**前言**

从Java SE2开始，就提供了四种类型的引用：**强引用**、**软引用**、**弱引用**和**虚引用**。Java中提供这四种引用类型主要有两个目的：

* 第一是可以让程序员通过代码的方式决定某些对象的生命周期
* 第二是有利于JVM进行垃圾回收。

## 总结

| **引用类型** | **被回收时间** | **用途** | **生存时间** |
| --- | --- | --- | --- |
| 强引用 | 从来不会 | 对象的一般状态 | JVM停止运行时 |
| 软引用 | 内存不足时 | 对象缓存 | 内存不足时 |
| 弱引用 | jvm垃圾回收时 | 对象缓存 | gc运行后 |
| 虚引用 | 未知 | 未知 | 未知 |

在实际程序设计中一般很少使用弱引用与虚引用，使用软引用的情况较多，这是因为软引用可以加速JVM对垃圾内存的回收速度，可以维护系统的运行安全，防止内存溢出（OutOfMemory）等问题的产生。

利用软引用和弱引用解决OOM问题：假如有一个应用需要读取大量的本地图片，如果每次读取图片都从硬盘读取，则会严重影响性能，但是如果全部加载到内存当中，又有可能造成内存溢出，此时使用软引用可以解决这个问题。

设计思路是：用一个HashMap来保存图片的路径和相应图片对象关联的软引用之间的映射关系，在内存不足时，JVM会自动回收这些缓存图片对象所占用的空间，从而有效地避免了OOM的问题。