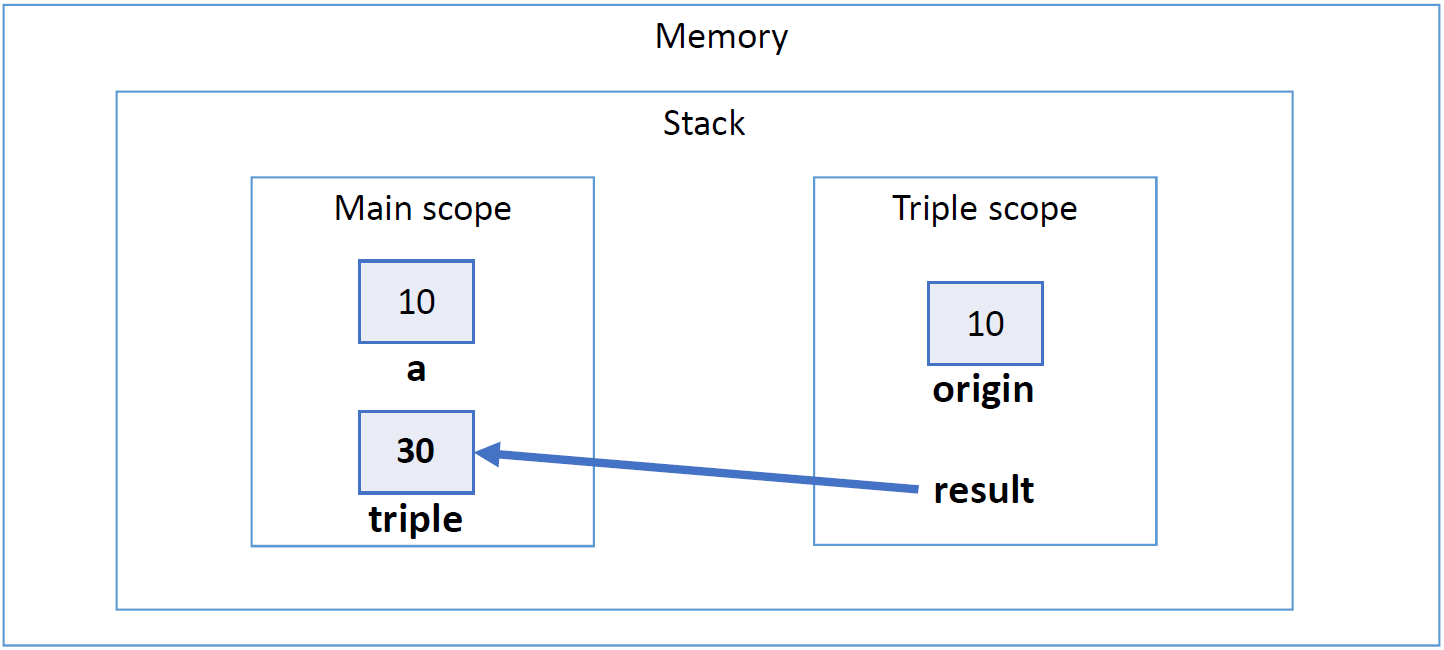
O modificador out é similar ao ref (faz o parâmetro ser uma referência para a variável original), mas não exige que a variável original seja iniciada.











# Considerações sobre ref e out

* Diferença:
  + A variável passada como parâmetro **ref** DEVE ter sido iniciada
  + A variável passada como parâmetro **out** não precisa ter sido iniciada

Conclusão: ambos são muito similares, mas ref é uma forma de fazer o compilador obrigar o usuário a iniciar a variável.

Nota: ambos são considerados "code smells" (design ruim) e devem ser evitados. Porém, as vezes ainda são necessários.