# 单例模式

程序生命周期内，保证对象只有一个实例，并提供全局访问点。属于创建型模式。主要作用是减少内存冗余，避免资源多重占用。

## 单例实现方式有七种：

1. 懒汉式
   1. 只有用到的时候才会进行创建，特性是延迟加载，优点是没使用前不占用内存空间，以及可以加快程序启动时间。缺点是要管理实例创建线程安全问题。
2. 饿汉式
   1. 用户体验比懒汉式更好，程序启动就直接加载，缺点是占用内存，优点是不用考虑线程创建安全问题。
3. 内部类
   1. 实现过程是利用静态内部类机制，属于懒汉式的一种
4. 枚举类
   1. 利用枚举的特性，防止被反射或者序列化破环。使用简单，而且安全，推荐实现单例的最好一种方式。
5. 容器
   1. 常见于Spring中，利用map和synchronize和反射进行创建单例
6. 序列化
   1. 实现serializable接口，重写readResolve接口，防止被序列化破坏。
7. ThreadLocal
   1. 线程副本单例，天生线性安全，也是一种map实现机制，确保一个线程内是单例，应用场景可以是在JDBC动态切换数据源。

单例分类：

懒汉式、饿汉式、注册式、ThreadLocal。

常用的破环单例的方式：

1. 多线程竞争创建单例
2. 反射，获取构造，强吻
3. 序列化

内部类执行逻辑时序图如下：

