

JVM把年轻代分为了三个部分，1个Eden区和2个survivor区（分别叫做from和to）。默认比例为8:1:1，为什么是这个比例？

答：一般情况下，新创建的对象都会被分配到eden区，这些对象经过第一次Minor GC后，如果仍然存活，将会移动到survivor区。对象在survivor区中每熬过一次Minor GC，年龄就会增加1岁，当它的年龄增加到一定程度时，就会被移动到老年代中。

因为年轻代中的对象基本都是朝生夕死（80%以上），所以设为8:1:1，因此年轻代的垃圾回收算法是用的复制算法，复制算法的基本思想是将内存分为两块，每次只用其中一块，当这一块内存用完，就将还活着的对象复制到另外一块上面。复制算法不会产生内存碎片。

在GC开始的时候，对象只会存在于Eden区和名为“From”的Survivor区，Survivor区“To”是空的。紧接着进行GC，Eden区中所有存活的对象都会被复制到“To”，而在“From”区中，仍存活的对象会根据他们的年龄值来决定去向。年龄达到一定值(年龄阈值，可以通过-XX:MaxTenuringThreshold来设置)的对象会被移动到老年代中，没有达到阈值的对象会被复制到“To”区域。经过这次GC后，Eden区和From区已经被清空。这个时候，“From”和“To”会交换他们的角色，也就是新的“To”就是上次GC前的“From”，新的“From”就是上次GC前的“To”。不管怎样，都会保证名为To的Survivor区域是空的。Minor GC会一直重复这样的过程，直到“To”区被填满，“To”区被填满之后，会将所有对象移动到老年代中。



