1. 单例设计模式

（1）什么时单例设计模式

单例设计模式就是无论程序如何运行，采用单例设计模式的类始终只有一个实例对象。

（2）单例设计模式的优缺点

优点：防止其它对象对自己实例化，确保了所有的对象访问的都是同一个实例，而且一直只有一个实例，节约了系统内存，对于需要频繁创建和销毁的对象，也能提高系统的性能。

缺点：显而易见，因为始终只有一个相同的实例，那么肯定不适用于变化的对象，不适用于再不同的场景会发生变化的对象。然后单例模式没有抽象层，所以不能扩展。然后如果实例化的对象长时间不被使用个，可能会被回收掉，这样就会导致对象状态的丢失。

（3）单例模式的类创建步骤。

a. 先把构造函数设计为private的。

b. 创建一个private static的对象。

c. 提供一个public static方法来共外界访问。

（4）懒汉式和饿汉式的区别

懒汉式是利用时间换空间，如果不需要的时候，就不创建实例，节省空间，但是每次获取都要判断。

饿汉式是利用空间换时间，当类装载的时候就会创建实例，不管你用不用，都会创建出来。

饿汉式是线程安全的。

但是懒汉式不是，所以再使用懒汉式的时候，要在对外提供的接口的创建操作执行之间要对当前类加锁，以保证线程安全性。

1. 工厂设计模式

（1）什么是工厂设计模式

程序在接口和子类之间加入了一个过渡端，通过此过渡端可以动态取得实现了共同接口的子类实例化对象。就是同一接口的子类的实例化都可以通过工厂类来完成。

简单工厂模式：工厂类含有必要的判断逻辑，可以决定在什么时候创建哪一个产品类的实例，客户端可以免除直接创建产品对象的责任，而仅仅"消费"产品。简单工厂模式通过这种做法实现了对责任的分割。外界与具体类隔离开来，偶合性低。明确区分了各自的职责和权力，有利于整个软件体系结构的优化。

工厂方法模式是为了克服简单工厂模式的缺点（主要是为了满足OCP）而设计出来的。简单工厂模式的工厂类随着产品类的增加需要增加很多方法（或代码），而工厂方法模式每个具体工厂类只完成单一任务，代码简洁。工厂方法模式完全满足OCP，即它有非常良好的扩展性。

抽象工厂模式就是工厂中不再只创建一种对象，而是创建多种对象，也就是说一个产品有不同的系列，而每个系列又有不同的属性，每一种操属性对应的就是一种新的创建方式。

简单工厂就是说一个工厂创建所有的对象，而工厂模式就是每个对象对应一个工厂类，有多个工厂类，而抽象工厂方法就是每个对象的不同的操作手段或者说属性，然后一个对象的工厂类里，有多个创建方法，调用不同的创建方法来创建不同的对象。

(2)工厂设计模式的优缺点。

有点： 开放-封闭原则（ocp）： 扩展是开放的，修改是封闭的。高内聚，低耦合。

无论是简单工厂模式、工厂模式还是抽象工厂模式，它们本质上都是将不变的部分提取出来，将可变的部分留作接口，以达到最大程度上的复用。

缺点：每增加一个产品都要增加一个工厂子类，这样就增加了额外的开发量。

1. 代理模式
2. 什么是代理模式

指由一个代理主题来操作真实主题，真实主题执行具体的业务，而代理主题执行其它业务，比如说生活中，客户通过网络代理连接网络（具体业务），而网络代理则是完成用户的验证等操作。

1. 代理类的应用

a.远程代理：就是为一个对象在不同的地址空间提供局部代表，这样可以隐藏一个对象存在于不同地址空间的事实。

b.虚拟代理：根据需要创建开销很大的对象，通过它来存放实例化需要很长时间的真实对象。这样子就可以达到性能的最优化，比如你打开一个很大的html页面，里面有很多是文字和图片，但是我们能很快的打开。原因是我们看到的所有的文字优先下载，然后在将图片一张一张地下载。那些未下载的图片先用图片库来代替，此时的虚拟代理就是通过图片库来代理图片。

c.安全代理：用来控制真实对象访问时的权限。一般用于对象应有不同的访问权限的时候。

d.智能代理：当调用真实对象时，代理处理另外一些事。