在创建复杂对象的时候，即Field很多的对象的时候。

程序员一般有这两种方法

1. 重叠构造器,即用很多构造器，每个构造器都有不同的参数来初始化该对象。这种方式可行，但是当有许多参数的时候，客户端代码将会难以编写，并且难以阅读。甚至有多个同类型参数时客户端还得注意顺序以防写错。
2. JavaBeans模式，这种方式一般使用一个无参构造器来创建对象，然后调用setter方法来设置每个必要的参数和一些可选参数。这种模式比起重叠构造器而言创建方式较简单，产生的代码读起来也容易。但是由于其构造过程分到了好几个调用中，在构造过程中JavaBean可能处于不一致的状态，而且无法通过构造器的参数来保证该类的一致性,这个对象在后续的使用中可能会出错，这可能是由于其中的参数并不完整所导致的。

这个时候，还有一种替代方法，既能保证像是重叠构造器那样的安全性，也可以保证像JavaBeans一样的可读性，这就是**Builder**模式**的一种形式**。不直接生成想要的对象，而是利用一个builder对象来代替你创建，在创建builder对象调用类似于setter的方法来设置创建对象所需要的各种参数，最后调用一个获取对象的无参方法来生成不可变的对象。

作为builder所创建出来的对象，本身是不可变的，所有的参数默认值都放在一个单独的地方，builder的set方法返回builder自身，所以可以实现**链式调用。**

builder就像个构造器一样，可以加强约束条件、验证约束条件，可指定必要的参数和方便的给可选参数赋值。如果其中某个参数违反了相关规定，则会直接抛出异常，整个对象都会创建失败。而不是像使用无参构造创建的对象一样，无法验证数据一致性。而且由于builder的赋值会调用多个方法的原因，比起单纯使用一个构造器而言，**可以使用多个可变参数(…)**这在某些地方会非常有用。