链滴

设计模式学习笔记之建造者模式

作者：hjljy
原文链接：https://ld246.com/article/1562252452399
来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20190319.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h2 id="前言">前言</h2>
这是一篇学习笔记，内容很多是来源于网上的资料，然后按照自己学习情况进行的总结。
我的个人博客：<a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.hjljy.cn" tar
et="_blank" rel="nofollow ugc">海加尔金鹰
<h2 id="什么是建造者模式">什么是建造者模式</h2>
在 Java 当中，当需要构建一个对象，并且这个对象的某些属性在构建好后有默认值或者自定义
值，通常有三种方法。
方法一：直接给属性一个默认值。缺点：不够灵活
方法二：通过构造器赋值。缺点：构造器的传参过多，不好分辨。
方式三：通过 set 方法赋值。缺点：如果需要设置的参数较多需要一个一个设置。
个人感觉的实际场景：
淘宝购物买电脑时经常看见的套餐，套餐一：鼠标 + 键盘，套餐二：鼠标 + 键盘 + 保护膜等很多的
餐。只需要选择套餐几就可以了，不需要一个一个选择。然后你觉得套餐里面的某个东西不喜欢，想
成其他的，还可以联系客服换。
然后就有了建造者模式。
<blockquote>
1、定义：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<br

2、主要作用：在用户不知道对象的建造过程和细节的情况下就可以直接创建复杂的对象。
3、如何使用：用户只需要给出指定复杂对象的类型和内容，建造者模式负责按顺序创建复杂对象（
内部的建造过程和细节隐藏起来）
4、解决的问题：
（1）、方便用户创建复杂的对象（不需要知道实现过程）
（2）、代码复用性 & 封装性（将对象构建过程和细节进行封装 & 复用）
5、注意事项：与工厂模式的区别是：建造者模式更加关注与零件装配的顺序，一般用来创建更为复
的对象 <sup class="footnotes-ref" id="footnotes-ref-1">1
/sup>
</blockquote>
上面是引用别人整理好的知识点。
<h2 id="建造者模式的实现">建造者模式的实现</h2>
<h2 id="常见的四大角色">常见的四大角色</h2>
<blockquote>
产品角色（Product）：它是包含多个组成部件的复杂对象，由具体建造者来创建其各个滅部件

抽象建造者（Builder）：它是一个包含创建产品各个子部件的抽象方法的接口，通常还包含一个返回
杂产品的方法 getResult()。
具体建造者(Concrete Builder）：实现 Builder 接口，完成复杂产品的各个部件的具体创建方法。<b
>
指挥者（Director）：它调用建造者对象中的部件构造与装配方法完成复杂对象的创建，在指挥者中
涉及具体产品的信息。<sup class="footnotes-ref" id="footnotes-ref-2"><a href="#footnotes-
ef-2">2</sup>
</blockquote>
个人理解最核心的就是具体建造者(Concrete Builder）和产品角色（Product），在实际创建当
，具体建造者负责具体复杂对象创建，然后返回创建的对象（也就是产品）就可以看成一个简单的建
者模式。
<h2 id="代码实现">代码实现</h2>
代码特征：链式编程，返回自身。
实现方式一：基于四大角色的方式
实现方式二：通过静态内部类方式实现零件无序装配话构造
具体代码见文章：<a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.jianshu.c
m%2Fp%2F47329a94f5dc" target="_blank" rel="nofollow ugc">Java 设计模式——建造者模式
Builder Pattern）

<h2 id="建造者模式的优缺点">建造者模式的优缺点</h2>
<h2 id="优点">优点</h2>
网上资料说了很多的优点，但是对我目前的感受来说，主要是:

不必关心具体的创建过程，我只需要传递相关参数信息，就可以返回我想要的具体实例。
相对于构造器创建对象，可以更加直观的看到参数和属性的关系，代码可读性更好。
不同的具体建造者之间相互独立，在进行扩展的时候就比较方便。

<h2 id="缺点">缺点</h2>
网上资料说了一些的缺点，我目前比较直观的感受只有一点：代码量相对来说变多了。
<h2 id="后记">后记</h2>

在很多的资料当中都将建造者模式和工厂模式进行了对比分析，不过目前还没有仔细学习工厂模，等学完了在进行比较。
在 GitHub 找到上一个不错的设计模式代码库：https://github.com/iluwatar/java-design-patterns
在 quartz 定时框架当中就用到了大量的建造者模式，例如
JobBuilder，TriggerBuilder，CronScheduleBuilder 等等。在使用 quartz 的时候就很充分的感受建造者模式的好处了。

<hr>
<li id="footnotes-def-1"> Java 设计式——建造者模式（Builder Pattern） ↩

<li id="footnotes-def-2"> 建造者模式（Bulider 模式）详解 ↩

