单例模式。

1. 常见的单例写法有：饿汉式、懒汉式、DCL式、内部类形式、枚举形式

2

饿汉式：

优点：在声明静态对象时已经初始化，调用获取实例的方法执行快

缺点：消耗不必要的资源

懒汉式：

优点：在第一次调用获取实例的方法时才会进行初始化，节省资源，通过添加关键字synchronized 可以在多线程情况下保证单例对象唯一

缺点：第一次加载时需要及时进行初始化，反应稍慢，每次调用获取实例的方法时都进行同步，造成不必要的开销

DCL式：

优点：既能在需要时才初始化单例，又能保证线程安全，且单例对象初始化后不会再进行同步锁

缺点：由于第一次反应稍慢，也由于java内存模型的原因偶尔会失败

内部类：

优点：能保证线程安全，单例对象的唯一性，也延迟了单例的实例化

缺点：第一次反应稍慢

枚举形式：

优点：默认枚举实例的创建时线程安全的，并且在任何情况下都是一个单例

3

破坏单例的方式主要有反射和序列化

1. 反射的预防方式

在类中添加一个静态布尔类型变量，记录构造构造函数是否被调用

在构造函数中，判断上述的变量，如果是第二次调用则抛出异常

1. 序列化的预防

通过重写反序列化中的钩子函数 Object readResolve() ， 让函数直接返回静态实例即可

4

，当第一次加载Singleton类时并不会初始化实例对象，只有在第一次调用获取内部类的静态实例变量时才会导致实例对象被初始化，因此第一次调用获取内部类静态变量的方法时，会导致虚拟机加载内部类