**注解方式**

Mybatis注解方式就是将SQL语句直接写在接口上,

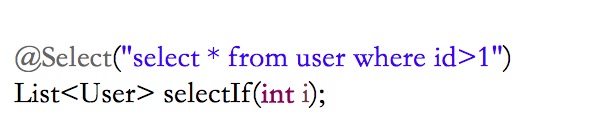
优点：如果需要编写的SQL语句不算复杂，效率会很高。一般情况下，如果系统需求比较简单，可以使用注解的方式编程。

缺点：如果当SQL发生变化时，都要重新编译代码。

 //用注解方法进行select

 @Select({"select id,userAge,userAddress from user where id=#{id}"})

 public User selectByID(int id);

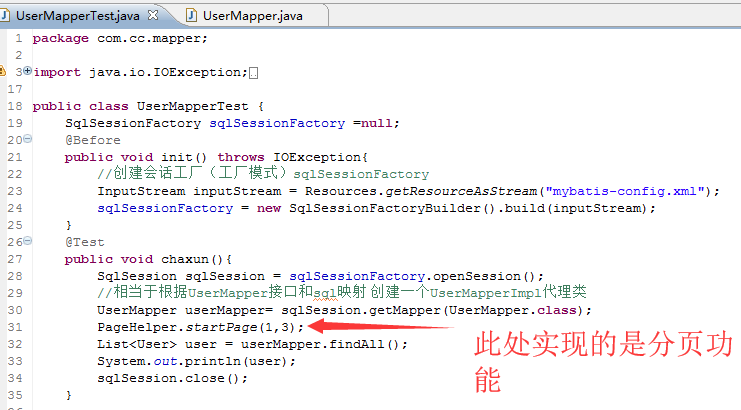


我们这个方法上面加了一个@Select注解，有了这个注解，我们就可以无需在User.xml里写SQL语句，而是直接在注解后面写我们要的SQL。

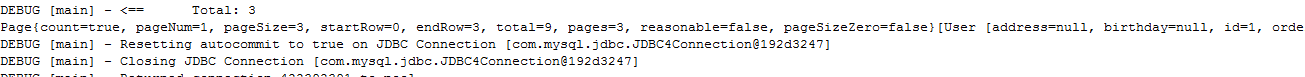
上面的@Select也可以写成

@Select({"select id,userAge,userAddress","from User","where id=#{id}"})

测试代码：

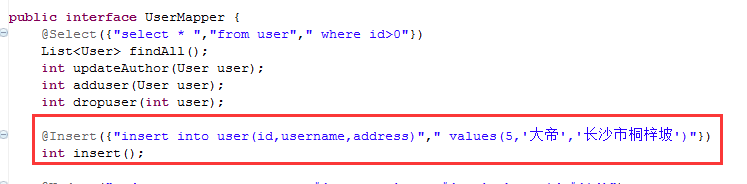


运行结果：

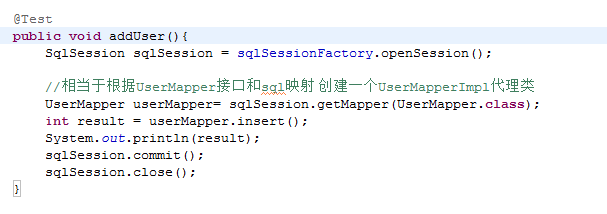




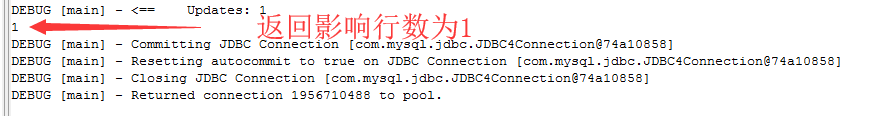
# @Insert注解



测试代码：



运行结果：

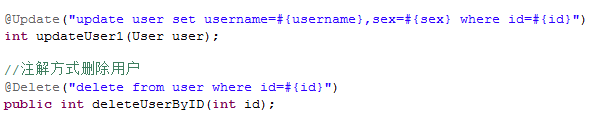


返回非自增主键

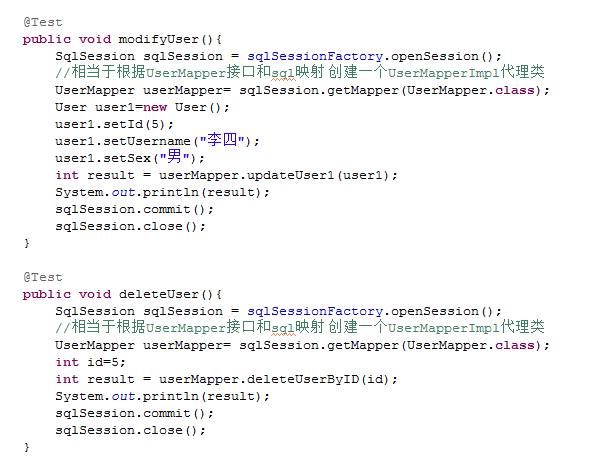
@SelectKey（statement="SELECT LAST\_INSERT\_ID()",keyProperty="id",resultType=Long.class,before=false）

注：MySQL的Order属性是AFTER，因为当前记录的主键值在insert语句执行成功后才能获取到，而Oracle是BEFORE，因为其需要先从序列获取值，然后将值作为主键插入到数据库中。

# 四、@Update和@Delete

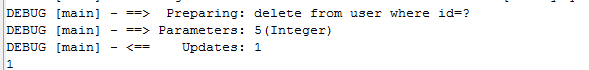


测试代码：



运行结果：





影响行数为一，证明都成功了。

而resultMap的方式代码则是这样使用：

@Results({

@Result(property="id",column="id",id=true)

@Result(property="userName",column="userName")

@Result(property="userAge",column="userAge")

......

})

【注意】注：id=true就表明对应XML的<id>元素 。

从Mybatis3.3.1后，可以为@Results设置id值，这样如果想用同个Results的话，直接引用其id即可，无需每个方法都这么写，如：

@Results(id="userResultMap",value={

@Result(property="id",column="id"),

//其他....

})

@Select(...)

List<User>selectAll();