## 模板方法模式

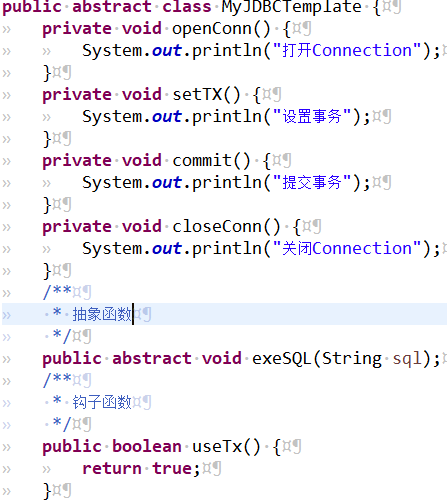
这个其实最容易理解，也是最经常见到的，比如我们在Spring中最常见的就是JdbcTemplate、HibernateTemplate，结合简单的代码来谈谈。

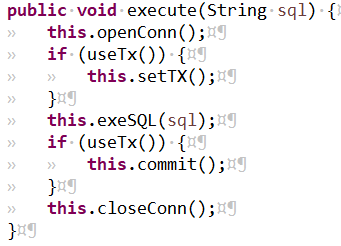
我们把执行SQL语句的步骤简化如下：

|  |
| --- |
| 1. 打开Connection 2. 设置事务 3. 执行SQL 4. 提交事务 5. 关闭Connection |

这个顺序没法省，也没法打乱。大家也看出来了，如果我们每执行一次SQL，都要按这种顺序写代码，代码重复的太多了。模板方法模式的解决方案如下。

模板类：





没啥难度，看execute()这个方法的代码应该就能很好理解了。

这里要多说一下钩子方法，也就是useTx()。

我们执行SQL语句，有的是需要启用事务的（增删改），有的则不需要（查），但是抽象类是不知道哪个SQL用哪个不用的，这个选择权就只能交给了具体的实现类。

useTX()方法并不是执行SQL所必须的步骤，但是它却可以将各个步骤进行挂钩。

你想用事务，将useTX()返回true，execute()就会执行事务相关的方法——通过useTX()这个钩子，把事务方法挂了上去。

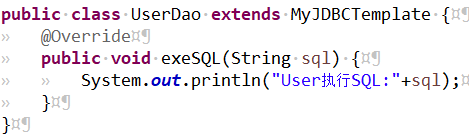
你不想用事务，将useTX()返回false，execute()就不执行事物相关的方法——通过useTX()这个钩子，把事务方法从原方法上脱了钩。

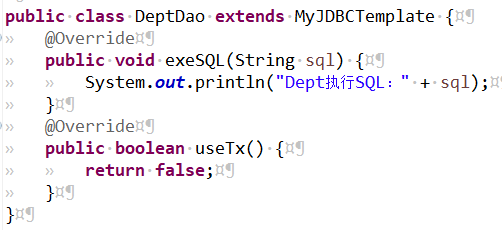
以前那种老式火车，各个车厢是不是用钩子连接的？



而所谓的钩子方法，在抽象类中，我们可以将其设置为空代码，也可以像上面的代码一样，给个默认的实现。那么在子类中我们就可以根据实际情况去决定是否要重写这个方法，从而根据这个方法，来影响具体的步骤。

实现类：





测试结果：

