

饿汉

```
public class Singleton {  
    // 1. 将构造方法私有化，使其不能在类的外部通过new关键字实例化该类对象。  
    private Singleton() {}  
  
    // 2. 在该类内部产生一个唯一的实例化对象，并且将其封装为private static类型的  
    成员变量。  
    private static final Singleton instance = new Singleton();  
  
    // 3. 定义一个静态方法返回这个唯一对象。  
    public static Singleton getInstance() {  
        return instance;  
    }  
}
```

懒汉

```
public class Singleton {  
    // 1. 将构造方法私有化，使其不能在类的外部通过new关键字实例化该类对象。  
    private Singleton() {}  
  
    // 2. 在该类内部产生一个唯一的实例化对象，并且将其封装为private static类型的  
    成员变量。  
    private static Singleton instance;  
  
    // 3. 定义一个静态方法返回这个唯一对象。  
    public static Singleton getInstance() {  
        if(instance==null){  
            instance=new Singleton();  
        }  
        return instance;  
    }  
}
```