

# 强引用、软引用、弱引用、幻象引用有什么区别？

作者：[zyjImmortal](#)

原文链接：<https://ld246.com/article/1586072983683>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



在 Java 语言中，除了原始数据类型的变量，其他所有都是所谓的引用类型，指向各种不同的对象，Java 根据其生命周期的长短，将引用分为4类，分别是强引用、弱引用、软引用和幻象引用(虚引用)，不同的引用类型，主要体现的是对象不同的可达性状态和对垃圾回收集的影响。

## 强引用

- 什么是强引用

`Object obj = new Object()`，这里的obj就是强引用，通过关键字new创建的对象所关联的引用就是强引用。只要有强引用指向一个对象，就能表明对象还活着，垃圾收集器就不会触碰这种对象，当JV内存空间不足，JVM宁愿抛出`OutOfMemoryError`运行时错误（OOM），使程序异常终止，也不会随意回收具有强引用的“存活”对象来解决内存不足的问题。对于一个普通的对象，如果没有其他的用关系，只要超过了引用的作用域或者显式地将相应（强）引用赋值为 `null`，就是可以被垃圾收集的，具体回收时机还是要看垃圾收集策略

## 软引用

- 什么是软引用

是一种相对强引用弱化一些的引用，可以让对象豁免一些垃圾收集，只有当 JVM 认为内存不足时，会去试图回收软引用指向的对象。

软引用通过`SoftReference`类实现。软引用的生命周期比强引用短一些。只有当 JVM 认为内存不足，才会去试图回收软引用指向的对象：即JVM 会确保在抛出 `OutOfMemoryError` 之前，清理软引用向的对象

- 有什么应用场景

软引用通常用来实现内存敏感的缓存。如果还有空闲内存，就可以暂时保留缓存，当内存不足时清理，这样就保证了使用缓存的同时，不会耗尽内存。

# 弱引用

- 什么是弱引用

并不能使对象豁免垃圾收集，仅仅是提供一种访问在弱引用状态下对象的途径。这就可以用来构建一没有特定约束的关系，比如，维护一种非强制性的映射关系，如果试图获取时对象还在，就使用它，则重现实例化

弱引用通过WeakReference类实现。弱引用的生命周期比软引用短。在垃圾回收器线程扫描它所管的内存区域的过程中，一旦发现了具有弱引用的对象，不管当前内存空间足够与否，都会回收它的内存

- 有什么应用场景

弱应用同样可用于内存敏感的缓存

# 幻象引用(虚引用)

- 什么是幻象引用

虚引用也叫幻象引用，通过PhantomReference类来实现。无法通过虚引用访问对象的任何属性或函数。幻象引用仅仅是提供了一种确保对象被 finalize 以后，做某些事情的机制。如果一个对象仅持有虚引用，那么它就和没有任何引用一样，在任何时候都可能被垃圾回收器回收

- 有什么应用场景

可用来跟踪对象被垃圾回收器回收的活动，当一个虚引用关联的对象被垃圾收集器回收之前会收到一系统通知