

Minor GC

1. Let us assume that there are already objects on the Eden space when we start(Blocks 01 to 06 marked as used memory)
2. The application creates a new object(07)
3. JVM tries to get required memory from Eden space, but there is no free space in Eden to accommodate our object and hence JVM triggers minor GC

Minor GC



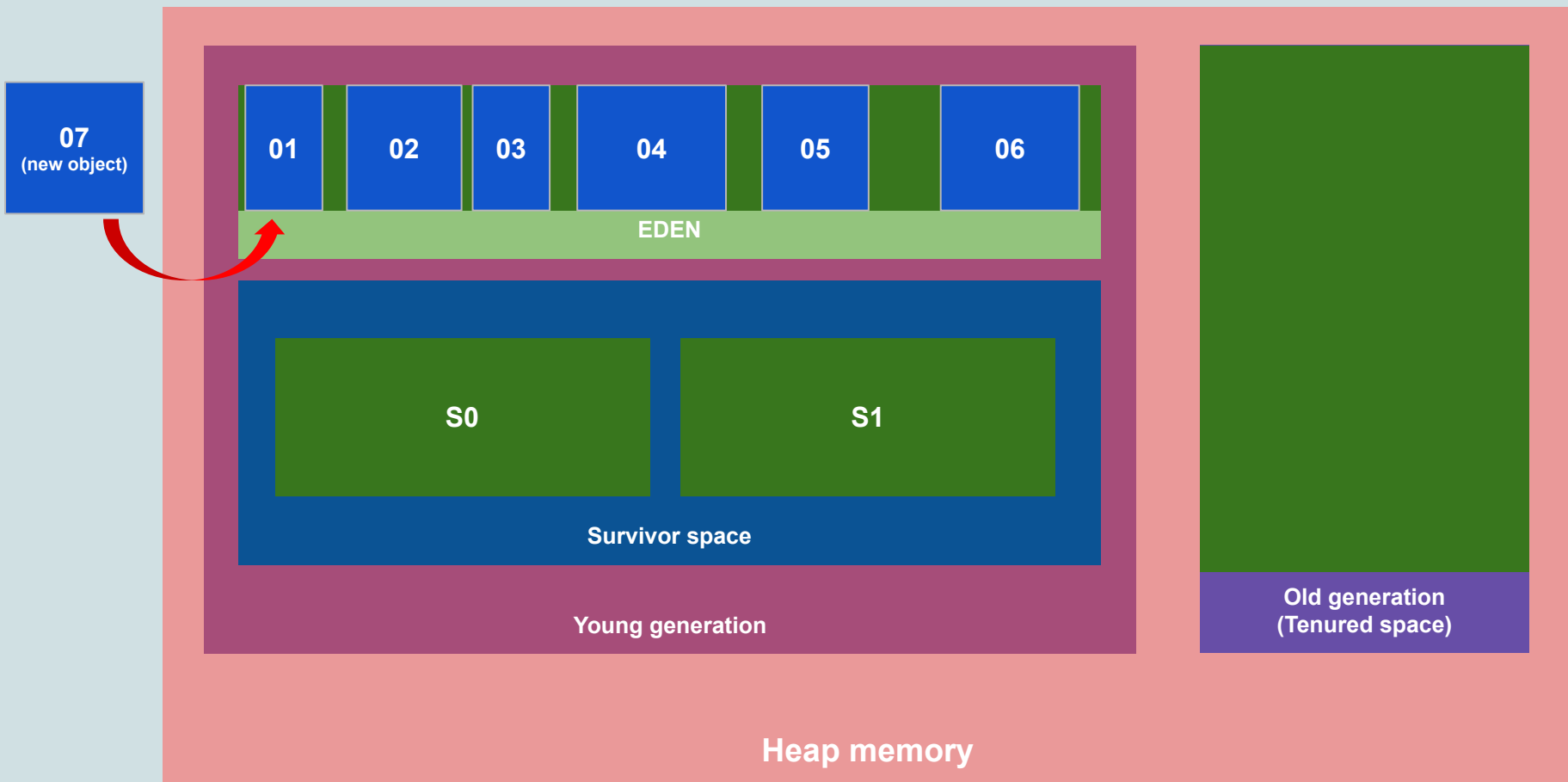
Orphan



Used memory



Free memory



Minor GC



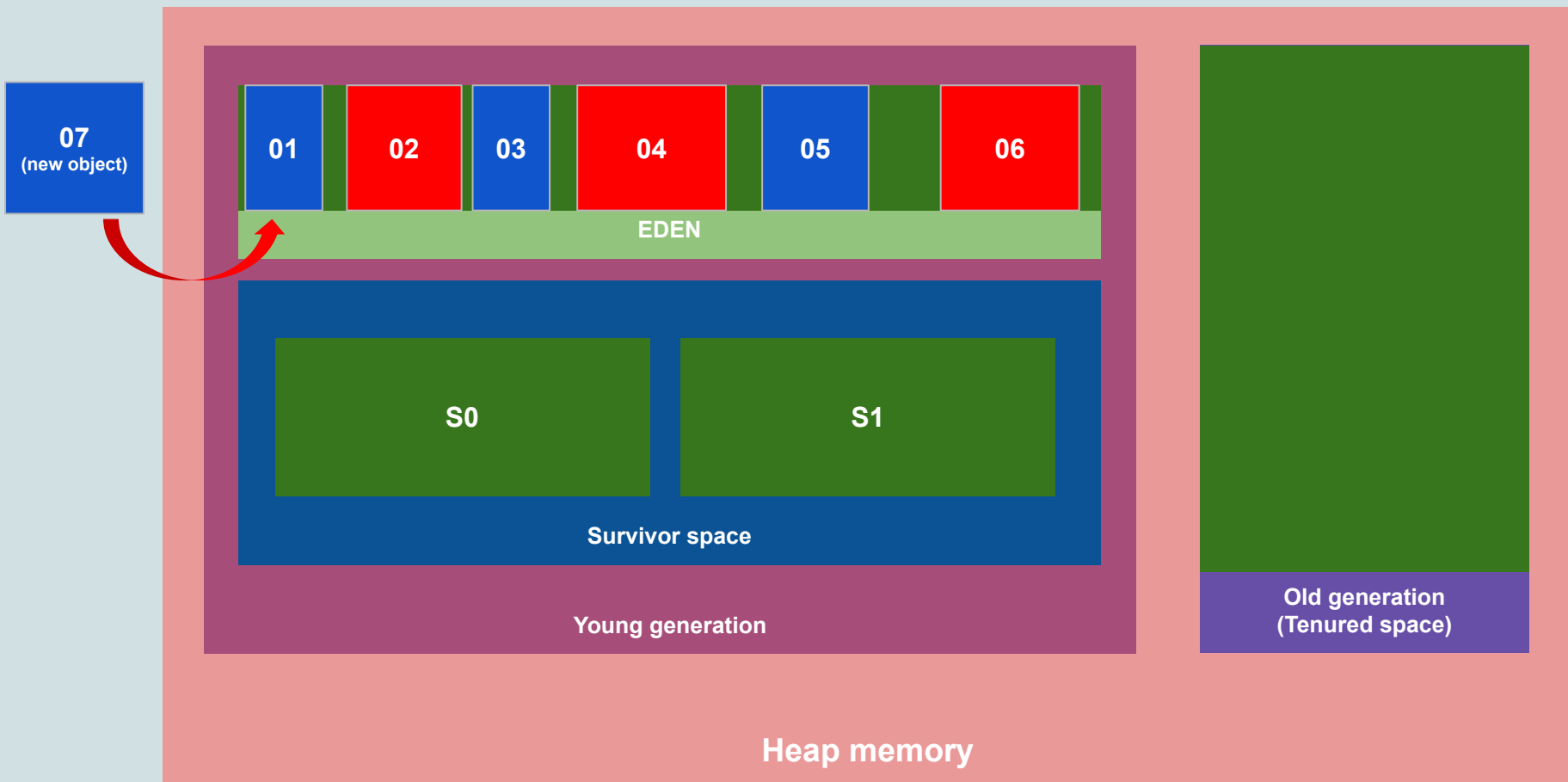
Orphan



Used memory



Free memory



Minor GC



Orphan

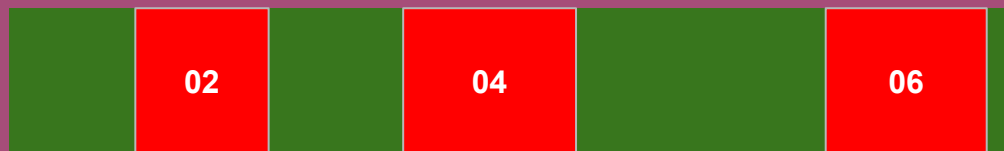


Used memory

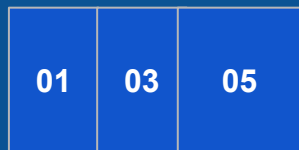


Free memory

07
(new object)



EDEN



To survivor space

S1

Survivor space

Young generation



Old generation
(Tenured space)

Heap memory

Minor GC



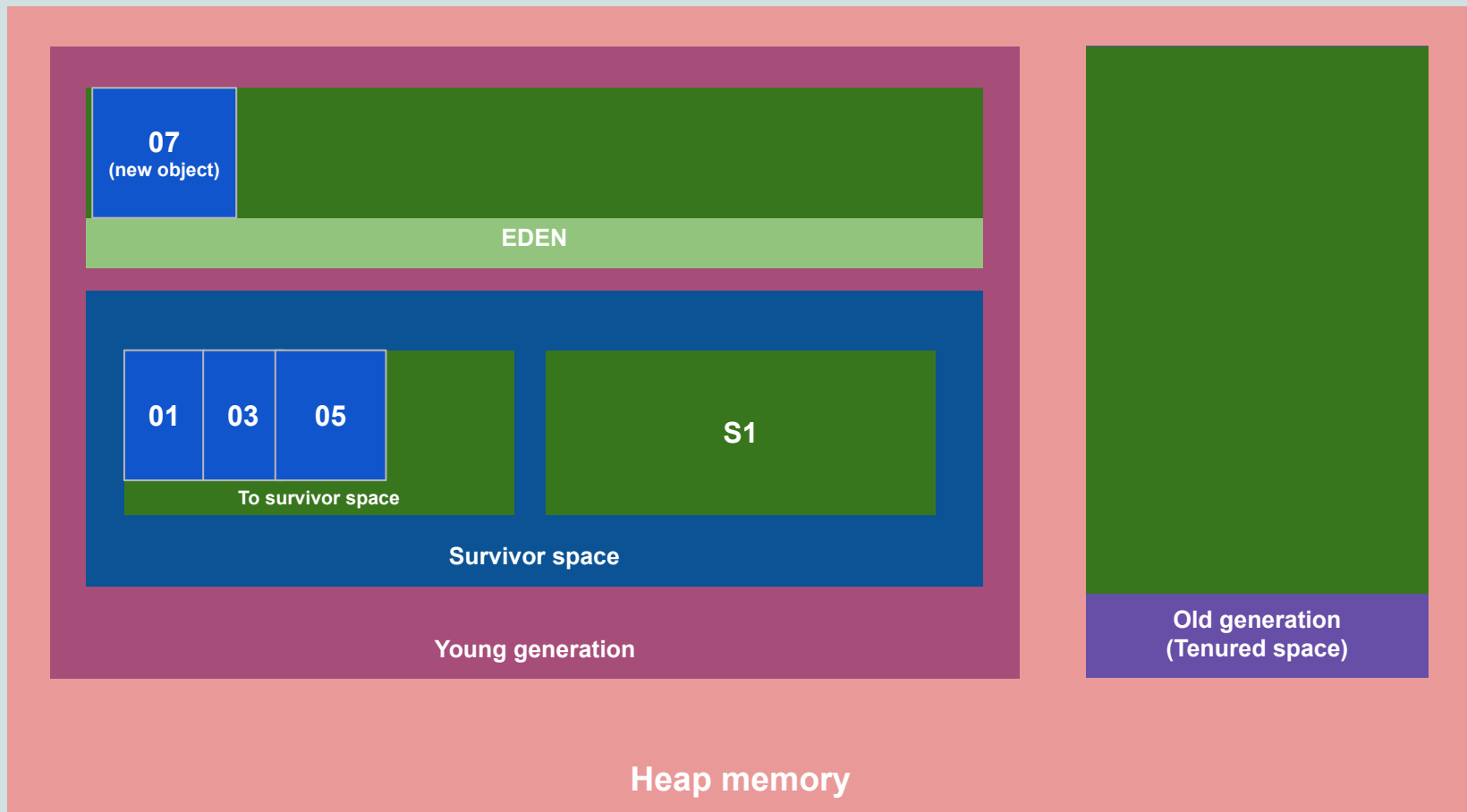
Orphan



Used memory



Free memory



Second Minor GC

1. Let us assume that some time have passed and there are more objects on the Eden space now(Blocks 07 to 13 marked as used memory)
2. The application creates a new object(14)
3. JVM tries to get required memory from Eden space, but there is no free space in Eden to accommodate our object and hence JVM triggers second minor GC

Minor GC



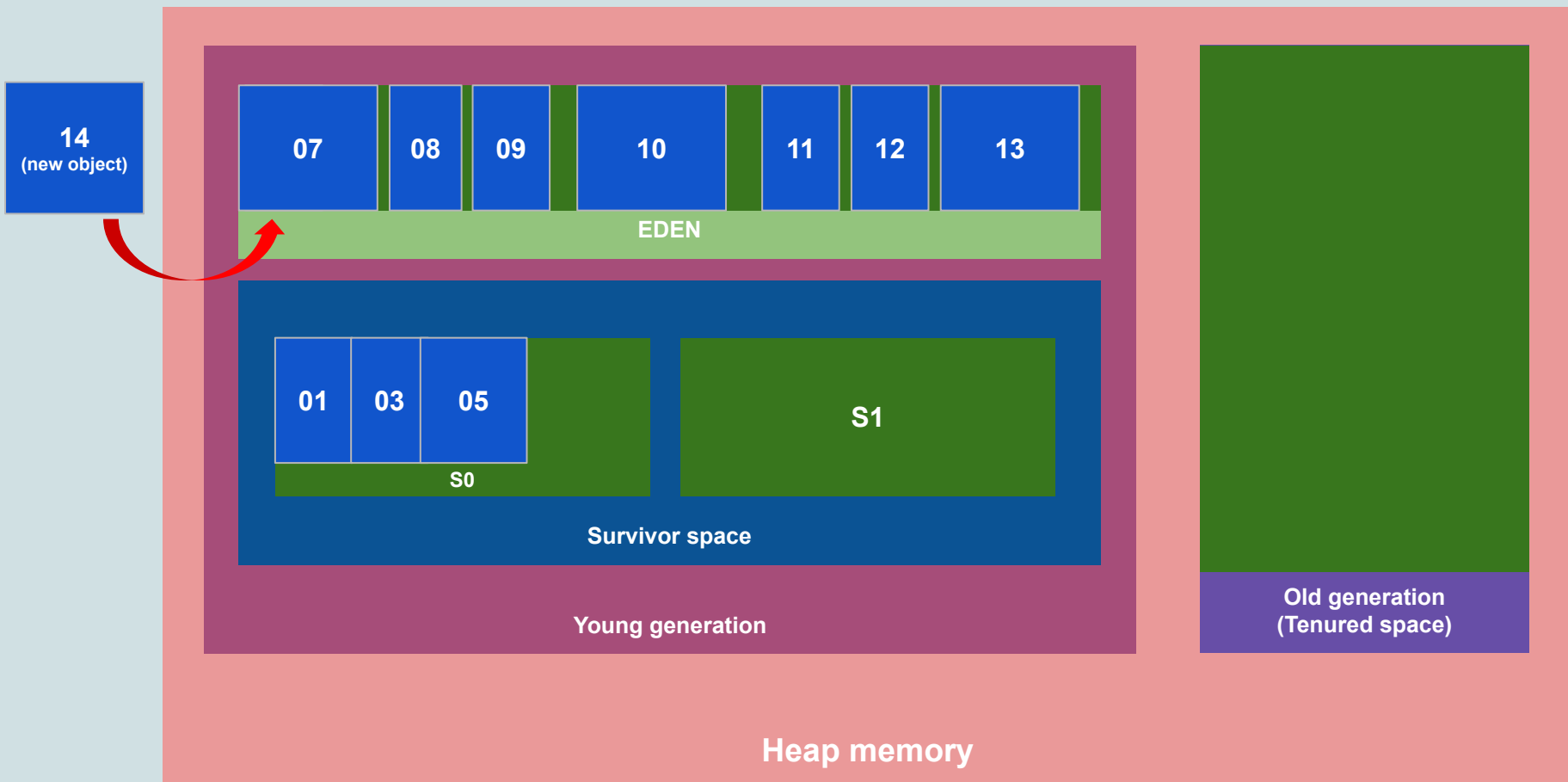
Orphan



Used memory



Free memory



Minor GC



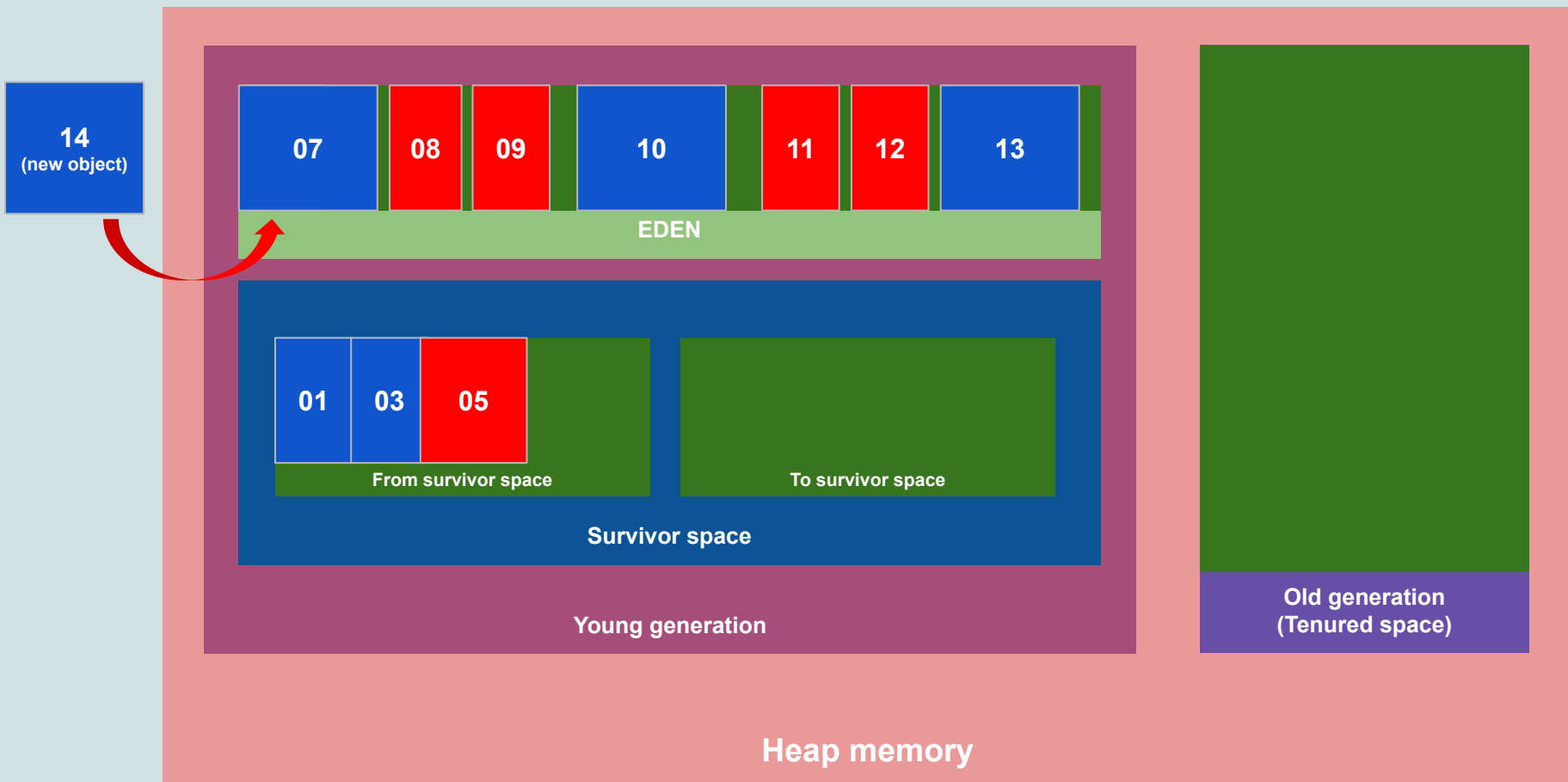
Orphan



Used memory



Free memory



Minor GC



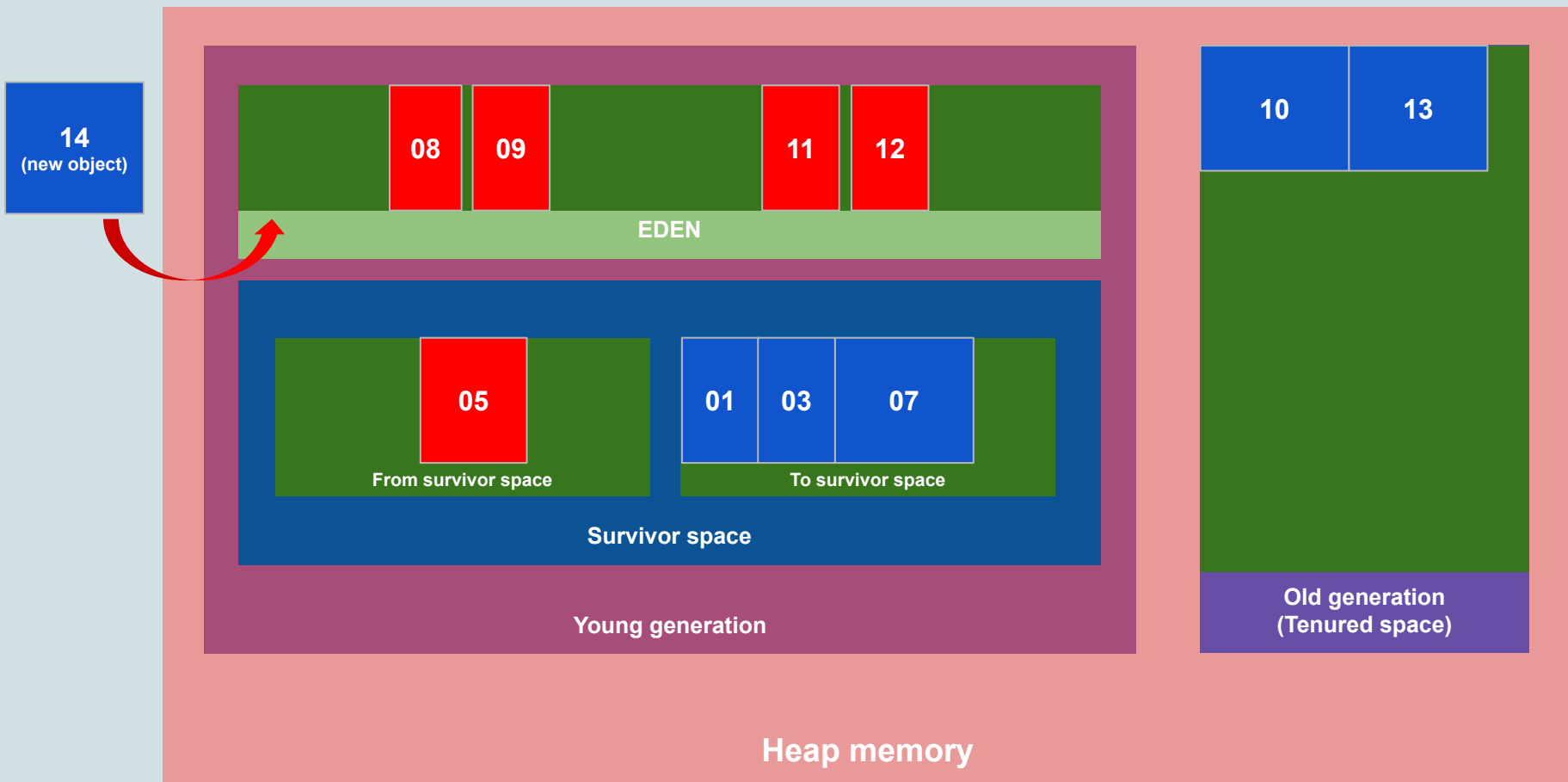
Orphan



Used memory



Free memory



Minor GC



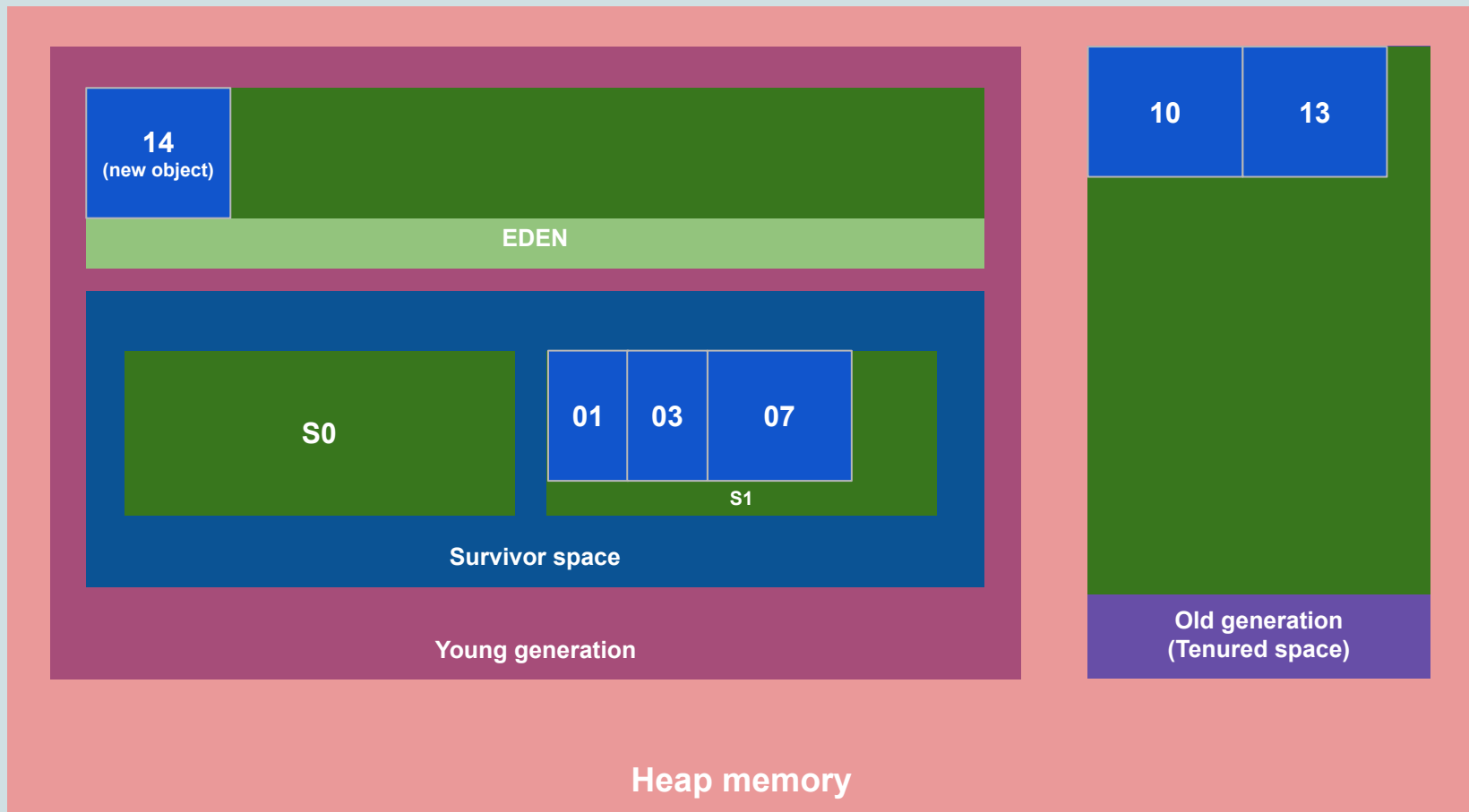
Orphan



Used memory



Free memory



13 more Minor GC cycles later

1. This keeps on repeating for each minor GC and the survivors are shifted between S0 and S1 and their age is incremented. Once the age reaches the "max age threshold", 15 by default, the object is moved to the "Tenured space"

Minor GC



Orphan



Used memory



Free memory

