单例模式：

一、实现步骤：

1. 私有化构造方法
2. 静态私有的本类实例
3. 访问该实例的静态方法

二、实现方式

1) 懒汉式（线程安全）（懒加载、延迟加载）

public class Singleton {  
 private static Singleton *instance*;  
 private Singleton(){}  
  
 public static Singleton getInstance(){  
 if(*instance*==null){  
 synchronized (Singleton.class){  
 if(*instance*==null){  
 *instance*=new Singleton();  
 }  
 }  
 }  
  
 return *instance*;  
 }  
}

2）饿汉式

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

三、其他思考

1）单例模式一般用于工具类中，并且该工具类还频繁被使用

2）单例模式vs私有化构造方法并且静态方法访问，即使用静态方法来做工具类中的方法

考虑因素：内存的占用

单例模式，对象存在堆内，每次方法调用使用实例来调用，方法调用时仅仅是入栈和出栈，然后销毁栈空间

私有构造方法并静态方法访问，类在加载时即存在于静态方法区中，一直到程序终止，占用了方法区的内存

通常，在工作项目中倾向于使用单例模式的工具类