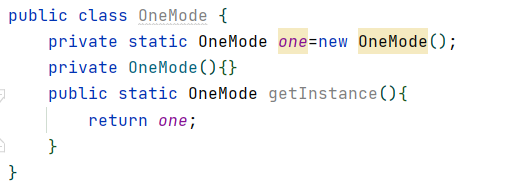
**单例设计模式——内存中只可以有一个实例对象**

**1，饿汉式**

**①类加载到JVM内存中，就实例化一个实例对象，保证线程安全**

**②构造方法私有化，保证只有一个实例对象**

**③提供一个public getInstance()方法用来获取实例对象**



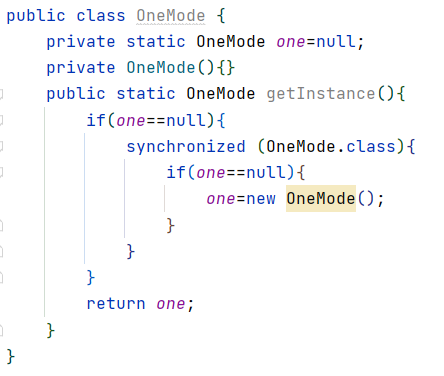
**如何保证线程安全？**

**JVM只会加载OneMode一次，在加载的初始化阶段会创建一个OneMode实例**

**二，懒汉式**

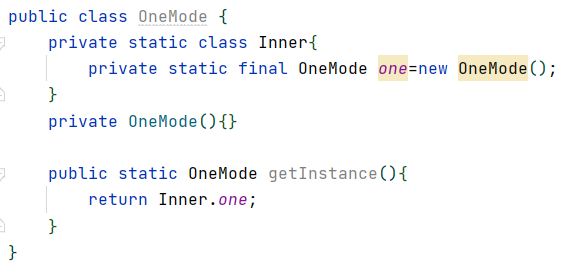
**什么使用需要使用这个类的实例对象了，什么时候才会进行初始化**

**①为了保证线程安全，使用sychronized**



**②通过静态内部类来完成懒汉式**

**前提：静态内部类与外部类之间彼此独立，加载外部类时，不会加载静态内部类**



只有我们使用OneMode.getInstance()时，才会加载静态内部类

如么保证线程安全？

JVM只会加载Inner一次，在类加载的初始化阶段，会创建一个OneMode实例

**3，通过枚举实现单例**

可以解决线程同步问题还可以解决序列化问题

