**public interface** FactoryBean<T> {String ***OBJECT\_TYPE\_ATTRIBUTE*** = **"factoryBeanObjectType"**;  
@Nullable  
 T getObject() **throws** Exception;  
@Nullable  
 Class<?> getObjectType();  
**default boolean** isSingleton() {  
 **return true**;  
 }  
}

* 实现接口

@Component  
**public class** ColorFactory **implements** FactoryBean<Color> {  
 @Override  
 **public** Color getObject() **throws** Exception {  
 System.***out***.println(**"getObject() 调用..."**);  
 **return new** Color();  
 }  
  
 @Override  
 **public** Class<?> getObjectType() {  
 **return** Color.**class**;  
 }  
  
 @Override  
 **public boolean** isSingleton() {  
 **return true**;  
 }  
}  
  
**class** Color {  
}

加入容器中，**applicationContext**.getBean(**"colorFactory"**)其实获取的是T对象。

默认获取的是T对象，要想获取FactoryBean对象在id前加&。即**applicationContext**.getBean(**"&colorFactory"**)。