

1 Recolección de lixo

1.1 O recolector de lixo

En Java o recolector de lixo, un proceso que se executa dentro da JVM, proporciona xestión automática de memoria. Todos os obxectos "viven" no *heap* que é a área de memoria usada para a asignación dinámica durante a execución dun programa. Cando se quere acceder a unha rexión de memoria asignada, é dicir, un obxecto, faise indirectamente, normalmente por medio dunha referencia a ese obxecto (unha variable do tipo do obxecto determinado).

O traballo do recolector de lixo é atopar e borrar obxectos no *heap* que non teñen ningunha referencia, é dicir, que non teñen ningunha variable apuntándoos. Unicamente a JVM decide exactamente cando executar o recolector de lixo. **O programador só pode recomendar cando executar o recolector de lixo** facendo a seguinte chamada:

```
System.gc();
```

A chamada ao método `gc` só lle suxire á JVM que faga un esforzo extra para liberar a memoria o antes posible, pero no executa o recolector explicitamente.

Antes de que un obxecto poda ser borrado polo recolector de lixo debe ser considerado como **elixible**. Un obxecto é elixible cando ningún thread activo do programa pode acceder a el, é dicir, cando non existe ningunha variable que referencia a ese obxecto.

A existencia da xestión automática de memoria non implica que as aplicacións Java non podan acceder a partes da memoria non asignadas (por exemplo, cando se accede a un índice dun array fóra de rango) Existen dúas formas de desreferenciar un obxecto e, polo tanto, conseguir que sexa elixible:

- Facer que a variable correspondente sexa `null`.
- Facer que a variable correspondente apunte a outro obxecto