# 强引用，软引用，弱引用，幻象引用有什么区别?

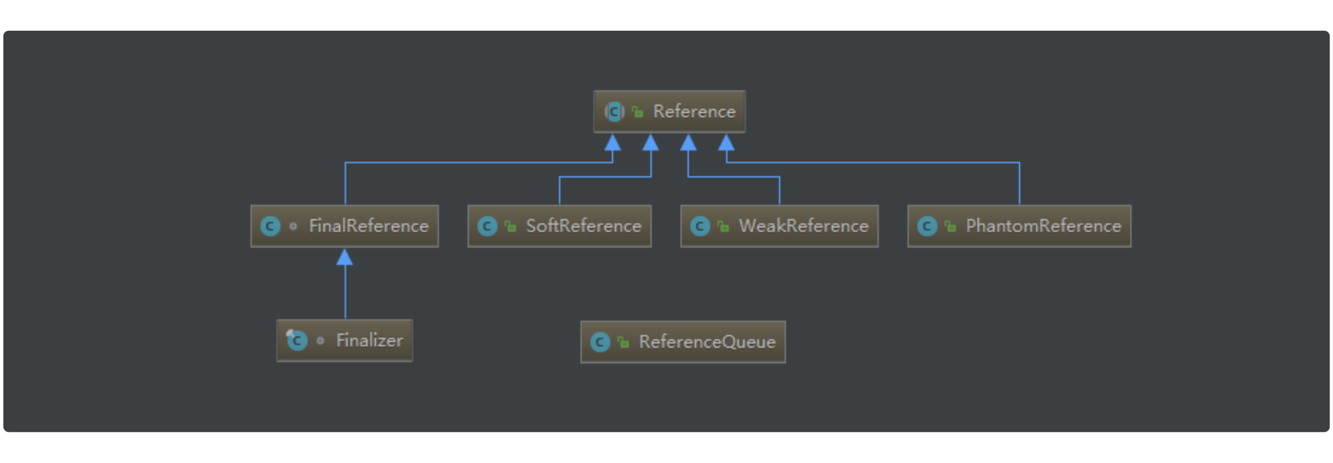
发布于2020-05-25 23:37:21 阅读 2K

当垃圾回收线程扫描所管辖的内存区域时，会通过判断有哪些类型的引用指向这内存中的对象，来决定是否回收这些对象

## 强引用，软引用，弱引用，幻象引用有什么区别?

不同的引用类型，主要体现的是对象的不同的可达性（reachable）状态和对垃圾收集的影响。

- 所谓强引用 (Strong Reference) 就是我们常见的普通对象引用，只要还有**强引用指向一个对象，就能表明对象还活着，垃圾回收不回收这种对象。**
- 软引用，是一种相对强引用弱化一些的引用，只有**当 JVM 认为内存不足时**，才会试图回收软引用指向的对象。JVM 会确保在排除 OutOfMemoryError 之前，清理软引用指向的对象，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，清理掉，这样就保证了使用缓存的同时，不会耗尽内存。
- 弱引用，比软引用拥有更短的生命周期，在垃圾回收线程扫码所管辖的内存区域的过程中，一大发现了只具有弱引用的对象，不管当前内存空间是否足够，都会回收它的内存，由于垃圾回收器是一个优先级，因此不一定很快发现那些只有弱引用的对象。
- 虚引用，**形同虚设**，虚引用不会决定对象的生命周期，如果一个对象仅持有虚引用，其实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。虚引用和软引用的一个区别是，虚引用必须和引用队列（ReferenceQueue）联合使用。当垃圾回收器准备回收一个对象时，如果发现它还有虚引用，就会在回收对象的内存之前，把这个虚引用加入到与之关联的引用队列中。



;

image

引用所指向的对象被垃圾回收时间

**四种引用类型对比**

| 引用类型 | 被垃圾回收时间 | 用途 | 生存时间 |
|---|---|---|---|
| 强引用 | 从来不会 | 对象的一般状态 | JVM 停止运行时终止 |
| 软引用 | 当内存不足时 | 对象缓存 | 内存不足时终止 |
| 弱引用 | 正常垃圾回收时 | 对象缓存 | 垃圾回收后终止 |
| 虚引用 | 正常垃圾回收时 | 跟踪对象的垃圾回收 | 垃圾回收后终止 |

### 四种引用的应用场景

**强引用**

强引用是使用最普遍的引用，一般声明如下：

```
Object strongReference = new Object();
```

如果要对强引用进行垃圾回收，需要设置强引用指向null,或者让其超出对象的生命周期范围，则认为改对象不存在引用。

让强引用指向null后，那它原所指的对象就减少了一个强引用

```
strongReference = null;
```

可以看下 ArrayList 是如何进行内存释放的

```
    public void clear() {
         modCount++;
         // clear to let GC do its work
         for (int i = 0; i < size; i++)
            elementData[i] = null;
         size = 0;
     }
```

**软引用**

软引用可用来实现内存敏感的高速缓存。

使用如下：

```
// 强引用
String strongReference = new String("abc");
// 软引用
String str = new String("abc");
SoftReference<String> softReference = new SoftReference<String>(str);
```

软引用可以和一个引用队列（ReferenceQueue）联合使用，如果软引用所引用的对象被垃圾回收，Java虚拟机就会把这个软引用加入到与之关联的引用队列中。

```
ReferenceQueue refQueue = new ReferenceQueue();
SoftReference softRef = new SoftReference(obj, refQueue);
```

软引用垃圾回收

```
if(JVM内存不足) {
      // 将软引用中的对象引用置为null
      str = null;
      // 通知垃圾回收器进行回收
      System.gc();
  }
```

代表软引用的类：java.lang.ref.SoftReference 代表弱引用的类：java.lang.ref.WeakReference 代表虚引用的类：java.lang.ref.PhantomReference 他们同时继承了：java.lang.ref.Reference

```
 public static void soft() throws Exception {
        Object obj = new Object();
        SoftReference<Object> softRef = new SoftReference<Object>(obj);
        System.out.println(softRef.get()); //有时候会返回Null
        // java.lang.Object@f9f9d8 System.out.println(refQueue.poll());
        // null
        // 清除强引用,触发GC
        obj = null;
        //System.gc();
        System.out.println(softRef.get());
        //Thread.sleep(200);
        //System.out.println(refQueue.poll());
    }
```

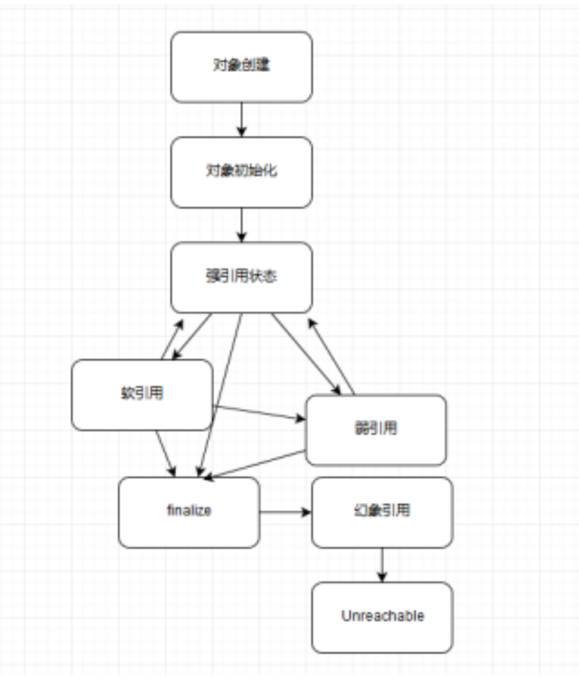
**弱引用**

弱引用和软引用的区别在于：弱引用拥有更短暂的生命周期，不管内存够不够，都会回收，都会回收它的内存。

```
Object obj = new Object();
WeakReference<Object> weakReference = new WeakReference<Object>(obj);
obj = null;
System.gc();
System.out.println(weakReference.get());// 有时候会返回Null
System.out.println(weakReference.isEnqueued());//判断是否有垃圾回收标记，表示即将回收的
```

**虚引用**

```
 // 虚引用
Object obj = new Object();
ReferenceQueue refQueue = new ReferenceQueue();
PhantomReference<Object> phantomReference = new PhantomReference<Object>(obj, ref
System.out.println(phantomReference.get());  // 永远返回Null
System.out.println(phantomReference.isEnqueued()); //返回时否从队列中删除
```



;

对象可达性分析

- 强可达（Strongly Reachable），就是当一个对象可以有一个或多个线程可以不通过各种引用访问到的情况。比如，我们新创建一个对象，那么创建它的线程对它就是强可达。
- 软可达（Softly Reachable），就是当我们只能通过软引用才能访问到对象的状态。
- 弱可达（Weakly Reachable），类似前面提到的，就是无法通过强引用或者软引用访问，只能通过弱引用访问时的状态。这是十分临近fnalize状态的时机，当弱引用被清除的时候，就符合fnalize的条件了。
- 幻象可达（Phantom Reachable），上面流程图已经很直观了，就是没有强、软、弱引用关联，并且fnalize过了，只有幻象引用指向这个对象的时候。当然，还有一个最后的状态，就是不可达（unreachable），意味着对象可以被清除了。

Java中4种引用的级别和强度由高到低依次为：强引用 –> 软引用 –> 弱引用 –> 虚引用

文章分享自微信公众号：

程序员财富自由之路

本文参与 ，欢迎热爱写作的你一起参与！

作者：猿星人
原始发表时间：2020-05-19

缓存 JVM Java

举报

点赞 1 分享

登录 后参与评论

0 条评论

## 相关文章

**强引用、软引用、弱引用、幻象引用有什么区别?**

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，理解引用对于掌握Java对象生命周期和JVM内部相关机制非常有帮助。

灬沙师弟

**强引用、软引用、弱引用、幻象引用有什么区别?**

特点：我们平常典型编码Object obj = new Object()中的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。 当JVM内存空...

葆宁

**强引用、软引用、弱引用、虚引用有什么区别?**

自 JDK1.2 开始，Java 提供了 4 种引用关系，以表示引用和实例对象的关系。

真正的飞鱼

**好未来面试官：说说强引用、软引用、弱引用、幻象引用有什么区别?**

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，理解引用对于掌握Java对象生命周期和JVM内部相关机制非常有帮助。

Java程序猿

**【必知必会】深入解析强引用、软引用、弱引用、幻象引用**

关于强引用、软引用、弱引用、幻象引用的区别，在BAT这样大公司的面试题中也经常出现，可能有些小伙伴觉得这个知识点比较冷门，但其实大家在开发中经常用到，如new一...

猿人谷

**Java引用类型：强引用，软引用，弱引用，虚引用**

Java中的引用，有点像C++的指针，通过引用，可以对堆中的对象进行操作。 在我们的代码生涯中，大部分使用的都是强引用，所谓强引入，都是形如Object o =...

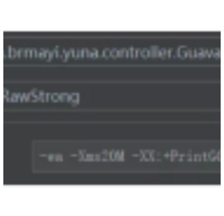
每天学Java

**软引用和弱引用的区别_强引用软引用弱引用虚引用的区别**

第一次GC的时候，软引用没有被回收，是因为这时候内存资源充足。第二次由于分配了较大的内存，导致GC，这时候由于内存资源紧张，软引用被回收了，也就是虽然User对...

全栈程序员站长

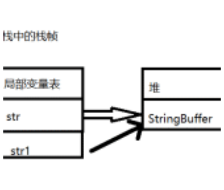
**Java的强引用、软引用、弱引用、虚引用**



工程中用到guava的本地缓存。它底层实现和API接口上使用了强引用、软引用、弱引用。所以温故知新下，也夯实下基础。

静儿

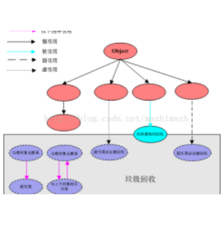
**什么是强引用、软引用、弱引用、虚引用?**



强引用（StrongReference）：一般指的是对像被new出来，强引用一般不会被jvm收回，但会报OutOfMemory（内存不足）。

逍遥壮士

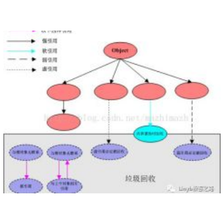
**Java 的强引用、弱引用、软引用、虚引用**



强引用是使用最普遍的引用。如果一个对象具有强引用，那垃圾回收器绝不会回收它。如下：

爱撸猫的杰

**Java 的强引用、弱引用、软引用、虚引用**



而这样 obj对象对后面new Object的一个强引用，只有当obj这个引用被释放之后，对象才会被释放掉，这也是我们经常所用到的编码形式。

lyb-geek

**强引用、软引用、弱引用、虚引用的对比**

从Jdk1.2开始，在java.lang.ref包下就提供了三个类：SoftReference（软引用），PhantomReference（虚引用）和WeakR...

黑洞代码

**【小家java】引用类型（强引用、软引用、弱引用、虚引用）**

本文不论述java中值传递和引用传递之间的问题（有需求的可移步理解java中值传递和引用传递），而重点讨论Java中提供了4个级别的引用：强应用、软引用、弱引用...

YourBatman

**Java中弱引用、软引用、虚引用、强引用、 Finalizer引用**

在Java层面，一共有四种引用：强引用、软引用、弱引用、虚引用，这几种引用的生命周期由强到弱。转换关系大致如下图所示：

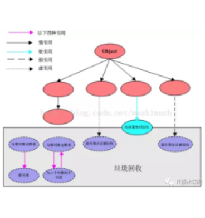
良辰美景TT

**【JVM从小白学成大佬】3.深入解析强引用、软引用、弱引用、幻象引用**

关于强引用、软引用、弱引用、幻象引用的区别，在很多公司的面试题中经常出现，可能有些小伙伴觉得这个知识点比较冷门，但其实大家在开发中经常用到，如new一个对象的时...

猿人谷

**【JVM】如何理解强引用、软引用、弱引用、虚引用?**



如果这个对象是偶尔的使用，并且希望在使用时随时就能获取到，但又不想影响此对象的垃圾收集，那么你应该用 Weak Reference 来记住此对象。

用户7353950

**java强引用、软引用、弱引用、虚引用以及FinalReference**

基于JDK1.8 rt.jar是java中的基础类库，在它的 java.lang.ref包下，有着一堆关于引用的类。软引用、弱引用以及虚引用的定义就在其中。另外...

早安嵩骏

**Java有关强引用，软引用，弱引用，虚引用的记录**

引用本身很好理解，引用类型的对象中存放的是实际对象的内存地址，垃圾回收时，就看对象是否存在引用。Java不需要开发人员分配内存和释放内存，但是可以通过四种引用类...

码农小麦

**理解Java中的强引用，软引用，弱引用，虚引用**

在JDK1.2以前的版本中，当一个对象不被任何变量引用，那么程序就无法再使用这个对象。也就是说，只有对象处于可触及状态，程序才能使用它。这就像在商店购买了某样物...

IT大咖说

更多文章 >