



Desalocação de memória e Garbage Collector

Garbage Collector

É um processo que automatiza o gerenciamento de memória de programas em execução

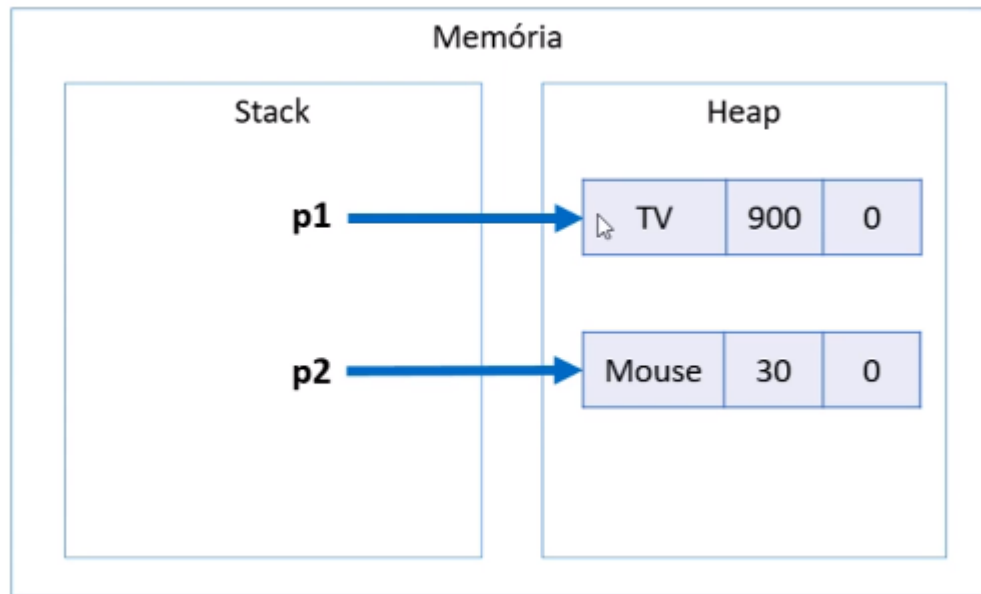
Ele monitora os objetos alocados pelo programa (no heap), e desaloca aqueles que não estão sendo mais utilizados.

Desalocação de memória dinâmica

Vamos supor que temos os seguintes objetos criados:

```
Product p1, p2;  
  
p1 = new Product("TV", 900, 0);  
p2 = new Product("Mouse", 30, 0);
```

A alocação de memória ficaria da seguinte forma:

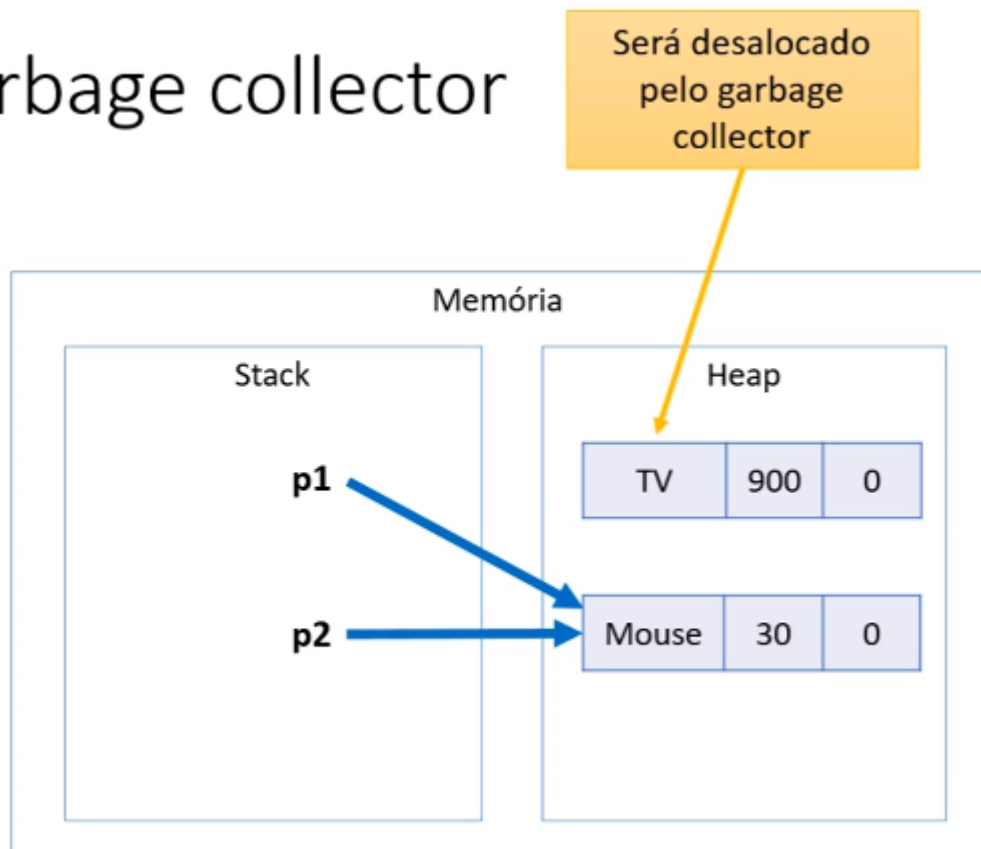


Mas e se eu fizesse o seguinte?

```
p1 = p2;
```

O garbage collector iria desalocar da memória o antigo p2 estanciado que não está mais sendo usado:

rbage collector



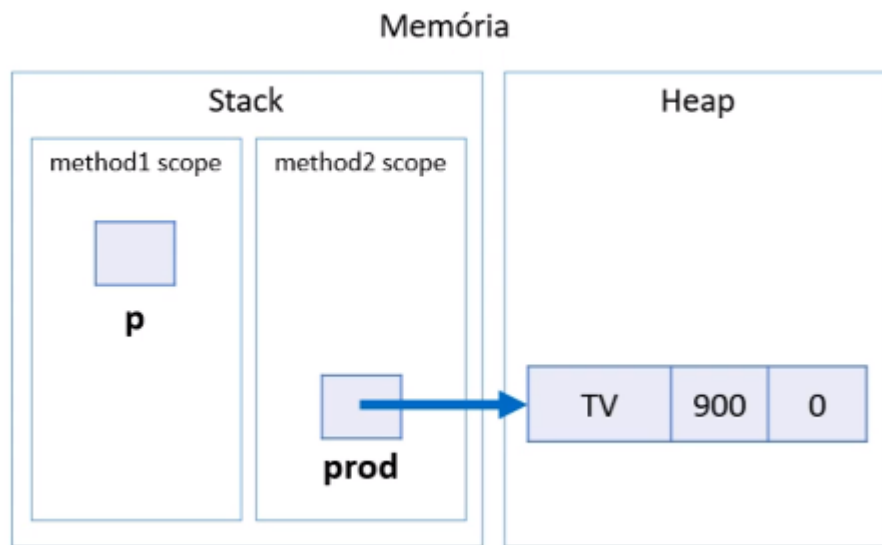
Desalocação por escopo

Vamos ao seguinte exemplo:

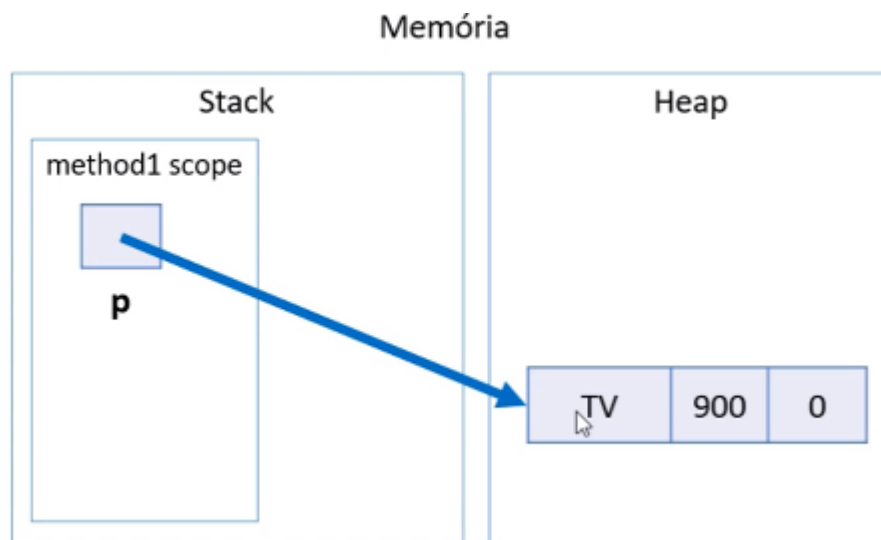
```
void method1()
{
    Product p = method2();
    Console.WriteLine(p.Name);
}

Product method2()
{
    Product prod = new Product("TV", 900.0, 0);
    return prod;
}
```

Ao executar o primeiro método que consecutivamente chama o segundo, a alocação de memória funciona da seguinte forma:



Após a execução do método 2, a alocação de memória se comporta da seguinte forma:



Podemos ver que o espaço em memória destinado pro método 2 foi desalocado da memória, pois ele já foi executado. (O escopo dele foi finalizado)

Porém o objeto criado por ele que ficou guardado na variável 'p' do primeiro método, e agora é essa variável que aponta pra referência de memória no heap.

Resumo

Objetos alocados **dinamicamente**, quando não possuem mais referência para eles, serão desalocados pelo garbage collector

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