**Q：高内聚、低耦合？**

A： 高内聚是指一个软件模块是由相关性很强的代码组成，只负责一项任务，也就是常说的单一责任原则；

对于低耦合，粗浅的理解是：一个完整的系统，模块与模块之间，尽可能的使其独立存在。也就是说，让每个模块，尽可能的独立完成某个特定的子功能。

**Q：几种常用的设计模式**

A： 单例模式

就是一个应用程序中，某个类的实例对象只有一个。设置构造函数私有，一个public的静态函数返回单例。

观察者模式（业务界面是观察者、业务数据是被观察者）

观察者模式中有一个称作“主题”的对象和若干个称作“观察者”的对象，“主题”和“观察者”间是一种一对多的依赖关系，当“主题”的状态发生变化时，所有“观察者”都得到通知。

抽象“主题”类包含了添加和删除观察者及通知观察者的接口。

具体“主题”类继承抽象“主题”类，需使用一个集合，比如ArrayList，存放观察者。

抽象“观察者”类是一个接口，该接口规定了具体观察者用来更新数据的方法。

具体“观察者”类继承于抽象“观察者”类，具体观察者包含有可以存放具体主题引用的主题接口变量

装饰模式（椅子、带轮子的椅子、带轮子和靠背的椅子）

对已有的业务逻辑进一步的封装，使其增加额外的功能；

适配器模式

将两种完全不同的事物联系到一起，就像现实生活中的变压器。假设一个手机充电器需要的电压是20V，但是正常的电压是220V，这时候就需要一个变压器，将220V的电压转换成20V的电压，这样，变压器就将20V的电压和手机联系起来了。

建造者模式（汽车；装配车间；装配线A、装配线B；工厂）

将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

Builder：给出一个抽象接口或抽象类，以规范产品的建造。这个接口规定要实现复杂对象的哪些部分的创建，并不涉及具体的对象部件的创建，一般由子类具体实现。

ConcreteBuilder：Builder接口的实现类，定义各个部件，并定义各部件的组装过程。

Director：调用具体建造者来创建复杂对象的各个部分，在指导者中不涉及具体产品的信息，只负责保证对象各部分完整创建或按某种顺序创建。

Product：要创建的复杂对象，产品类。

工厂模式（手机工厂；手机；iphone、华为、小米；客户）

工厂模式和建造者模式类似，但是工厂模式关注的是产品实体。

代理模式（广告代言、明星、经纪人、广告商）

为其他对象提供一种代理，以控制对该对象的访问。访问对象前，先访问代理，代理可以对访问进行预处理与后处理。

外观模式（做饭、扫地；保姆；主人）

外观模式定义了一个高层接口，这个接口使得这一子系统更加容易使用

**Q：工厂模式和建造者模式比较**

A： 工厂模式关注整体、建造者模式关注过程。

以创建手机为例，工厂模式关注的是创建手机实体、建造者模式关注的是装配手机的过程。

**Q：装饰模式、代理模式比较**

A： 装饰模式是一层层继承包装；代理模式中代理和被代理者继承于同一基类，代理包含对被代理者的指针或引用。

**Q：代理模式和外观模式**

A： 代理模式中代理和被代理者继承于同一基类，具有相同的接口；装饰模式中，由高层接口类完成复杂的底层过程，用户类只需要调用高层接口类就行。

**Q：工厂模式、工厂方法模式、抽象工厂模式**

A： 简单工厂模式：在工厂类中决定创建具体的产品；实现简单，增加或删除产品种类时，需要改动工厂类；

工厂方法模式：增加一层间接性，屏蔽了产品类；增加删除产品种类时，不需要改动抽象工厂类。

抽象工厂模式：工厂方法模式是针对单个产品结构的，抽象工厂针对多个产品等级结构，增加了一个产品族管理的类。

<https://blog.csdn.net/wfg18801733667/article/details/60954744>

<https://blog.csdn.net/seu_calvin/article/details/52268027>