**建造者模式：**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。包含Builder（抽象建造者）、ConcreteBuilder（具体建造者）、Director（指挥者）、Product（产品角色）这四个角色。

与工厂模式相比，建造者模式的结构中还引入了一个指挥者类Director，该类的作用主要有两个：

一、它隔离客户与生产过程；

二、它负责控制产品的生成过程。指挥者针对抽象建造者编程，客户端只需要知道具体建造者的类型，无须关心产品对象的具体组装过程，即可通过指挥者类调用建造者的相关方法，返回一个完整的产品对象。

**建造者模式与抽象工厂模式的比较:**

与抽象工厂模式相比， 建造者模式返回一个组装好的完整产品 ，而抽象工厂模式返回一系列相关的产品，这些产品位于不同的产品等级结构，构成了一个产品族。

在抽象工厂模式中，客户端实例化工厂类，然后调用工厂方法获取所需产品对象，而在建造者模式中，客户端可以不直接调用建造者的相关方法，而是通过指挥者类来指导如何生成对象，包括对象的组装过程和建造步骤，它侧重于一步步构造一个复杂对象，返回一个完整的对象。

**结合建造者模式与工厂模式对KFC案例进行分析：**

如果将抽象工厂模式看成KFC产品生产工厂 ，生产一个产品族的产品（包含饮料和主食、甜品等），那么建造者模式就是一个KFC产品（包含饮料和主食、甜品等）组装工厂 ，通过对产品（包含饮料和主食、甜品等）的组装可以返回客户所需要的套餐。在工场模式中，新增加一个产品就要实例化一个相对应的工厂，它们是成对出现的；而在建造者模式中，只有一个指挥者，而有多种产品。

综上所述，在设计者模式中，如果多了一类产品，且它们的构造过程与之前的产品不同，则可以相对应的多设计一个指挥者，这便结合了工厂模式。在KFC套餐实例中，可以实现建造者模式与工厂模式的组合（工厂模式负责生产产品，而建造者模式负责组装）。