



链滴

单例设计模式

作者: [liaoshengzhe](#)

原文链接: <https://ld246.com/article/1508124736044>

来源网站: [链滴](#)

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

单例设计模式是23种设计模式中最简单的一种模式。它要有以下几个要素：

- 一、私有的构造方法
- 二、指向自己实例的私有静态引用
- 三、以自己实例为返回值的静态公有方法

单例模式根据实例化对象时机的不同分为两种：一种是饿汉式单例，一种是懒汉式单例。前者是在单类被加载的时候就实例化一个对象交给自己的引用；而后者是在调用取得实例方法的时候才会实例化对。代码如下：

饿汉式单例

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

懒汉式单例

```
public class Singleton {
    private static Singleton singleton;
    private Singleton(){}

    public static synchronized Singleton getInstance(){
        if(singleton==null){
            singleton = new Singleton();
        }
        return singleton;
    }
}
```

单例模式的优点：

- 在内存中只有一个对象，节省内存空间。
- 避免频繁的创建销毁对象，可以提高性能。
- 避免对共享资源的多重占用。
- 可以全局访问。

****适用场景：****由于单例模式的以上优点，所以是编程中用的比较多的一种设计模式。我总结了一下所知道的适合使用单例模式的场景：

- 需要频繁实例化然后销毁的对象。
- 创建对象耗时过多或者耗资源过多，但又经常用到的对象。
- 有状态的工具类对象。

- 频繁访问数据库或文件的对象。
- 以及其他我没用过的所有要求只有一个对象的场景。

单例模式注意事项：

- 只能使用单例类提供的方法得到单例对象，不要使用反射，否则将会实例化一个新对象。
- 不要做断开单例类对象与类中静态引用的危险操作。
- 多线程使用单例使用共享资源时，注意线程安全问题。

关于java中单例模式的一些争议：

单例模式的对象长时间不用会被jvm垃圾收集器收集吗

看到不少资料中说：如果一个单例对象在内存中长久不用，会被jvm认为是一个垃圾，在执行垃圾收集的时候会被清理掉。对此这个说法，笔者持怀疑态度，笔者本人的观点是：**在hotspot虚拟机1.版本中，除非人为地断开单例中静态引用到单例对象的联接，否则jvm垃圾收集器是不会回收单例对象的。**

在一个jvm中会出现多个单例吗

在分布式系统、多个类加载器、以及序列化的情况下，会产生多个单例，这一点是无庸置疑的。那么在同一个jvm中，会不会产生单例呢？使用单例提供的getInstance()方法只能得到同一个单例，非是使用反射方式，将会得到新的单例。代码如下

```
Class c = Class.forName(Singleton.class.getName());
Constructor ct = c.getDeclaredConstructor();
ct.setAccessible(true);
Singleton singleton = (Singleton)ct.newInstance();
```

懒汉式单例线程安全吗

主要是网上的一些说法，懒汉式的单例模式是线程不安全的，即使是在实例化对象的方法上加**ynchronized**关键字，也依然是危险的，但是笔者经过编码测试，发现加**synchronized**关键字修饰后虽然对性能有部分影响，但是却是线程安全的，并不会产生实例化多个对象的情况。

单例模式只有饿汉式和懒汉式两种吗

饿汉式单例和懒汉式单例只是两种比较主流和常用的单例模式方法，从理论上讲，任何可以实现一个类只有一个实例的设计模式，都可以称为单例模式。

单例类可以被继承吗

饿汉式单例和懒汉式单例由于构造方法是private的，所以他们都是不可继承的，但是其他很多例模式是可以继承的，例如登记式单例。

饿汉式单例好还是懒汉式单例好

在java中，饿汉式单例要优于懒汉式单例。