1. 使用静态内置类实现单例模式

```java
public class MySingleton {
    //内部类
    private static class MySingletonHandler{
        private static MySingleton instance = new MySingleton();
    }
    private MySingleton(){}

    public static MySingleton getInstance() {
        return MySingletonHandler.instance;
     }
}
```

2. 使用双重同步锁

```java
public class Singleton {
    // volatile防止new Singleton()时重排序
    private volatile static Singleton instance;
    private Singleton (){}
    public static Singleton getInstance(){     //对获取实例的方法进行同步
        if (instance == null){
            synchronized(Singleton.class){
                if (instance == null)
                    instance = new Singleton();
            }
        }
        return instance;
    }
}
```

3. 饿汉式单例

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