目录

[1. 单例模式下懒汉和饿汉的优缺点和区别 2](#_Toc68978282)

[2. 2](#_Toc68978283)

[3. 2](#_Toc68978284)

[4. 2](#_Toc68978285)

[5. 2](#_Toc68978286)

[6. 2](#_Toc68978287)

[7. 2](#_Toc68978288)

[8. 2](#_Toc68978289)

1. 单例模式下懒汉和饿汉的优缺点和区别

一个在类加载时就初始化，一个在第一次被使用时构建初始化

1. 在时间和空间上

饿汉模式资源效率不高，可能getInstance（）永远不会执行到，但执行该类的其他静态方法或者加载了该类，那么这个实例仍然初始化。

懒汉模式资源利用率高，不执行getInstance（）就不会被实例，可以执行该类的其他静态方法。缺点：第一次加载时不够快，多线程使用不必要的同步开销大。

1. 线程安全上

饿汉模式线程安全，在类加载的同时已经创建好一个静态对象，调用时反应速度快

懒汉模式线程不安全，加双重检测，C++11规定后不需要加。

特点和选择：

懒汉：在访问量较小时，采用懒汉实现。这是以时间换空间。

　　 饿汉：由于要进行线程同步，所以在访问量比较大，或者可能访问的线程比较多时，采用饿汉实现，可以实现更好的性能。这是以空间换时间。

懒汉：故名思义，不到万不得已就不会去实例化类，也就是说在第一次用到类实例的时候才会去实例化。与之对应的是饿汉式单例。（注意，懒汉本身是线程不安全的，如上例子）

饿汉式会提前浪费我们的内存空间以及资源，如果有项目中要求我们在使用到实例的时候再去实例化，则还是需要使用懒汉式。