**工厂模式：**

简单工厂：定义一个创建对象的接口，由工厂类根据接受的信息创建不同的对象，让其子类实现创建细节。

工厂方法模式：工厂类中不同方法对应不同的产品，创建不同的对象。

抽象工厂模式： 定义一个抽象工厂，抽离出相同的抽象方法，让其子类根据不同信息创建不同对象。

作用：1)缩短开发时间；2)降低维护成本；3)屏蔽产品的具体实现，调用者只关心产品的接口。

**代理模式：**代理类为委托类提供代理，用来控制对目标类的访问。

静态代理：在程序运行前就已经存在代理类的字节码文件，代理类和委托类的关系在运行前就确定了。（代理类、目标类需要实现相同接口）

动态代理：是在实现阶段不用关心代理类，而在运行阶段才指定哪一个类被代理。（动态代理处理器类实现InvocationHandler接口，实现invoke方法）

作用：1)保护了目标类；2）高扩展性

**MVC模式：**把M（业务模型）和V（用户界面）实现代码分离，从而使同一个程序可以使用不同的表现形式。比如一批统计数据可以分别用[柱状图](https://baike.baidu.com/item/%E6%9F%B1%E7%8A%B6%E5%9B%BE)、[饼图](https://baike.baidu.com/item/%E9%A5%BC%E5%9B%BE)来表示。C（控制器）存在的目的则是确保M和V的同步，一旦M改变，V应该同步更新。

**单例模式：**保证一个类仅有一个实例，自行实例化，并向整个系统提供一个访问它的全局访问点。（一般用恶汉模式）

**原型模式：**用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。（浅克隆：被克隆的类实现 Cloneable接口，重写clone方法。深克隆：实现Serializable接口）