# Desalocação de memória – 19/11/2020

# garbage collector e escopo local

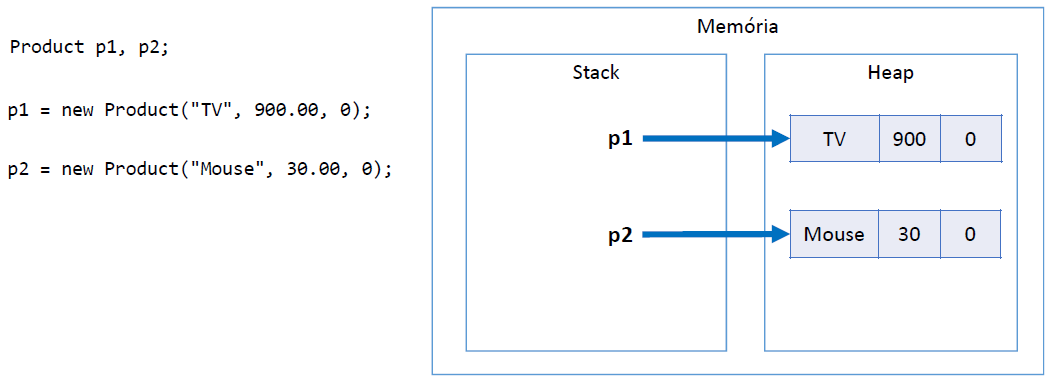
# Garbage collector

É um processo que automatiza o gerenciamento de memória de um programa em execução.

O garbage collector monitora os objetos alocados dinamicamente pelo programa (no heap), desalocando aqueles que não estão mais sendo utilizados.

# Desalocação por garbage collector

Suponhamos que foi declarado duas variáveis, p1 e p2. A representação na memória seria essa:

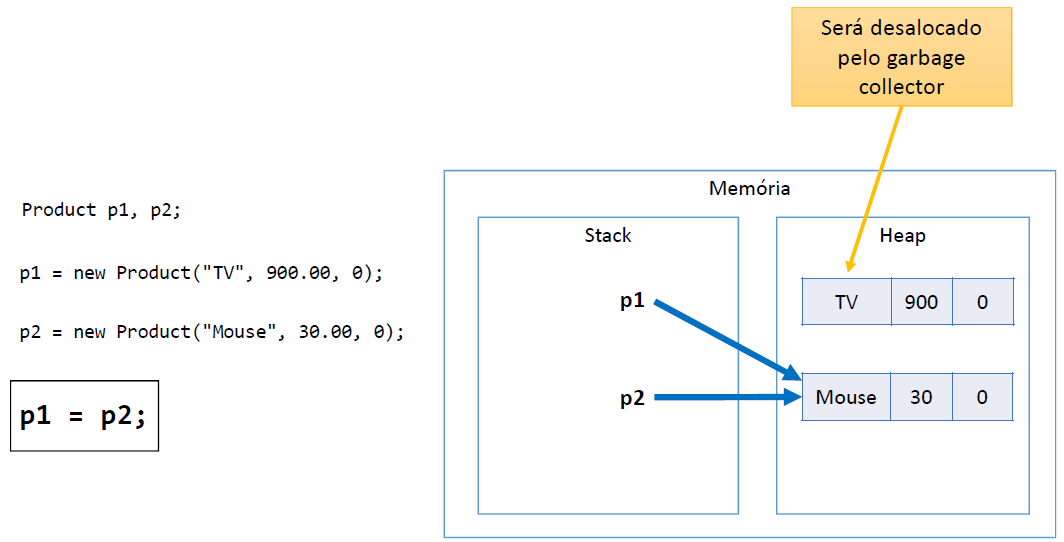


Agora, imagine que p1 recebeu p2.

p1 passa a apontar para onde p2 está apontando.

O objeto “ TV, 900, 0” perde toda referência.

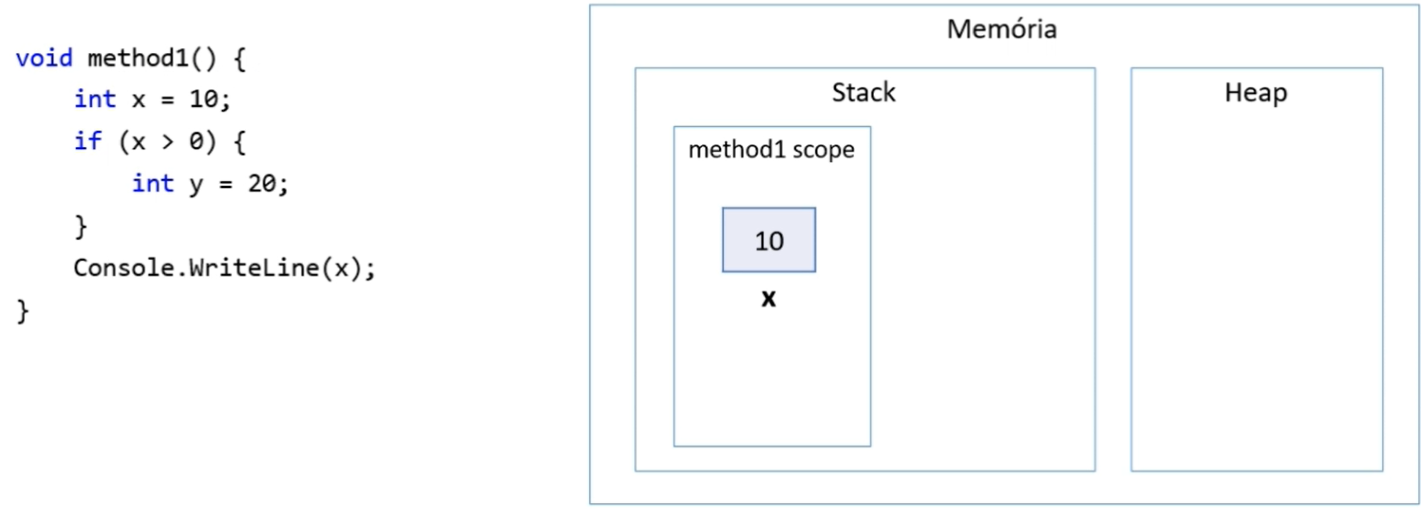
Com isso, o garbage collector, detecta a ausência de referência e desaloca isso da memória lá no Heap.



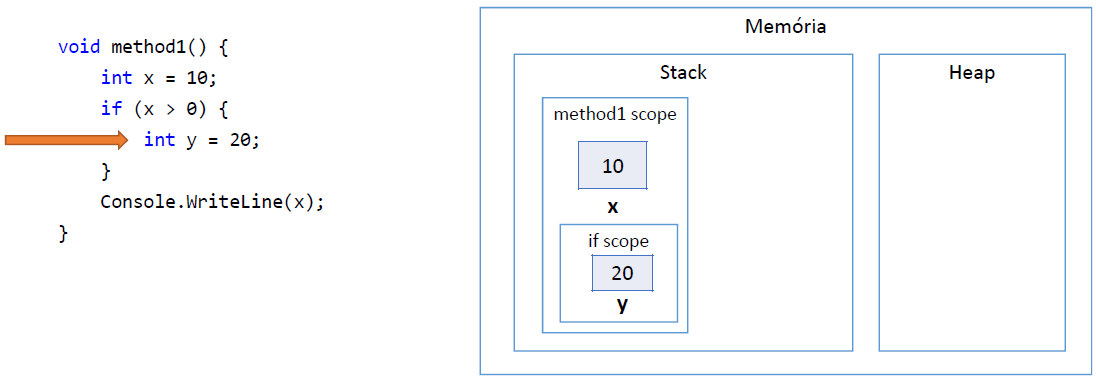
Resumindo. Quando um objeto perde a referência, o garbage collector “esvazia” desaloca a memória em um momento futuro.

# Desalocação por escopo

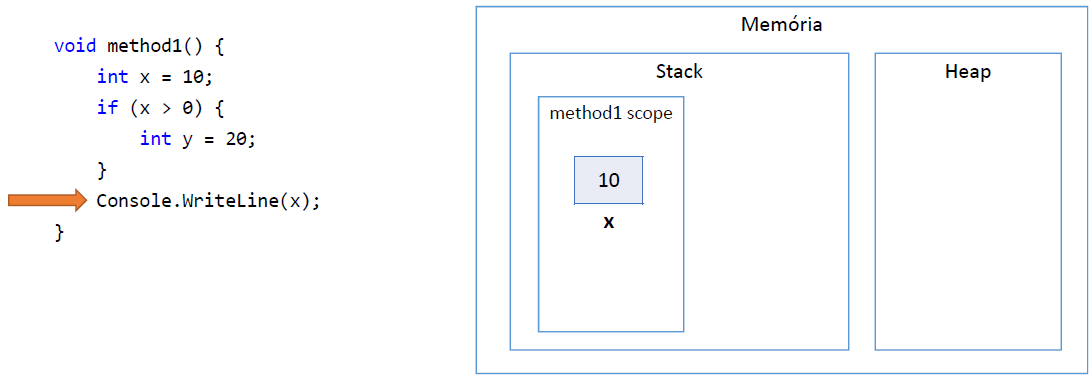
Imagine um método. Nesse método, quando x recebe 10. A memória se comporta da forma da imagem abaixo. No stack da memória é criado um escopo, e dentro desse escopo, uma caixa recebe o valor 10.



Quando o programa entra dentro do if. É criado a variável y recebendo o valor 20.

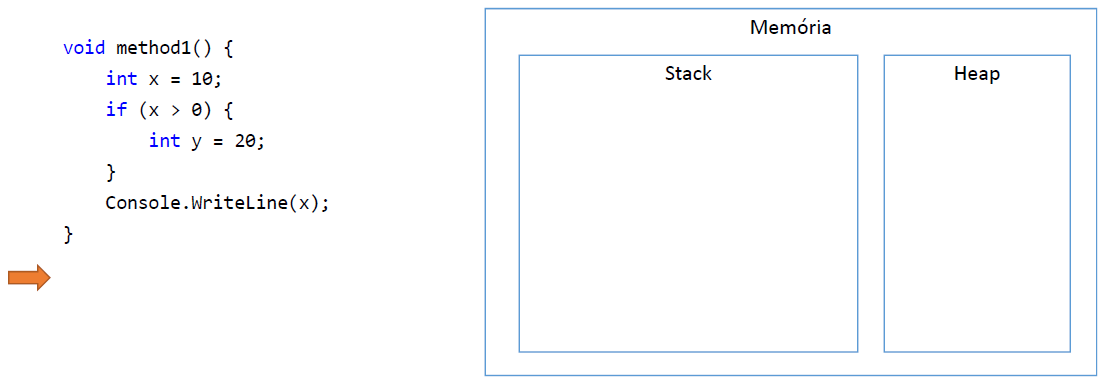


Porém, depois que o if é executado e o programa sair dele. Imediatamente a variável y some de dentro da memória.



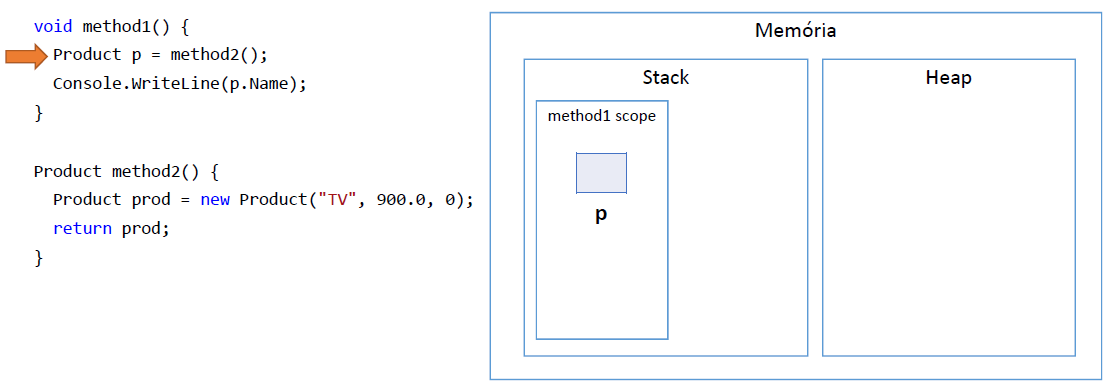
A desalocação por escopo se comporta assim. Assim que saí da estrutura, a memória é desalocada imediatamente. Diferente do garbage collector.

Saiu do método. A variável x também é desalocada.

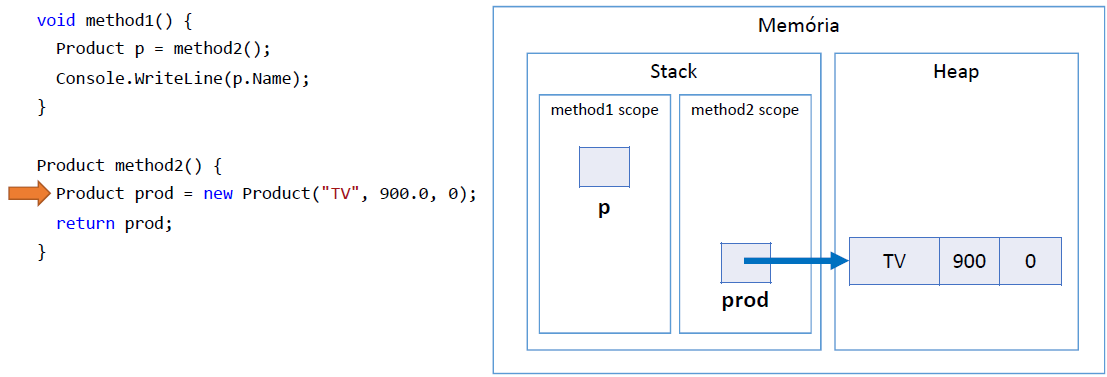


# Outro exemplo

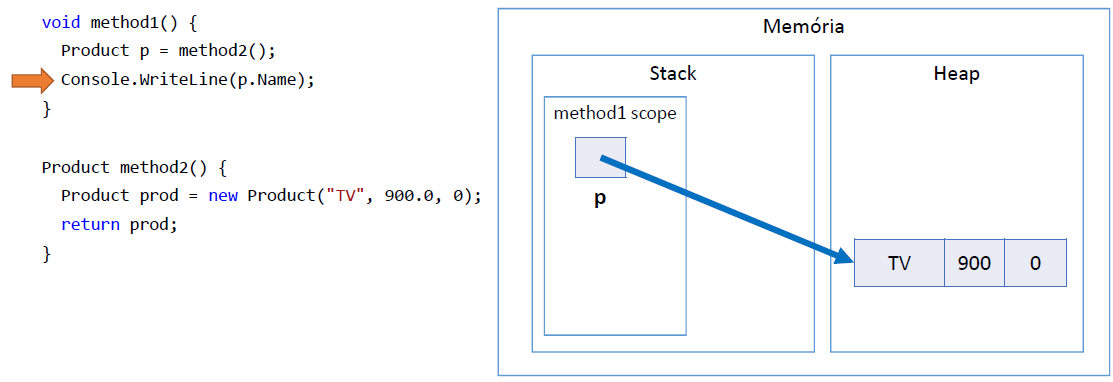
Nesse exemplo. Ao iniciar o programa, entrando no método 1. A variável é criada dentro do escopo do método no Stack da memória.



Como aqui, houve uma chamada para o método 2. A execução é desviada para a criação de uma outra variável dentro do escopo do método 2. A variável prod foi criada e recebe o objeto que foi instanciado.



No return, a variável p recebe a referência que prod possuía. Imediatamente, o escopo e todas as suas variáveis são eliminadas da memória.



Ou seja. O objeto “TV, 900, 0” foi criado no Heap através da instanciação do método 2. Porém, como o método 2 retorna o resultado para a variável p. O objeto continua existindo dentro do Heap, sendo referenciado pela variável p. O objeto não será desalocado pelo garbage collector.

# Resumo

• Objetos alocados dinamicamente, quando não possuem mais referência para eles, serão desalocados pelo garbage collector em um momento futuro.

• Variáveis locais são desalocadas imediatamente assim que seu escopo local sai de execução.