# JVM内存的分代模型

JVM内存的一个分代模型：年轻代、老年代、永久代。

VM将Java堆内存划分为了两个区域，一个是年轻代，一个是老年代。

年轻代: 创建和使用完之后立马就要回收的对象存放的位置

老年代：创建之后需要一直长期存在的对象放在里面

为什么要分成年轻代和老年代？

因为这跟垃圾回收有关，对于年轻代里的对象，他们的特点是创建之后很快就会被回收，所以需要用一种垃圾回收算法

对于老年代里的对象，他们的特点是需要长期存在，所以需要另外一种垃圾回收算法，所以需要分成两个区域来放不同的对象。

什么是永久代？

很简单，JVM里的永久代其实就是我们之前说的方法区。

大部分的正常对象，都是优先在新生代分配内存的。

JVM就有一条规定了

如果一个实例对象在新生代中，成功的在15次垃圾回收之后，还是没被回收掉，就说明他已经15岁了。

这是对象的年龄，每垃圾回收一次，如果一个对象没被回收掉，他的年龄就会增加1。

如果一个对象在新生代中成功躲过10多次垃圾回收，成为一个“老年人”，那么就会被认为

是会长期存活在内存里的对象。

然后他会被转移到Java堆内存的老年代中去，顾名思义，老年代就是放这些年龄很大的对象。

**老年代会垃圾回收吗？**

答案是**肯定的**，因为老年代里的对象也有可能随着代码的运行，不再被任何人引用了，就需要被垃圾回收。