



链滴

设计模式:Builder(生成器)-- 创建型模式

作者: [sirwsl](#)

原文链接: <https://ld246.com/article/1591783623848>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

一、介绍:

将一个复杂对象的构建与它的表示相分离，使得同样的构建过程可以创建不同的表示

就是我们在创建一系列复杂对象的时候，我们将其过程抽象出来，而每个抽象的部分有不同的实现方式，因此我们在创建对象时，不同的实现方式就导致了创建对象的不同

二、适用性

-

- 当创建复杂对象的算法应该独立于该对象的组成部分以及他们的装配方式时。

- 当构造过程必须允许被构造的对象有不同的表示时

-

三、参与者

Builder

创建一个 product 对象的各个部件的制定抽象接口。

ConcreteBuilder

实现 Builder 接口以构造和装配产品各个部件

定义和跟踪它所创建的表示

提供一个检索产品的接口

Director

构造一个使用 Builder 接口的对象

Product

被构造的复杂对象。包括定义组件的类与这些部件装配的产品接口

四、效果

-

- 可以改变一个产品得到内部表示。

-

Builder 对象提供一个构造产品的抽象接口，该接口隐藏了这个产品的表示和内部结构与如何装配。

-

- 将构造代码与表示代码分开

-

Builder 通过封装一个复杂对象的创建与表示提供了对象的模块性。而使用者直接出接口，而不要知道内部结构类信息。每个 ConcreteBuilder 包含了创建于装配一个特定产品的代码。这些代码只要写一次，由 Director 进行复用，使用在相同部件的基础上构建不同的产品

-

- 实现对构造过程更加精细的控制

-

Builder 模式下可实现对产品一步步的构造，只有当产品完成时才返回对象。Builder 接口其他创建型模式更好的反应产品的构造过程，这使得它可以更加精确的控制构建过程。

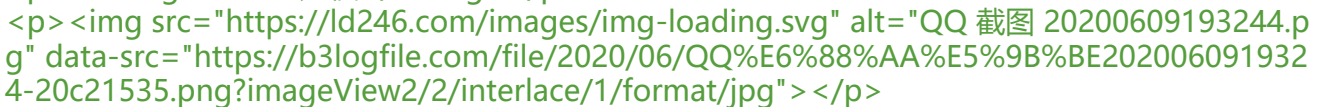
五、实现

1.场景

当我们在购买电脑时候就需要考虑内存、cpu、主板、外观等等因素。当你说出需求后老板会寻对应的部件给你组装。因此我们抽象这些安装步骤出来。

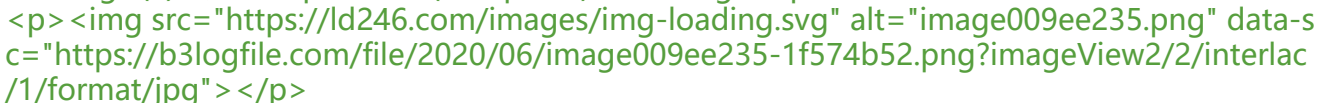
换句话说就是，我们抽象所有的安装步骤为一个接口 (Builder) 然后通过 ConcreteBuilder 实具体的步骤 (选择什么 CPU、主板、内存等等)。然后老板 (Director) 将电脑组装后返回给你一台电脑 (Product)

2.UML 建模

 <https://b3logfile.com/file/2020/06/QQ%E6%88%AA%E5%9B%BE2020060919324-20c21535.png?imageView2/2/interlace/1/format/jpg>

3.编码:

(1)创建一个 product (computer)

 <https://b3logfile.com/file/2020/06/image009ee235-1f574b52.png?imageView2/2/interlace/1/format/jpg>

<p>(2)创建各个步骤的抽象接口</p>

<p></p>

<p>(3)创建各个部分的接口的实现类</p>

<p></p>

<p>(4)对接口的使用（老板开始组装电脑）</p>

<p></p>

<p>(5)测试用例</p>

<p>结果:</p>

<h4 id="4--源码-参见我的github中DesignPatten">4. 源码: 参见我的 github 中 DesignPatten</h4>

https://github.com/sirwsl/DesignPatten

