

单例对象会被JVM回收吗，答：不会

原创 当我谈编程时我谈些什么 于 2021-08-17 19:08:28 发布 阅读量2.1k 收藏 3 点赞数

版权

分类专栏：[java永无止境](#)



java永无止境 专栏收录该内容

3 订阅

113 篇文章

订阅专栏

一、前言

理清概念之后，就可以多思考一些相关的问题，触类旁通。

这里有一个比较常问到的问题，**单例模式** 创建的对象是否会被JVM回收？为什么？

二、分析

• 何为单例？

单例模式涉及到一个单一的类，该类负责创建自己的对象，同时确保只有单个对象被创建。这个类提供了一种访问其唯一的对象的方式，可以直接访问，不需要实例化该类的对象。

注意：

- 1、单例类只能有一个实例。
- 2、单例类必须自己创建自己的唯一实例。
- 3、单例类必须给所有其他对象提供这一实例。

例如，以最简单的饿法式实现单例模式

```
public class Singleton {
    private static Singleton instance = new Singleton();
    private Singleton (){}
    public static Singleton getInstance() {
        return instance;
    }
}
```

• 何以判断是否可回收？

以主流的JVM HotSpot来说，使用的是可达性分析，通过从GC Roots的引用作为起点，来判断对象可达性，从而决定是否回收该对象。

那什么可以作为GC Roots呢？

- 虚拟机栈（栈帧中的本地变量表）中的引用的对象。
- **方法区中的类静态属性引用的对象。**
- 方法区中的常量引用的对象。
- 本地方法栈中JNI的引用的对象。

从单例模式创建的对象来看，可以判断其符合**方法区中的类静态属性引用的对象**这条定义。

方法区中的类中的静态属性必然引用堆中的对象。

那如此看，如果没办法解决方法区中的类的话，可以认为单例对象不会被回收。

类会放在方法区，对象会放在堆中。

• 何以判断方法区中的类？

再想深一层，什么情况下有可能触发方法区（JDK8-）或元数据区（JDK8+）中单例类消亡？这样的话，堆中的对象就不可达，从而会被回收。如果这个假想不现实，则可认为单例模式实现的对象不可回收。

符合下列三点则可认为类卸载：

- 该类所有的实例都已经被回收，也就是Java堆中不存在该类的任何实例。
- 加载该类的Class Loader已经被回收。
- 该类对应的java.lang.Class对象没有任何地方被引用，无法在任何地方通过反射访问该类的方法。

要求相当苛刻，所以一般情况下可以认定不可回收了。