

```
package com.singleton.type6;
```

```
public class SingletonTest06 {
```

```
    public static void main(String[] args) {  
        System.out.println("双重检查");  
        Singleton instance = Singleton.getInstance();  
        Singleton instance2 = Singleton.getInstance();  
        System.out.println(instance == instance2); // true  
        System.out.println("instance.hashCode=" + instance.hashCode());  
        System.out.println("instance2.hashCode=" + instance2.hashCode());  
    }  
}
```

```
class Singleton {  
    private static volatile Singleton instance;
```

```
    private Singleton() {}
```

//提供一个静态的公有方法，加入双重检查代码，解决线程安全问题，同时解决懒加载问题
//同时保证了效率，推荐使用

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