1. 熟练掌握单例模式的常见写法。

饿汉式单例模式，懒汉式单例，同步，内部内单例，双重校验，容器式单例，枚举单例

1. 总结每种单例写法的优、缺点。

2.1 饿汉式单例

饿汉式单例是在类加载的时候就立即初始化，并且创建单例对象。绝对线程安全，在线

程还没出现以前就是实例化了，不可能存在访问安全问题。

优点: 没有加任何的锁、执行效率比较高，在用户体验上来说，比懒汉式更好。

缺点：类加载的时候就初始化，不管用与不用都占着空间，浪费了内存，有可能占着茅坑不拉屎。

2.2 懒汉式单例

懒汉式单例是被外部类调用的时候内部类才会加载

优点：与饿汉式单例对比，它只是被调用时才会占用内存空间

缺点：多个线同时调用getInstance方法时，线程是不安全的，需要人为去加sychronized关键字和双重锁机制等去解决线程问题，底层JVM给对象分配内存地址和初始化对象等等操作时会出现JVM执行顺序不一样问题。

2.3 内部内单例

内部内单例是结合了懒汉式单例的优点和饿汉式优点（不用加sychronized关键字）

优点： 调用方法时加载内部类去创建实例，创建的实例是静态的用了final修饰，不存在线程安全问题;

缺点：通过反射构造方法去创建一个新的实例，破坏了单例原则，可以在私有的构方法加决断，阻止通过反射来创建一个新的单例;

2.4 注册式单例

注册式单例就是将每一实例都登记到某个地方，使用唯一的标识获取实例。

它分为2种写法：一种为容器缓存，一种为枚举登记

优点：枚举登记在java中是安全单例模式，它的层底很好的规避了通过反射、序列化接口来创建新的实例;

缺点：底层是通过饿汉式创建的实例，可以说不算是缺点;

2.5本地线程单例

优点：在同一个线程内创建的实例是唯一的单例对象

缺点：多个线程创建的实例的对象不是唯一的，破坏单例原规

1. 思考破坏单例模式的方式有哪些？并且归纳总结。

1反射，2序列化，3多线程访问

饿汉式无法通过以上三种破坏

懒汉式以上三种都能破坏

内部内单例 1 反射可以破坏 2 3不能破坏

注册式单例 枚举 以上三种均不能破坏

1. 梳理内部类的执行逻辑，并画出时序图。

ClassLoader加载静态的类只会的执行一次，通过调用方法getInstance() 才会加载静态内部的实例