

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



# 私塾在线 《研磨设计模式》 ——跟着CC学设计系列精品教程

10101010101010101010101010101

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 本节课程概览

### n 学习生成器模式

#### 一：初识生成器模式

包括：定义、结构、参考实现

#### 二：体会生成器模式

包括：场景问题、不用模式的解决方案、使用模式的解决方案

#### 三：理解生成器模式

包括：认识生成器模式、生成器模式的实现、  
使用生成器模式构建复杂对象、生成器模式的优点

#### 四：思考生成器模式

包括：生成器模式的本质、何时选用

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

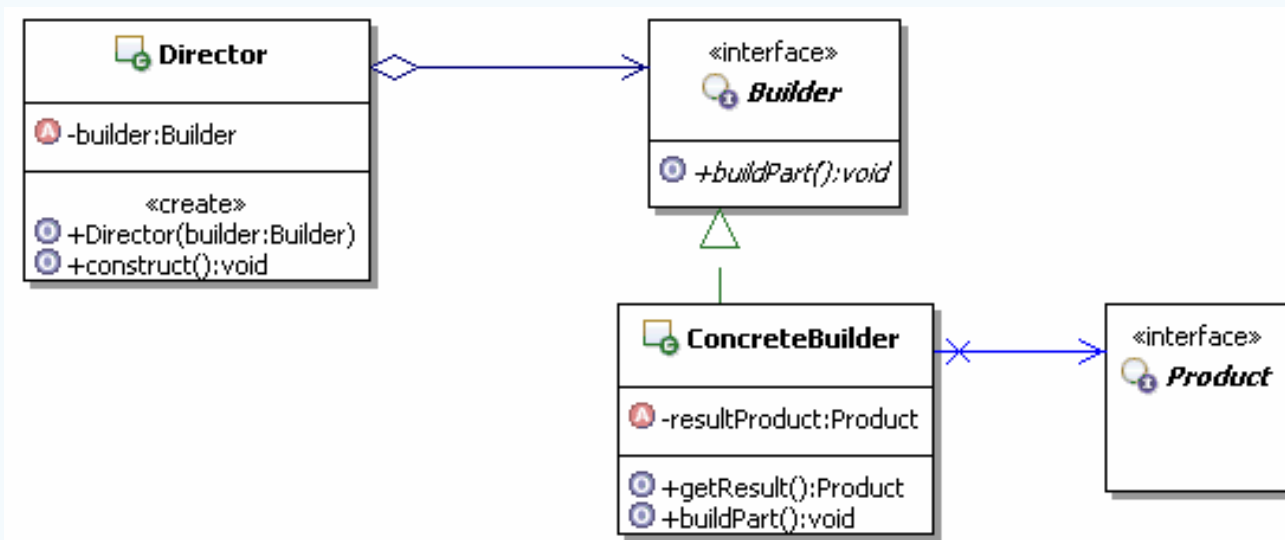
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 初识生成器模式

### n 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。

### n 结构和说明



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 初识生成器模式

### Builder:

生成器接口，定义创建一个Product对象所需的各个部件的操作。

### ConcreteBuilder:

具体的生成器实现，实现各个部件的创建，并负责组装Product对象的各个部件，同时还提供一个让用户获取组装完成后的产品对象的方法。

### Director:

指导者，也被称为导向者，主要用来使用Builder接口，以一个统一的过程来构建所需要的Product对象。

### Product:

产品，表示被生成器构建的复杂对象，包含多个部件。

## 体会生成器模式

### n 继续导出数据的应用框架

对于导出数据的应用框架，通常对于具体的导出内容和格式是有要求的，

假如现在有如下的要求，简单描述一下：

- (1) 导出的文件，不管什么格式，都分成三个部分，分别是文件头、文件体和文件尾
- (2) 在文件头部分，需要描述如下信息：分公司或门市点编号、导出数据的日期，对于文本格式，中间用逗号分隔
- (3) 在文件体部分，需要描述如下信息：表名称、然后分条描述数据。对于文本格式，表名称单独占一行，数据描述一行算一条数据，字段间用逗号分隔。
- (4) 在文件尾部分，需要描述如下信息：输出人

现在就要来实现上述功能。为了演示简单点，在工厂方法模式里面已经实现的功能，这里就不去重复了，这里只关心如何实现导出文件，而且只实现导出成文本格式和XML格式就可以了，其它的不去考虑了。

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

## 体会生成器模式

n 不用模式的解决方案

n 存在的问题

对于不同的输出格式，处理步骤是一样的，但是具体每步的实现是不一样的。

(1) 先拼接文件头的内容

(2) 然后拼接文件体的内容

(3) 再拼接文件尾的内容

(4) 最后把拼接好的内容输出出去成为文件

按照现在的实现方式，就存在如下的问题：

(1) 构建每种输出格式的文件内容的时候，都会重复这几个处理步骤，应该提炼出来，形成公共的处理过程

(2) 今后可能会有很多不同输出格式的要求，这就需要在处理过程不变的情况下，能方便的切换不同的输出格式的处理

换句话说，也就是构建每种格式的数据文件的处理过程，应该和具体的步骤实现分开，这样就能够复用处理过程，而且能很容易的切换不同的输出格式。

做最好的在线学习社区

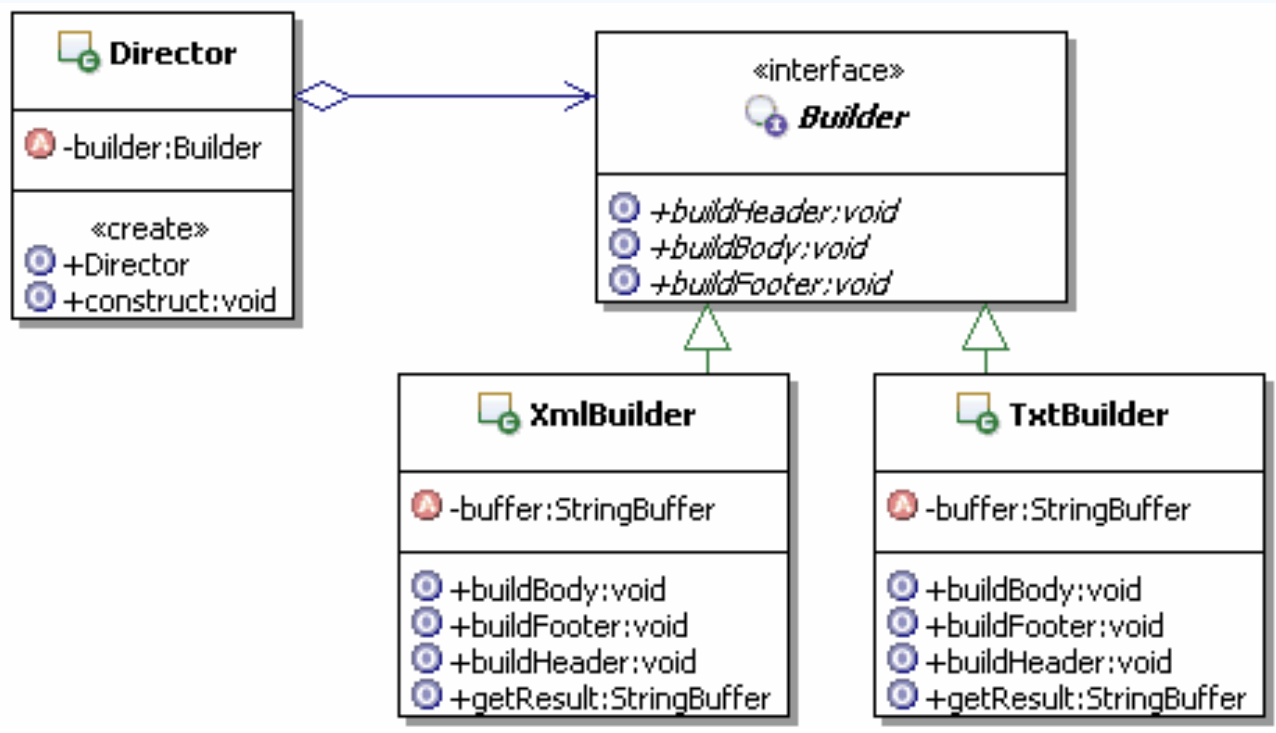
网 址：<http://sishuok.com>

咨询QQ：2371651507



## 体会生成器模式

### n 使用模式的解决方案



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解生成器模式

### n 认识生成器模式

#### 1: 生成器模式的功能

生成器模式的主要功能是构建复杂的产品，而且是细化的，分步骤的构建产品，也就是生成器模式重在解决一步一步构造复杂对象的问题。如果光是这么认识生成器模式的功能是不够的。

更为重要的是，这个构建的过程是统一的，固定不变的，变化的部分放到生成器部分了，只要配置不同的生成器，那么同样的构建过程，就能构建出不同的产品表示来。

直白点说，生成器模式的重心在于分离构建算法和具体的构造实现，从而使得构建算法可以重用，具体的构造实现可以很方便的扩展和切换，从而可以灵活的组合来构造出不同的产品对象。



## 理解生成器模式

### 2: 生成器模式的构成

要特别注意，生成器模式分成两个很重要的部分：

- (1) 一个部分是Builder接口这边，这边是定义了如何构建各个部件，也就是知道每个部件功能如何实现，以及如何装配这些部件到产品中去；
- (2) 另外一个部分是Director这边，Director是知道如何组合来构建产品，也就是说Director负责整体的构建算法，而且通常是分步骤的来执行。

不管如何变化，Builder模式都存在这么两个部分，一个部分是部件构造和产品装配，另一个部分是整体构建的算法。认识这点是很重要的，因为在生成器模式中，强调的是固定整体构建的算法，而灵活扩展和切换部件的具体构造和产品装配的方式，所以要严格区分这两个部分。

在Director实现整体构建算法的时候，遇到需要创建和组合具体部件的时候，就会把这些功能通过委托，交给Builder去完成。

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解生成器模式

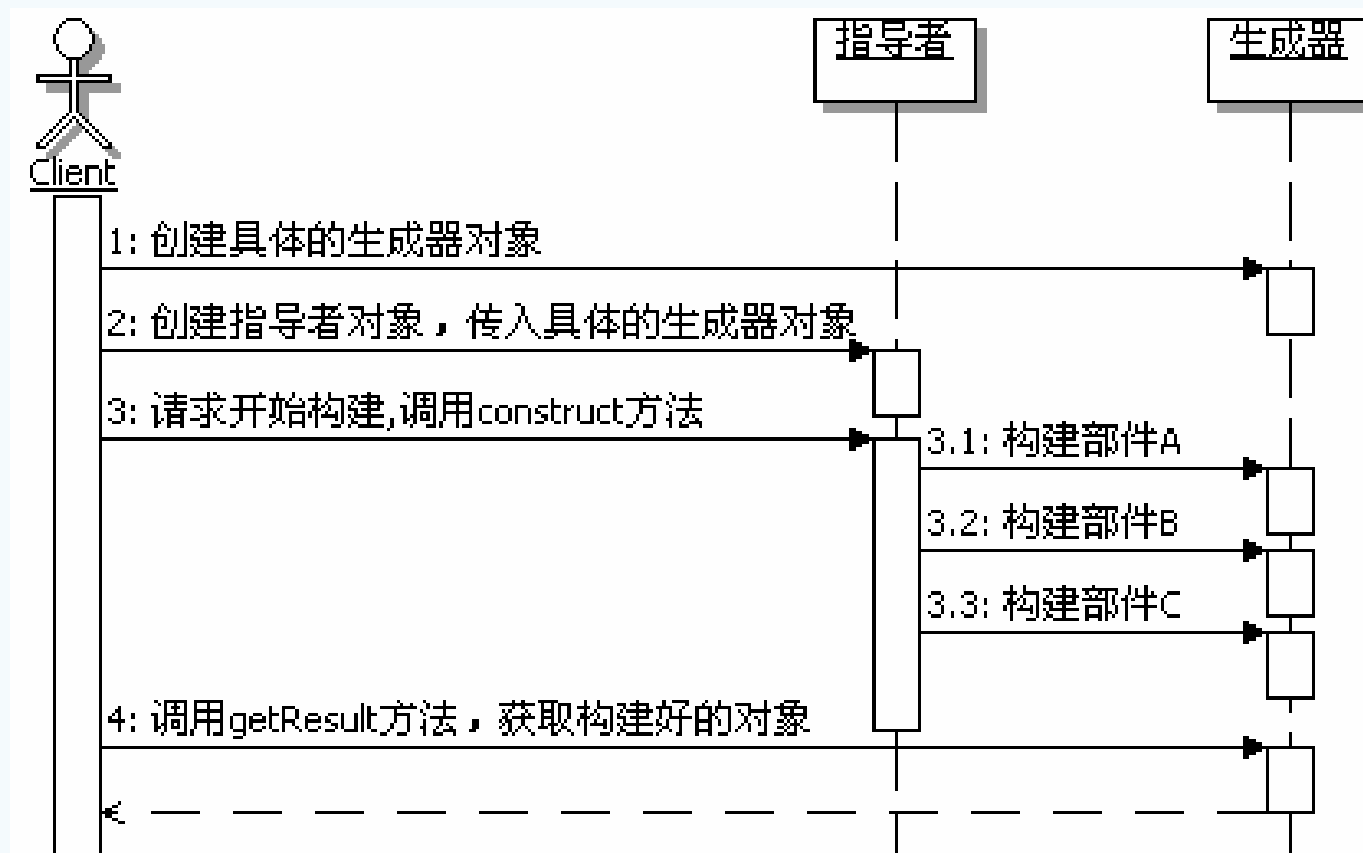
### 3: 生成器模式的使用

应用生成器模式的时候，可以让客户端创造Director，在Director里面封装整体构建算法，然后让Director去调用Builder，让Builder来封装具体部件的构建功能，这就跟前面的例子一样。

还有一种退化的情况，就是让客户端和Director融合起来，让客户端直接去操作Builder，就好像是指导者自己想要给自己构建产品一样。

## 理解生成器模式

### 4: 生成器模式的调用顺序示意图



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解生成器模式

### n 生成器模式的实现

#### (1) 生成器的实现

实际上在Builder接口的实现中，每个部件构建的方法里面，除了部件装配外，也可以实现如何具体的创建各个部件对象，也就是说每个方法都可以有两部分功能，一个是创建部件对象，一个是组装部件。

在构建部件的方法里面可以实现选择并创建具体的部件对象，然后再把这个部件对象组装到产品对象中去，这样一来，Builder就可以和工厂方法配合使用了。

再进一步，如果在实现Builder的时候，只有创建对象的功能，而没有组装的功能，那么这个时候的Builder实现跟抽象工厂的实现是类似的。

这种情况下，Builder接口就类似于抽象工厂的接口，Builder的具体实现就类似于具体的工厂，而且Builder接口里面定义的创建各个部件的方法也是有关联的，这些方法是构建一个复杂对象所需要的部件对象，仔细想想，是不是非常类似呢。

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解生成器模式

### (2) 指导者的实现

在生成器模式里面，指导者承担的是整体构建算法部分，是相对不变的部分。因此在实现指导者的时候，把变化的部分分离出去是很重要的。

其实指导者分离出去的变化部分，就到了生成器那边，指导者知道整体的构建算法，就是不知道如何具体的创建和装配部件对象。

因此真正的指导者实现，并不仅仅是如同前面示例那样，简单的按照一定顺序调用生成器的方法来生成对象，并没有这么简单。应该是有较为复杂的算法和运算过程，在运算过程中根据需要，才会调用生成器的方法来生成部件对象。



## 理解生成器模式

### (3) 指导者和生成器的交互

在生成器模式里面，指导者和生成器的交互，是通过生成器的那些 `buildPart` 方法来完成的。指导者通常会实现比较复杂的算法或者是运算过程，在实际中很可能会有这样的情况：

- a: 在运行指导者的时候，会按照整体构建算法的步骤进行运算，可能先运行前几步运算，到了某一步骤，需要具体创建某个部件对象了，然后就调用 `Builder` 中创建相应部件的方法来创建具体的部件。同时，把前面运算得到的数据传递给 `Builder`，因为在 `Builder` 内部实现创建和组装部件的时候，可能会需要这些数据
- b: `Builder` 创建完具体的部件对象后，会把创建好的部件对象返回给指导者，指导者继续后续的算法运算，可能会用到已经创建好的对象
- c: 如此反复下去，直到整个构建算法运行完成，那么最终的产品对象也就创建好了

通过上面的描述，可以看出指导者和生成器是需要交互的，方式就是通过生成器方法的参数和返回值，来回的传递数据。事实上，指导者是通过委托的方式来把功能交给生成器去完成。

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507



## 理解生成器模式

### （4）返回装配好的产品的方法

在标准的生成器模式里面，在Builder实现里面会提供一个返回装配好的产品的方法，在Builder接口上是没有的。它考虑的是最终的对象一定要通过部件构建和装配，才算真正创建了，而具体干活的就是这个Builder实现，虽然指导者也参与了，但是指导者是不负责具体的部件创建和组装的，因此客户端是从Builder实现里面获取最终装配好的产品。

### （5）关于被构建的产品的接口

在使用生成器模式的时候，大多数情况下是不知道最终构建出来的产品是什么样的，所以在标准的生成器模式里面，一般是不需要对产品定义抽象接口的，因为最终构造的产品千差万别，给这些产品定义公共接口几乎是没有意义的。

## 理解生成器模式

### n 使用生成器模式构建复杂对象

考虑这样一个实际应用，要创建一个保险合同的对象，里面很多属性的值都有约束，要求创建出来的对象是满足这些约束规则的。约束规则比如：保险合同通常情况下可以和个人签订，也可以和某个公司签订，但是一份保险合同不能同时与个人和公司签订。这个对象里面有很多类似这样的约束，那么该如何来创建这个对象呢？

- 1: 使用Builder模式来构建复杂对象，先不考虑带约束
- 2: 使用Builder模式来构建复杂对象，考虑带约束规则
- 3: 进一步，把构建器对象和被构建对象合并

## 理解生成器模式

### n 生成器模式的优缺点

- 1: 松散耦合
- 2: 可以很容易的改变产品的内部表示
- 3: 更好的复用性

## 思考生成器模式

### n 生成器模式的本质

生成器模式的本质是：分离整体构建算法和部件构造

### n 何时选用生成器模式

- 1: 如果创建对象的算法，应该独立于该对象的组成部分以及它们的装配方式时
- 2: 如果同一个构建过程有着不同的表示时