20190621学习心得

1. 单例模式

被单例的对象只能有一个实例存在。单例模式的实现方式是，一个类能返回对象的一个引用（永远是同一个）和一个获得该唯一实例的方法（必须是静态方法）。通过单例模式，我们可以保证系统中只有一个实例，从而在某些特定的场合下达到节约或者控制系统资源的目的

1. 饿汉模式

“饿汉模式” 就是很 “饥渴”，所以一上来就需要给它新建一个实例。但这种方法有一个明显的缺点，那就是不管有没有调用过获得实例的方法，每次都会新建一个实例

1. 懒汉模式

“懒汉模式” 就是它很懒，一开始不新建实例，只有当它需要使用的时候，会先判断实例是否为空，如果为空才会新建一个实例来使用

1. 线程安全的懒汉模式

在基本的懒汉模式上，把它设为线程同步就好了。synchronized 的作用就是保证在同一时刻最多只有一个线程运行，这样就避免了多线程带来的问题

1. 双重检验锁

在前面的基础上，又有了 “双重检验锁” 的方法，给实例加一个 volatile 关键字，它的作用就是防止编译器自行优化代码

1. 静态内部类

利用了 JVM 自身的机制来保证线程安全，因为 WifeHolder 类是私有的，除了 getWife() 之外没有其它方式可以访问实例对象，而且只有在调用 getWife() 时才会去真正创建实例对象

1. 枚举

通过 Wife.INSTANCE 来访问实例对象，这比 getWife() 要简单得多，而且创建枚举默认就是线程安全的，还可以防止反序列化带来的问题

1. 单例模式应用

在资源共享的情况下，避免由于多个资源操作导致的性能或损耗等就可以使用单例模式