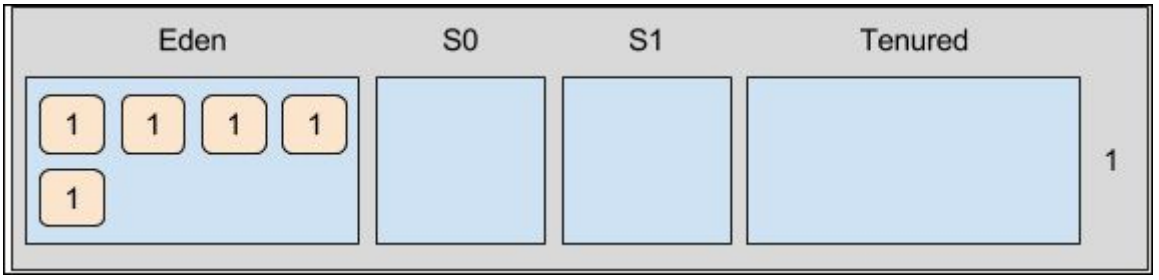
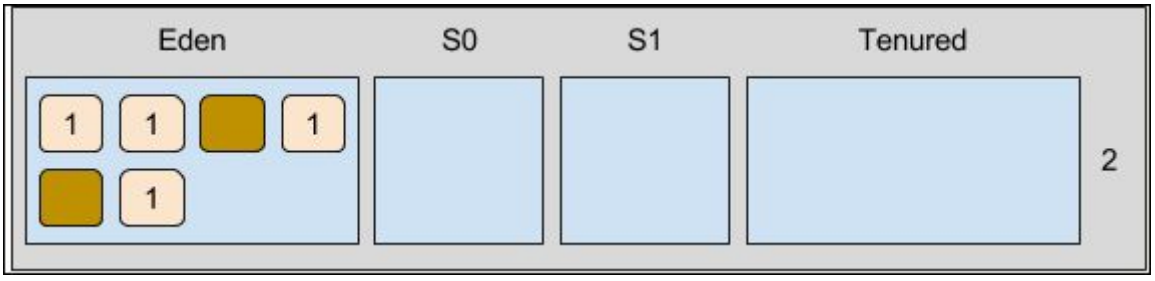
Gestión de memoria JVM con la GC

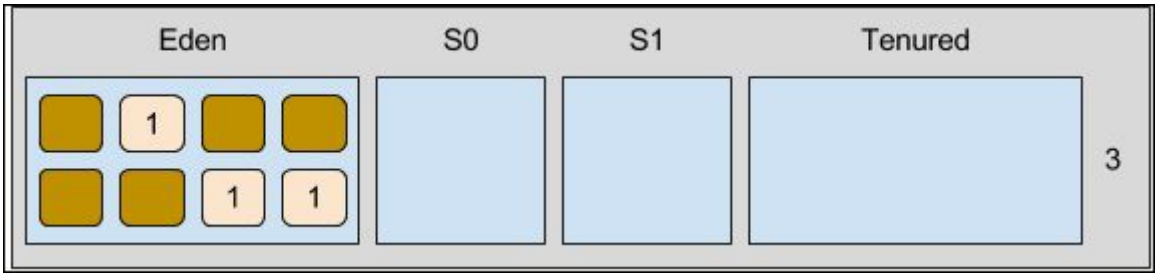
En los siguientes diagramas, disponemos de las áreas de memoria heap, **Eden**, **S0**, **S1,**y **Tenured,** repetidos en las filas numeradas **del 1** al **13** (al lado derecho de cada fila). Cada fila representa cómo las áreas de memoria se ven en un momento determinado la edad.
Los objetos asignados en las áreas de memoria se representan con los números dentro de ellos, lo que denota la generación o la edad que pertenecen a. Generación es una medida de la cantidad de veces que un objeto ha mudado de un área de memoria a otra. Una generación se completa cuando un área de memoria se llena con objetos y necesita mover todos sus objetos vivos a la siguiente área de memoria.
En el paso **1** en el siguiente diagrama, los objetos se están creando y por lo tanto comienzan a asignar memoria en **el**Eden.Nuevas estructuras de datos siempre comienzan su vida en **el**Eden,y su número generacional se establece en 1.



El heap en generacional edad 1
Paso **2** nos muestra más objetos habitación en Eden reparto y algunos objetos que ya no están en uso. Estos se muestran en el siguiente diagrama como objetos sin números. La memoria que ya no está en uso se recogerá en la próxima GC. Por ahora, sin embargo, todas estas áreas están todavía encerrados y sin embargo no puede ser de- o re-asignado.



El heap en el paso 2
A medida que alcanzamos el paso 3, **el Eden** está lleno y no hay más asignaciones se puede hacer aquí. Eden es, sin embargo, donde se hacen nuevas asignaciones, como se muestra en el diagrama siguiente:

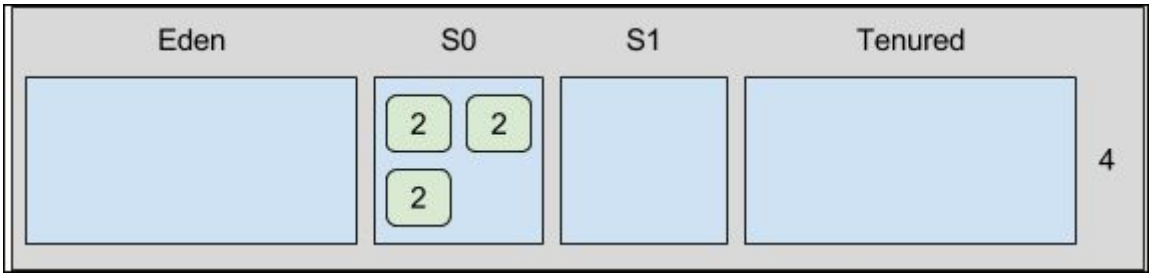


El heap en el paso 3

Ahora, entre las etapas **3** y **4,**una colección menor se lleva a cabo por GC y, después de eso, como podemos ver en el paso **4** en el siguiente diagrama, los objetos vivos se han movido en **S0** y todo el **Eden** barrida a fin de recibir nuevas asignaciones. Nótese cómo todos los objetos activos tienen ahora su número generacional incrementa a 2.

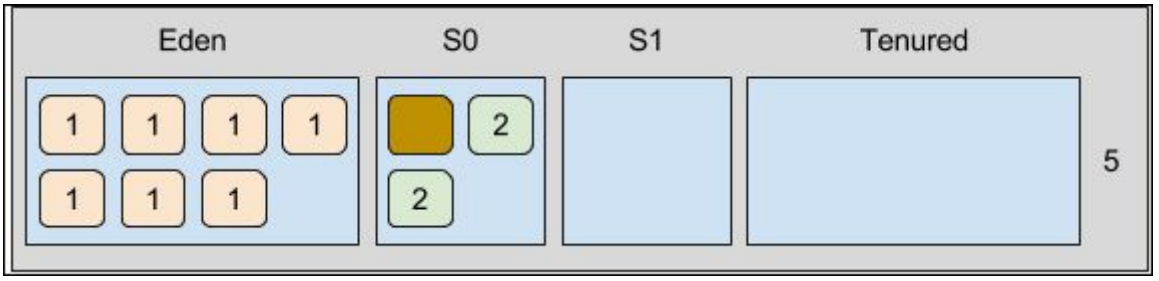
Tip

Un GC young o minor es una colección que consiste en movimiento (también conocido como el envejecimiento o la promoción de) los objetos de un área de memoria a otra, así como la liberación de memoria mediante la eliminación de todos los objetos ya no se hace referencia en la generación joven (Eden, S0, o S1).



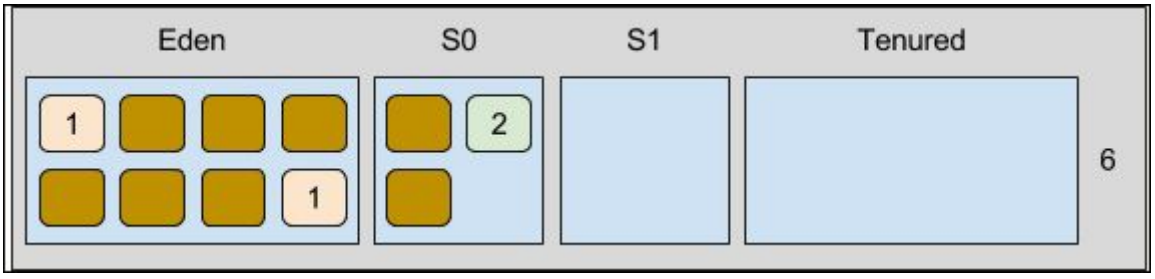
El heap en el paso 4

En el paso **5,**como se muestra en el siguiente diagrama, este comienza de nuevo. Ahora, la generación de **1** asignaciones se realizan en el Eden y, en **S0,**una asignación ya no está vivo.



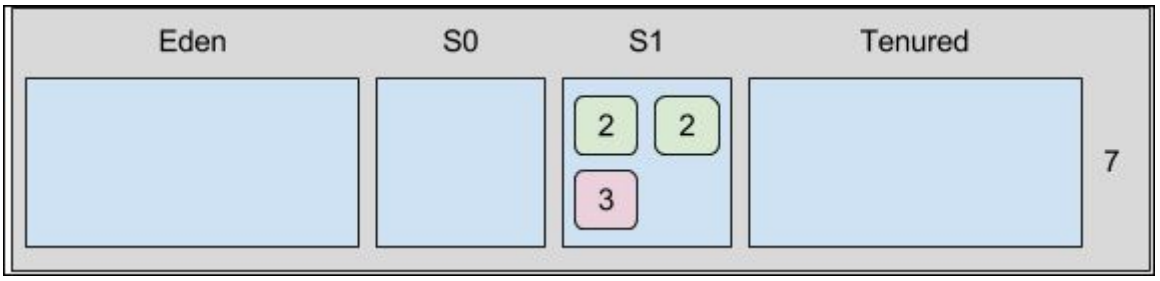
El heap en el paso 5

Paso **6** nos muestra cómo el Eden, una vez más es completa y el número de asignaciones no son en vivo. En S0, otra asignación ya no está vivo, como se muestra en el diagrama siguiente:



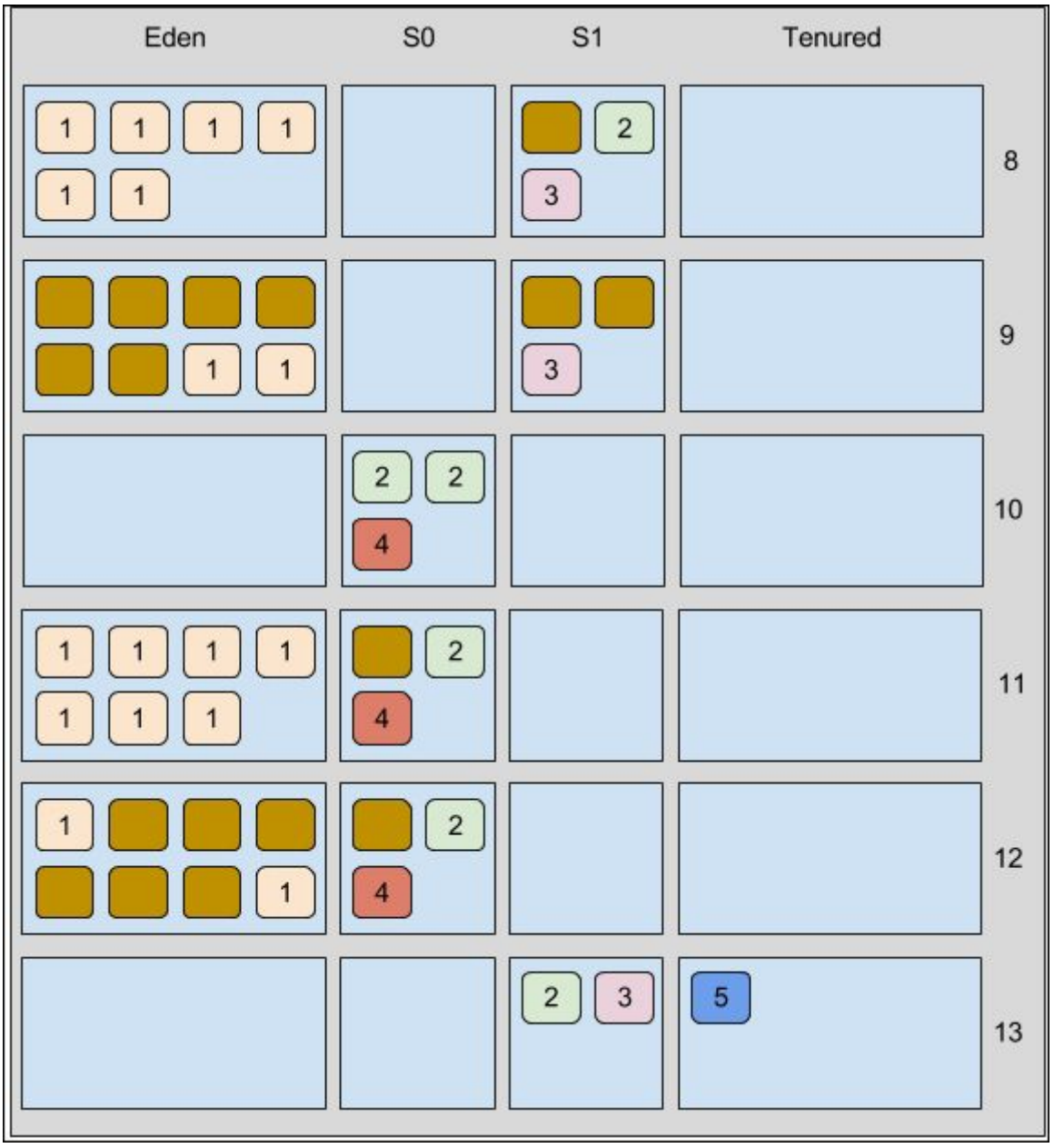
El heap en el paso 6

entrepasos **6** losy **7,**otro ejecuta GC, los objetos en movimiento en vivo desde S0 a S1, la mejora de su número generacional de **2** a **3,**y también el movimiento y la actualización de los objetos vivos de Eden, para S1, la limpieza de este modo tanto el Eden y S0. Esto se muestra en el diagrama siguiente:



El heap en el paso 7

Este comportamiento se repite entonces en los pasos siguientes. Como un área de memoria se llena, GC recibe la orden de trabajo. Objetos vivos se mueven de Eden, un espacio de supervivencia de trabajo se convierte en un espacio de supervivencia libre, y su edad generacional se incrementa un paso cada vez. Esto continúa hasta que se alcanzó el límite de edad titular. Cuando es, todas las asignaciones de esta edad se mueven en el área de memoria titular como se representa en el paso **13** en el siguiente diagrama, donde se pretende que la edad tenured es de cuatro.



El heap en la etapa 8-13

medida que la generación tenured finalmente se llena, un GC completa se lleva a cabo para recoger cualquier asignaciones en el mismo que ya no están en vivo.

Tip

Una GC old, major, o Full GC mediante la eliminación de todos los objetos ya no se hace referencia en el old (permanent) generation.

La realización de una Full GC es muy costoso en comparación con la recogida en el **Eden**. Desde la perspectiva del rendimiento, es, por lo tanto, deseable tener su mayoría objetos y objetos relativamente de corta duración que no son promovidos al espacio titular demasiado rápido. A menudo es también deseable mantener los objetos de larga vida (que lo hacen en el espacio permanencia) con vida el mayor tiempo posible.