单例模式的特点

```java
public class SingleObject {
 //创建 SingleObject 的一个对象
 private static SingleObject instance = new SingleObject();
 //让构造函数为 private，这样该类就不会被实例化
 private SingleObject(){}
 //获取唯一可用的对象
 public static SingleObject getInstance(){
 return instance;
 }
}
```

懒汉模式(线程不安全)

多线程下没有synchronized锁(即多线程不能工作)

```java
public class Singleton {
 private static Singleton instance;
 private Singleton (){}

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

懒汉模式(线程安全)

多线程下有synchronized锁(即多线程能工作但是影响效率)

```java
public class Singleton {
 private static Singleton instance;
 private Singleton (){}

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

饿汉模式

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