# 建造者模式

## 问题引入

软件开发过程中有时需要创建一个复杂的对象，这个复杂对象通常由多个子对象按一定的步骤组合而成。生活中这样的例子也很多，比如王者荣耀中的英雄由不同的性别、语音、技能、皮肤等特性组成，游戏中的各种英雄有上百个，但都是由这几个特性拼凑出来的。建造者模式（Builder Pattern）适用多个简单的对象逐步构建出一个复杂的对象。

## 模式定义

将一个复杂对象的构造与其表示分离，使同样的构建过程可以创建不同的表示，这样的设计模式就是建造者模式。

建造者模式将一个复杂对象分解为多个简单的对象，逐步完成复杂对对象的构建。建造者模式将“变“与”不变“相分离，即产品的组成部分不变，但每个部分的具体实现可以灵活选择。就好比肯德基的“汉堡+薯条+饮料”套餐，汉堡可以是香辣鸡腿堡、鳕鱼堡、牛肉堡……，薯条可以是小薯、中薯、大薯，饮料可以是可乐、雪碧、芬达、奶茶……。

## 意图

将一个复杂的构建与其表示相分离，使得同样的构建过程可以创建不同的表示。在软件系统中，有时候面临着"一个复杂对象"的创建工作，其通常由各个部分的子对象用一定的算法构成；由于需求的变化，这个复杂对象的各个部分经常面临着剧烈的变化，但是将它们组合在一起的算法却相对稳定。

## 何时使用

一些基本部件不会变，而其组合经常变化。比如肯德基的汉堡、薯条、饮料基本不会变，但套餐会经常改变。

## 模式的结构

创建者模式由产品、抽象建造者、具体建造者、指挥者等4个要素组成，各要素的功能如下：

1. 产品：描述复杂的对象由哪些成分组成，各组成成分由建造者负责创建。
2. 抽象建造者：包含创建各组成成分的方法，通常还提供一个返回复杂对象的方法。
3. 具体建造者：实现抽象建造者中创建各组成成分的方法，提供各成分具体创建的细节。
4. 指挥者：调用创建者中成分创建和装配的方法，完成复杂对象的创建。

## 模式特点

建造者模式具有以下优点：

1. 建造者独立，易拓展。
2. 便于控制细节风险。

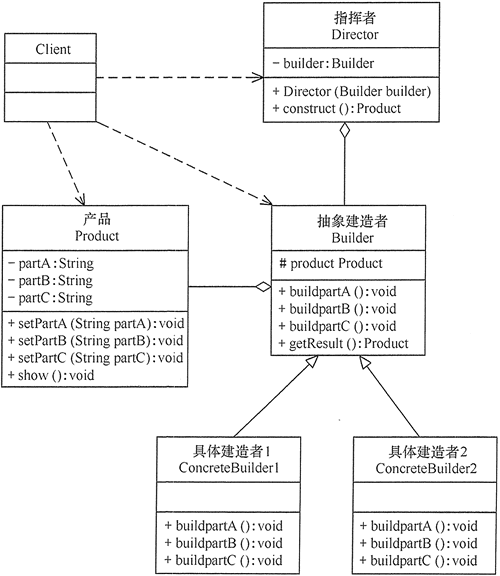
建造者模式具有以下缺点：

1. 产品必须具有共同点，范围有限制。
2. 如内部变化复杂，会由很多的建造类。

## 建造者模式与工厂方法模式

工厂方法模式更加关注零件的创建过程，建造者模式更加关注零件的组装过程。

## 实例

从实例的UML图中可以看出：

1. 产品的组成是不变的，具体如何装配产品由具体的建造者决定。
2. 各具体建造者相互独立，如果需要添加新的装配方案，不需要修改现有的建造者，只需要添加新的建造者，满足开闭原则。
3. 客户端不知道产品装配的细节。
4. 如果产品具有很多装配方案，将会创建很多的建造者。