1、熟练掌握单例模式的常见写法。

a、饿汉式单列

b、懒汉式单例

c、双重检查锁单例

d、注册式单例（枚举、ThreadLocal单例）

2、总结每种单例写法的优、缺点。

a、饿汉式：饿汉式单例是在类加载的时候就立即初始化，并创建单例对象

优点：没有加锁，执行效率比较高。

缺点：加载时就创建实列，占用内存。

b、懒汉式：懒汉式单例是被外部类调用的时候内部类才会加载

优点：只有被调用时，在占用内存空间。

缺点：线程不安全，需要为对象加上sychronized关键字保证线程安全。

c、双重检查锁单例

优点：保证线程安全。

d、注册是单例：

* 枚举式单例

优点：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

* ThreadLocal单例：

优点：保证线程内部的全局唯一，且天生线程安全，线程内部，伪线程安全，内部安全

缺点：多个线程就不安全了。

3、思考破坏单例模式的方式有哪些？并且归纳总结。

反射、深克隆

4、梳理内部类的执行逻辑，并画出时序图。

通过反射拿到私有的构造方法——>构造（LazyInnerClassSingleton）——>getInstance()——>返回一个实例对象（InnerClass内部类）