

Spring中的Template和Callback模式 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/213/2021_2022_Spring_E4_B8_AD_E7_c104_213744.htm Spring中的Callback模式与Template模式合用，随处可见。Template method被广泛的使用，像Servlet就是使用这个模式。Template mothod模式虽然能简化很多重复的代码，但这种模式的也有不少限制。Template mothod将一个功能的实现分成许多小的步骤，在父类中定义了这些步骤的顺序，让子类来具体实现每一个小的步骤。这些小的步骤是protected，以防止用户不正确的使用这些小的步骤而产生异常。这样就产生了一个限制，那就是你需要继承Template然后重新实现具体的小步骤。如果这个Template有许多方法，就像JdbcTemplate,如果你每次继承这个庞大的类，然后只是重写某个小步骤中来订制你自己的功能，就会显得非常笨重，更何况数据库操作使用的如此频繁，难道你每进行一个操作就通过继承订制一个，显然不可能这么做。Spring使用Callback模式与之配合，达到了去掉重复代码的效果，同时增加了很大的灵活性，你只需要实现某些CallBack就可轻松订制出Template。那么什么时候才是Callback模式与Template模式结合的最佳时机呢。显然如果每个具体的步骤都需要真正去具体实现而不是简单的改变参数或设置某个对象就ok的话，使用Callback很难去订制，因为你可能需要传递多个Callback作为参数，并让用户去实现，是用Java的内部类本来就是一个比较丑陋的语法，更何况参数是多个。这相当于你把每个小步骤封装成为接口，然后分别继承之然后实现。显然没有达到方便灵活的效果，这时候直接使

用Template method模式就比结合Callback要好。而如果用户只需要定制一个方法能达到用户的要求，或者更简单，只是设置不同的参数，那么使用Callback就具有很好的灵活性

100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com