

1、Spring Data JPA 环境搭建

简介:

官网: <https://spring.io/projects/spring-data>

开源中国: <https://www.oschina.net/p/spring-data>

Spring Data 项目的目的是为了简化构建基于 Spring 框架应用的数据访问计数, 包括非关系数据库、Map-Reduce 框架、云数据服务等; 另外也包含对关系数据库的访问支持

Spring Data 是 Spring 框架的一个子项目, 用于简化基于 Spring 框架的应用的数据访问操作

Spring Data JPA 属于 Spring Data 的子项目, 用于简化 JPA 的持久层开发

Hibernate、JPA 和 Spring Data JPA:

Hibernate: 底层封装的是 JDBC 代码, 提供一个 ORM 思想框架

JPA: 一个 ORM 框架的规范, 是接口, 底层可以调用 Hibernate 框架

Spring Data JPA: 底层调用 JPA 框架, 为了简化 JPA 的开发



Spring Data JPA:

持久层开发, 无需任何实现类, 只需要接口即可




Spring 整合 Hibernate:

导入 jar 包:

spring-aop

 aopalliance.jar	2017/6/29 10:32	Executable Jar File	5 KB
 aspectjrt.jar	2017/6/29 10:32	Executable Jar File	115 KB
 aspectjweaver.jar	2017/6/29 10:32	Executable Jar File	1,816 KB
 spring-aop-4.2.0.RELEASE.jar	2017/6/29 10:32	Executable Jar File	356 KB
 spring-aspects-4.2.0.RELEASE.jar	2017/6/29 10:32	Executable Jar File	57 KB

spring-ioc

 spring-beans-4.2.0.RELEASE.jar	2017/6/29 10:31	Executable Jar File	713 KB
 spring-context-4.2.0.RELEASE.jar	2017/6/29 10:31	Executable Jar File	1,060 KB
 spring-core-4.2.0.RELEASE.jar	2017/6/29 10:31	Executable Jar File	1,028 KB
 spring-expression-4.2.0.RELEASE.jar	2017/6/29 10:31	Executable Jar File	257 KB

spring-jdbc

 spring-jdbc-4.2.0.RELEASE.jar	2017/6/29 10:32	Executable Jar File	412 KB
 spring-tx-4.2.0.RELEASE.jar	2017/6/29 10:32	Executable Jar File	257 KB

spring-orm

 spring-orm-4.2.0.RELEASE.jar	2017/6/29 10:32	Executable Jar File	455 KB
--	-----------------	---------------------	--------










spring-test

 spring-test-4.2.0.RELEASE.jar	2017/6/29 10:32	Executable Jar File	537 KB
---	-----------------	---------------------	--------

spring-web

 spring-web-4.3.3.RELEASE.jar	2017/12/11 21:48	Executable Jar File	795 KB
--	------------------	---------------------	--------




hibernate-required

 antlr-2.7.7.jar	2017/12/11 16:02	Executable Jar File	435 KB
 dom4j-1.6.1.jar	2017/12/11 16:02	Executable Jar File	307 KB
 geronimo-jta_1.1_spec-1.1.1.jar	2017/12/11 16:02	Executable Jar File	16 KB
 hibernate-commons-annotations-5.0...	2017/12/11 16:02	Executable Jar File	74 KB
 hibernate-core-5.0.7.Final.jar	2017/12/11 16:02	Executable Jar File	5,453 KB
 hibernate-jpa-2.1-api-1.0.0.Final.jar	2017/12/11 16:02	Executable Jar File	111 KB
 jandex-2.0.0.Final.jar	2017/12/11 16:02	Executable Jar File	184 KB
 javassist-3.18.1-GA.jar	2017/12/11 16:02	Executable Jar File	698 KB
 jboss-logging-3.3.0.Final.jar	2017/12/11 16:02	Executable Jar File	66 KB

mysql-driver

 mysql-connector-java-5.1.7-bin.jar	2017/12/11 16:02	Executable Jar File	694 KB
--	------------------	---------------------	--------

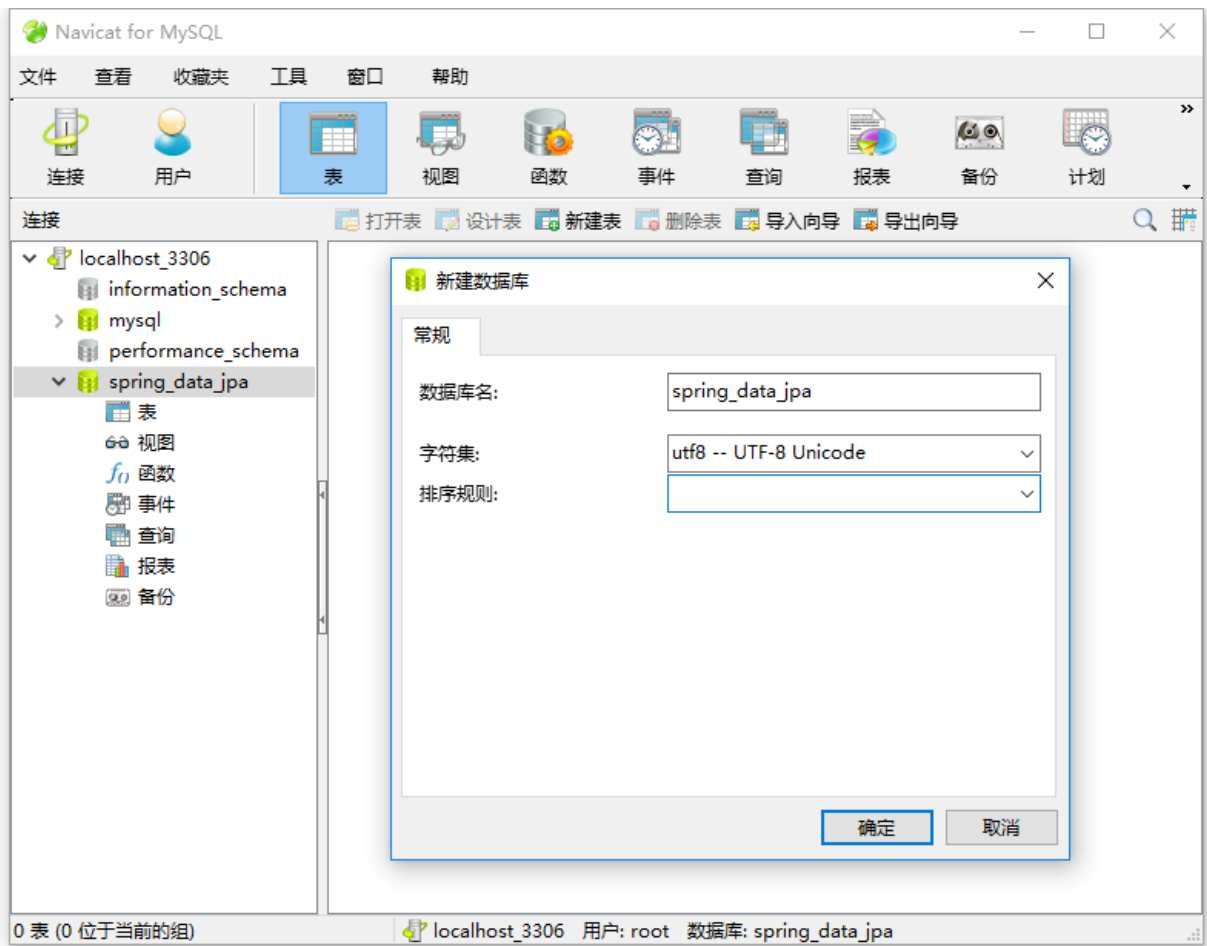
c3p0

 c3p0-0.9.2.1.jar	2017/12/11 16:03	Executable Jar File	414 KB
 hibernate-c3p0-5.0.7.Final.jar	2017/12/11 16:03	Executable Jar File	12 KB
 mchange-commons-java-0.2.3.4.jar	2017/12/11 16:03	Executable Jar File	568 KB

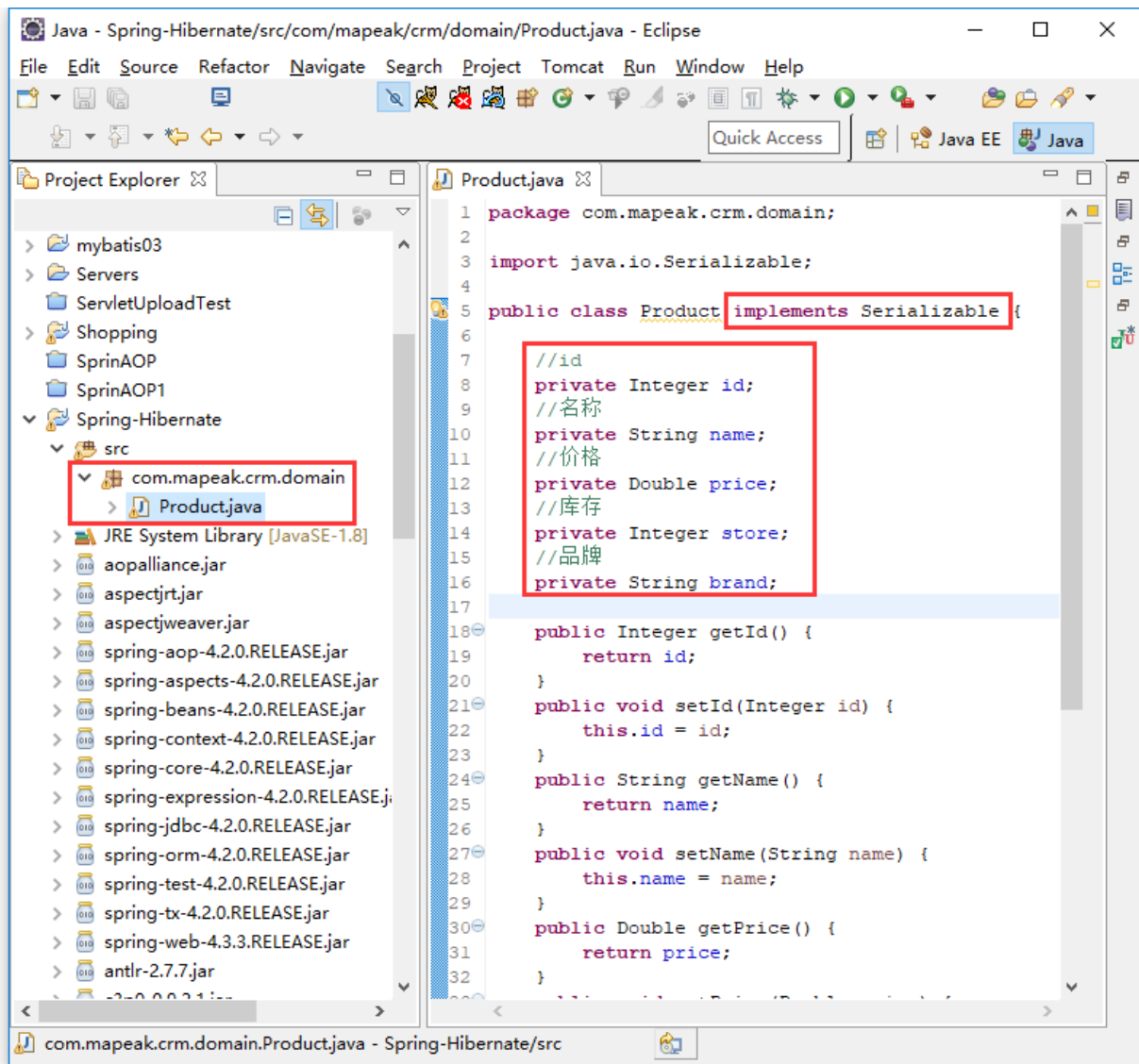
apache-logging

 commons-logging-1.2.jar	2017/12/11 16:02	Executable Jar File	61 KB
---	------------------	---------------------	-------

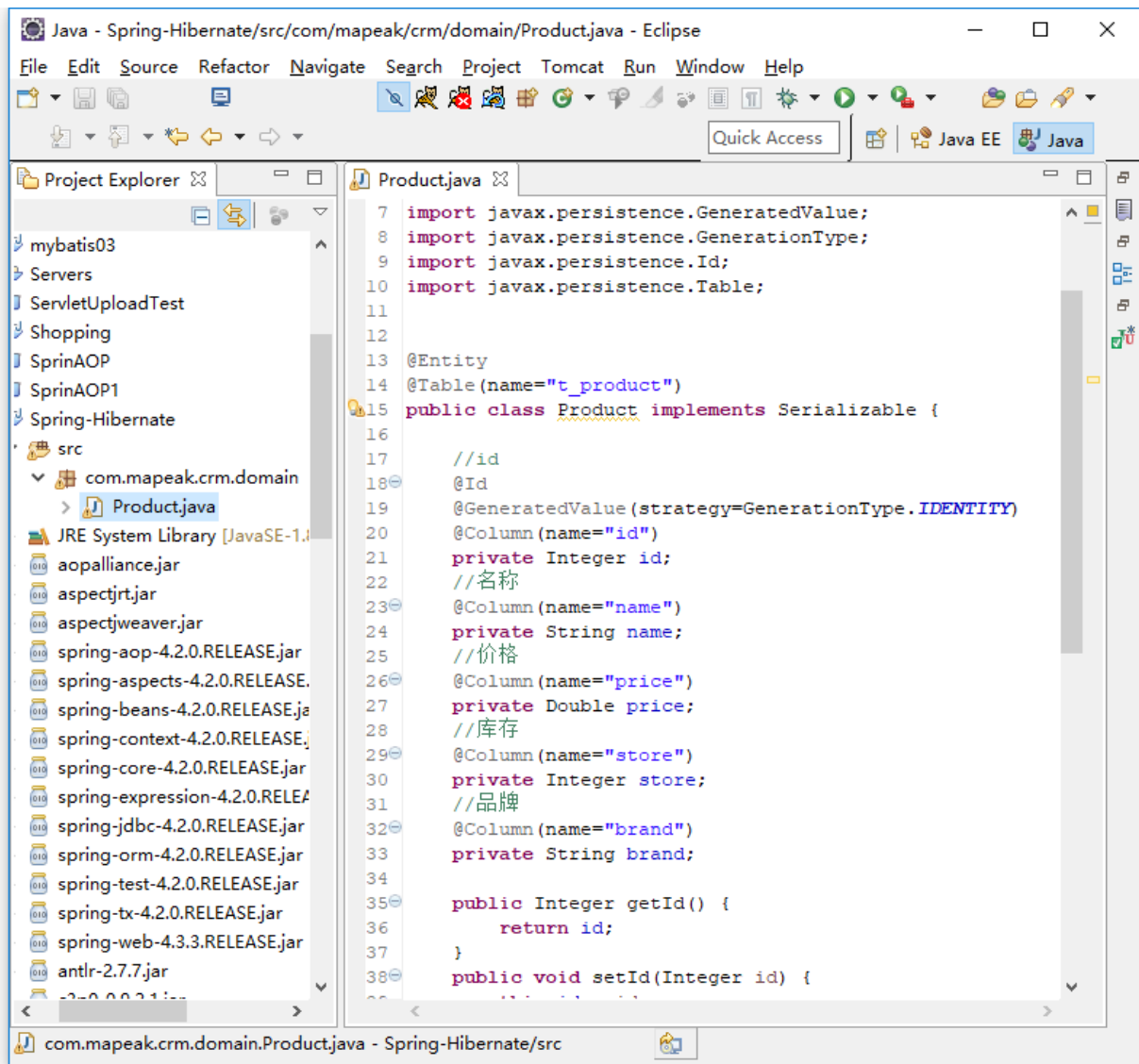
建库:



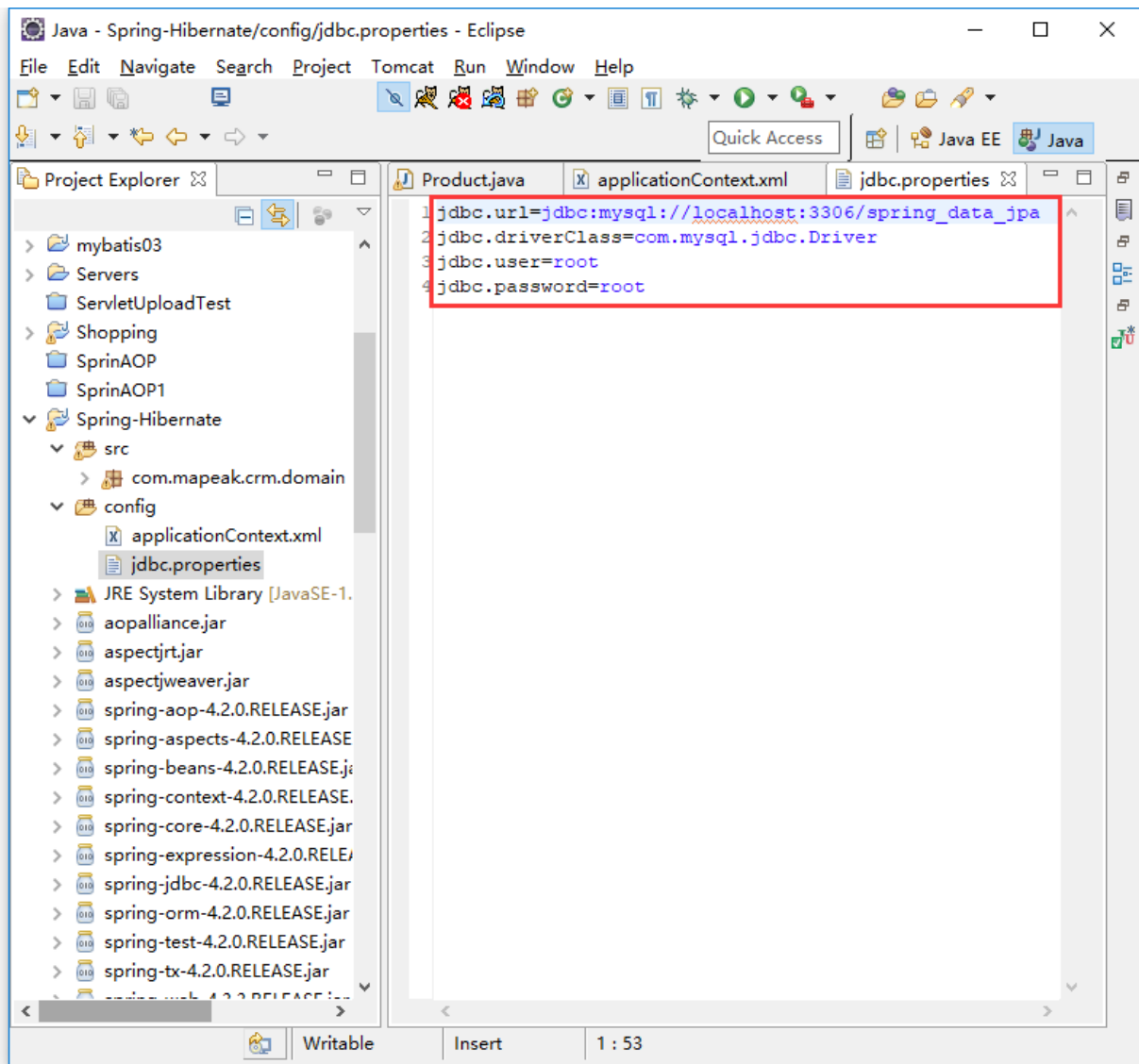
创建实体:

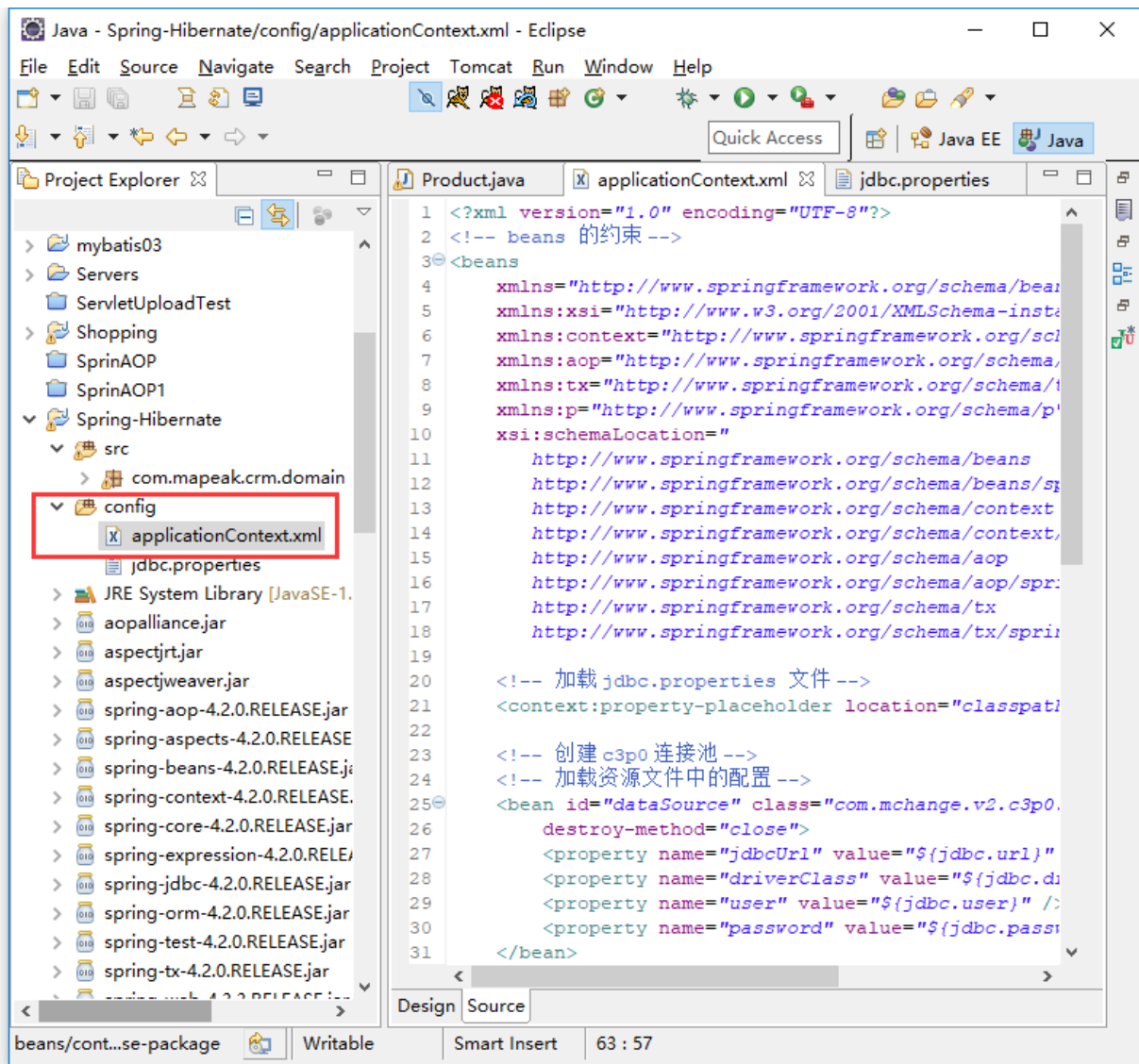


对象映射;



spring 配置文件:
applicationContext.xml





```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!-- beans 的约束 -->
```

```
<beans
```

```
    xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:p="http://www.springframework.org/schema/p"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/aop
        http://www.springframework.org/schema/aop/spring-aop.xsd
        http://www.springframework.org/schema/tx
        http://www.springframework.org/schema/tx/spring-tx.xsd">
```

```
<!-- 加载 jdbc.properties 文件 -->
```

```
<context:property-placeholder location="classpath:jdbc.properties" />
```

```
<!-- 创建 c3p0 连接池 -->
```

```
<!-- 加载资源文件中的配置 -->
```

```
<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"
    destroy-method="close">
    <property name="jdbcUrl" value="${jdbc.url}" />
    <property name="driverClass" value="${jdbc.driverClass}" />
    <property name="user" value="${jdbc.user}" />
    <property name="password" value="${jdbc.password}" />
</bean>
```

```

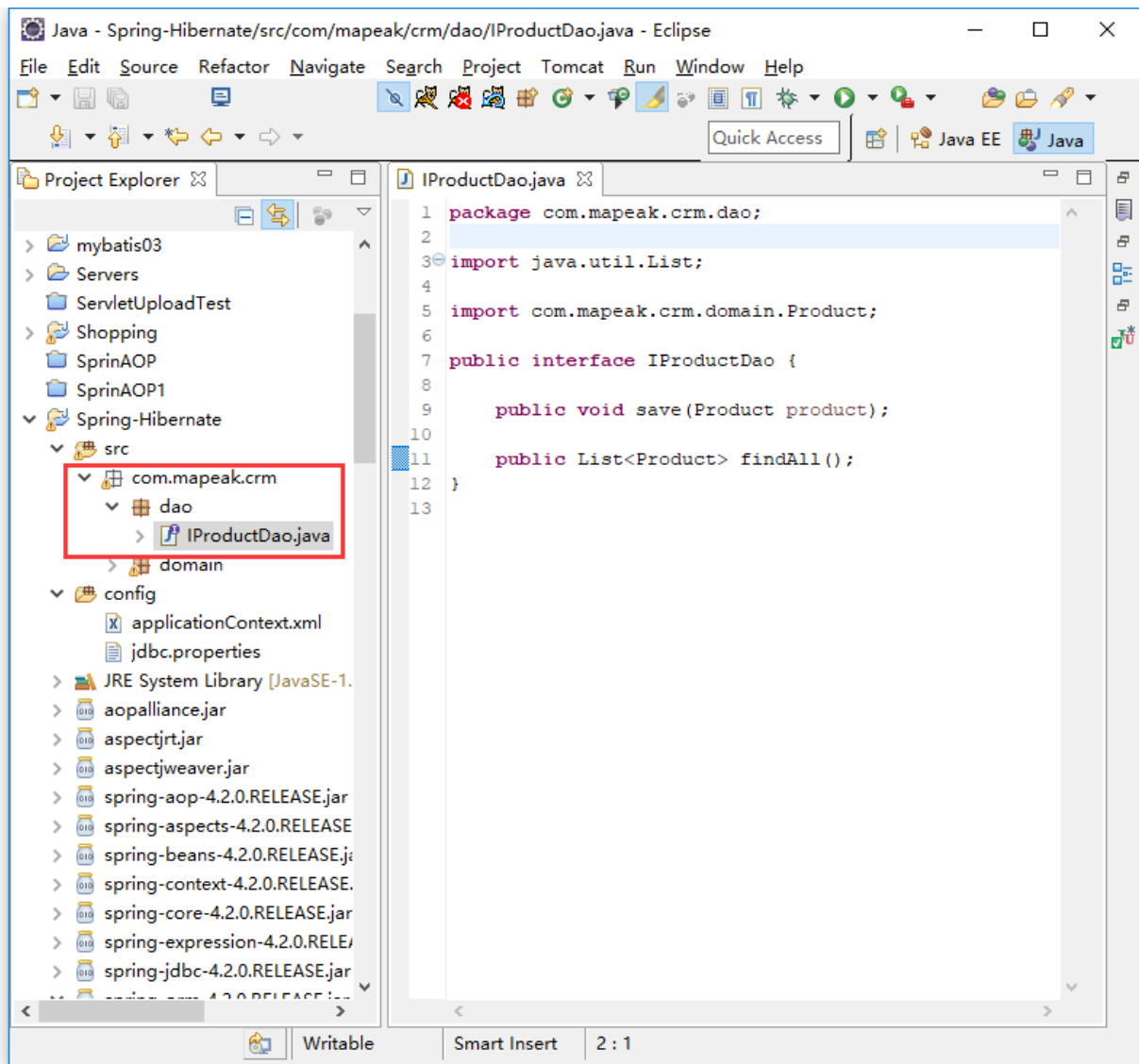
<!-- Spring 整合 Hibernate -->
<bean id="sessionFactory"
    class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">
    <property name="dataSource" ref="dataSource" />
    <property name="hibernateProperties">
        <props>
            <!-- 是否输出 SQL 语句 -->
            <prop key="hibernate.show_sql">true</prop>
            <!-- 自动维护表 -->
            <prop key="hibernate.hbm2ddl.auto">update</prop>
        </props>
    </property>
    <property name="packagesToScan">
        <!-- 扫描实体所在的包 -->
        <list>
            <value>com.mapeak.crm.domain</value>
        </list>
    </property>
</bean>

<!-- 开启 Spring 的事务管理 -->
<!-- 使用注解的方式使用事务 -->
<tx:annotation-driven transaction-manager="transactionManager" />
<bean id="transactionManager"
    class="org.springframework.orm.hibernate5.HibernateTransactionManager">
    <property name="sessionFactory" ref="sessionFactory" />
</bean>

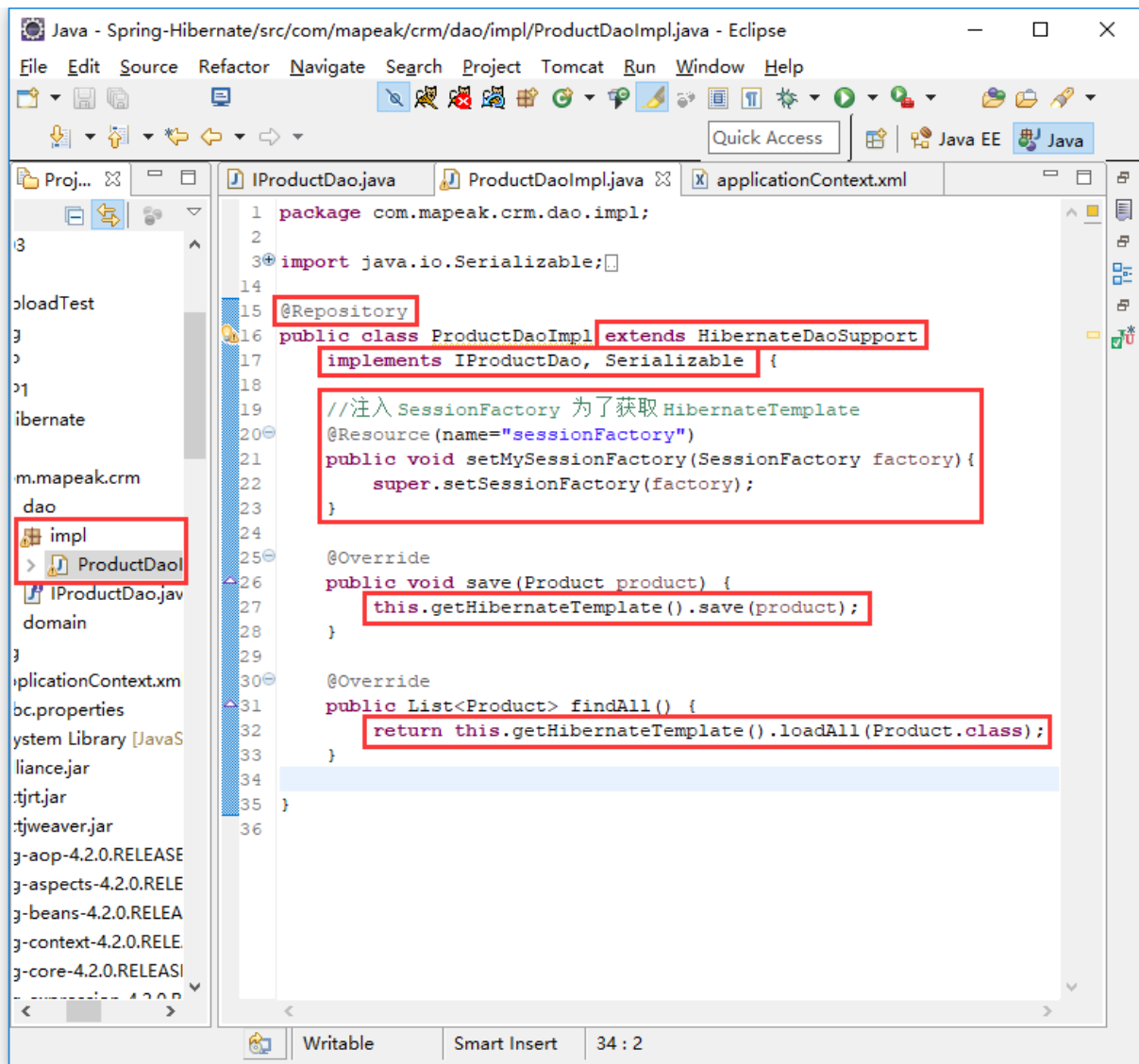
<!-- 开启 Spring 注解功能 -->
<!-- base-package: IOC 注解的类所在包, 即包的范围, Spring 也会扫描该包的子包 -->
<context:component-scan base-package="com.mapeak.crm" />
</beans>

```

DAO:



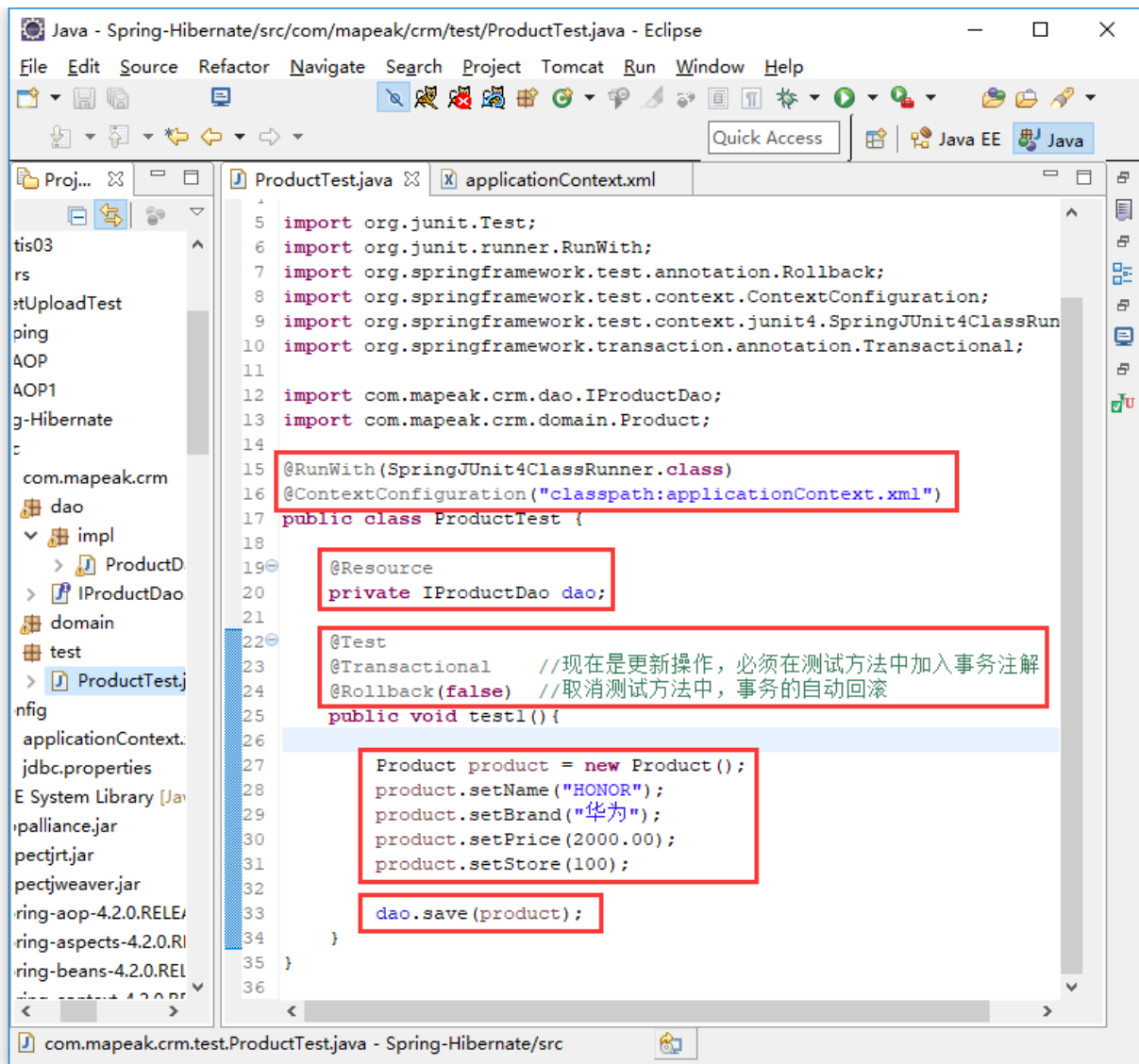
DAO-Impl:

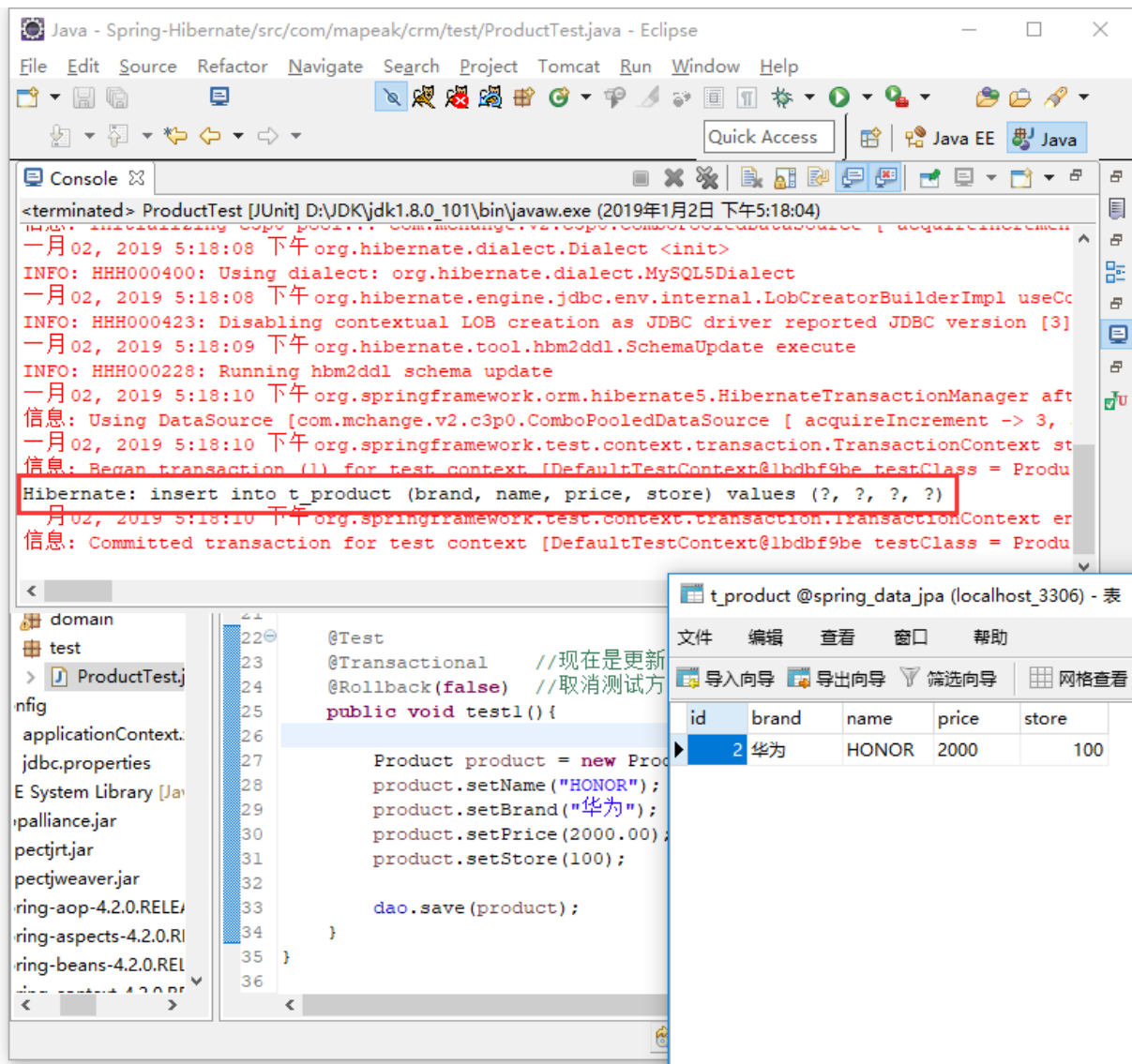


测试:

@Transactional: 增删改操作, 必须在测试方法中加入事务注解

@Rollback(false): 取消测试方法中, 事务的自动回滚





Spring 整合 JPA:

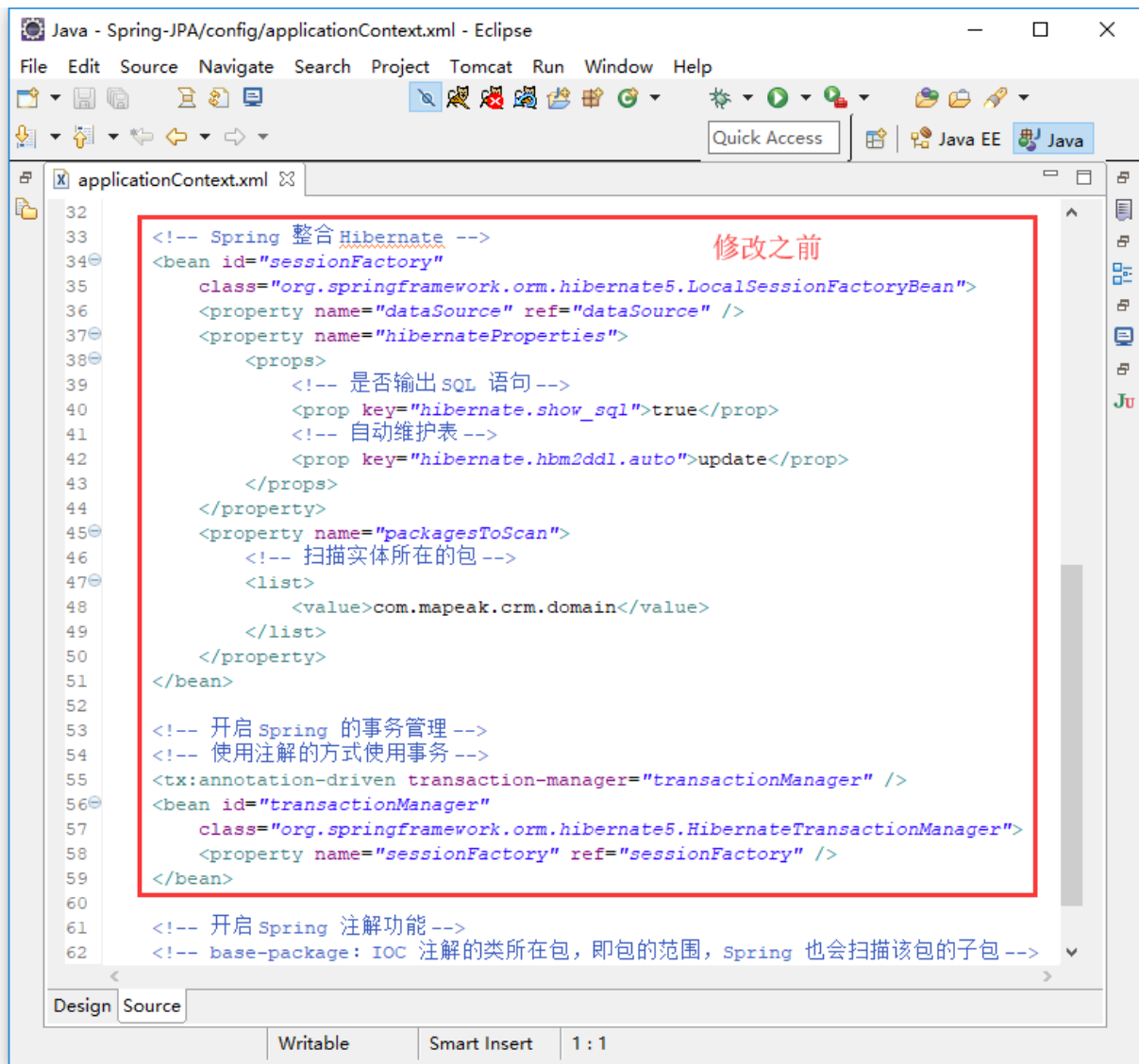
前提: 在 Spring 框架已经整合了 Hibernate 框架的基础之上

导入 jar 包:

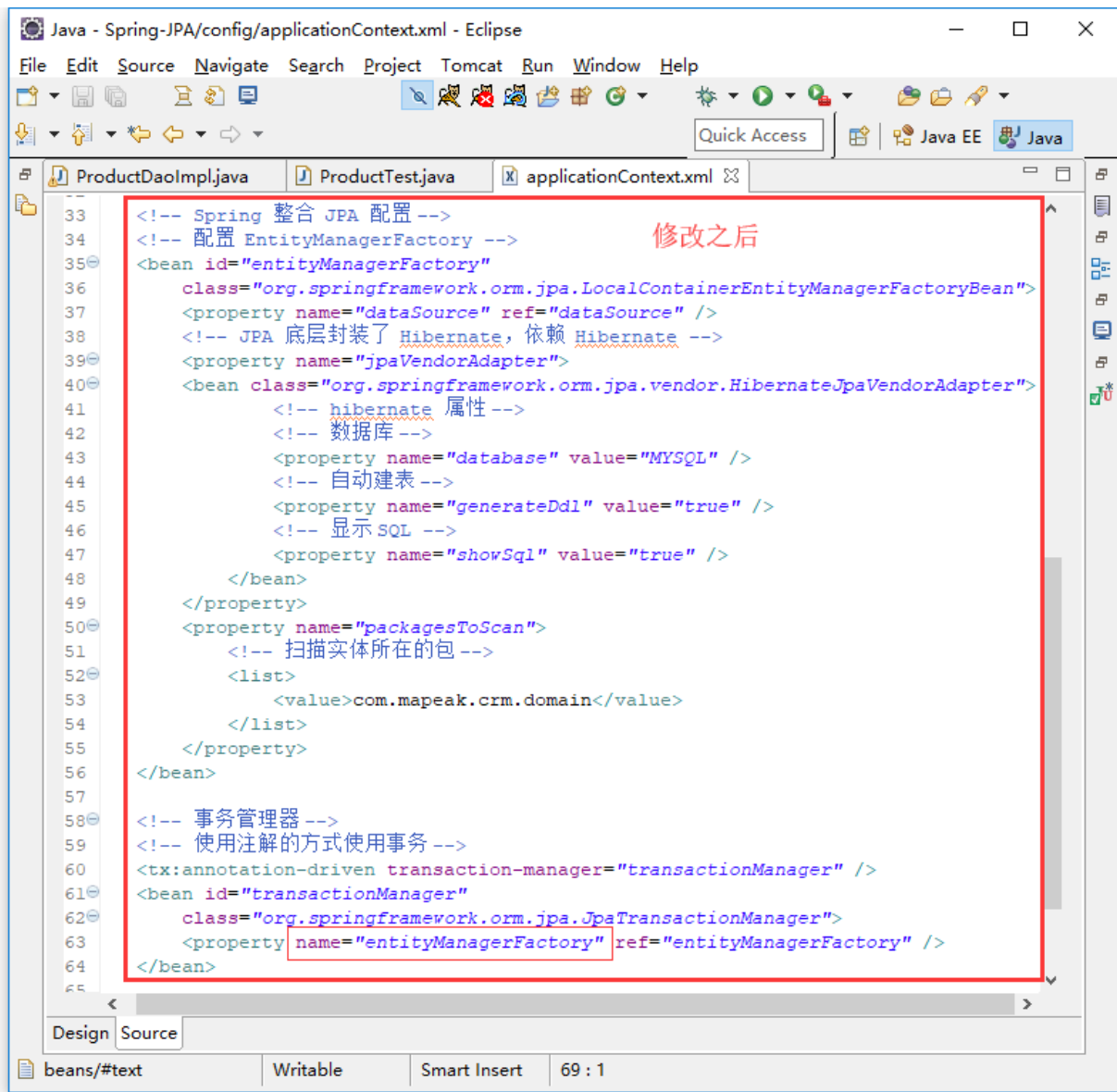
hibernate-entitymanager-5.0.7.Final.jar 2017/6/29 11:18 Executable Jar File 585 KB

修改 applicationContext.xml 配置:

1、Spring 整合 Hibernate 的配置:



2. Spring 整合 JPA 的配置:



修改 DAO 实现:

@PersistenceContext(name="entityManagerFactory"): 从哪个 EntityManagerFactory 获取 EntityManager, 即创建 EntityManager

SQL 查询表达式: 和 Hibernate 的 HQL 用法一样

PS: 没有继承 HibernateDaoSupport

```
1 package com.mapeak.crm.dao.impl;
2
3 import java.io.Serializable;
4
5
6
7
8
9
10
11
12
13
14
15 @Repository
16 //没有继承 HibernateDaoSupport
17 public class ProductDaoImpl implements IProductDao {
18
19     //获取 EntityManager
20     //name: 从哪个 EntityManagerFactory 获取 EntityManager, 即创建 EntityManager
21     @PersistenceContext(name="entityManagerFactory")
22     private EntityManager em;
23
24
25     @Override
26     public void save(Product product) {
27         em.persist(product);
28     }
29
30     @Override
31     //SQL 查询表达式, 和 Hibernate 的 HQL 用法一样
32     public List<Product> findAll() {
33         return em.createQuery("FROM Product").getResultList();
34     }
35 }
36
```

测试:

Java - Spring-JPA/src/com/mapeak/crm/test/ProductTest.java - Eclipse

File Edit Source Refactor Navigate Search Project Tomcat Run Window Help

Quick Access Java EE Java

Console

<terminated> ProductTest (1) [JUnit] D:\JDK\jdk1.8.0_101\bin\javaw.exe (2019年1月3日 上午9:25:01)

INFO: HHH000228: Running hbm2ddl schema update
 一月 03, 2019 9:25:07 上午 org.springframework.test.context.transaction.TransactionContext
 信息: Began transaction (1) for test context [DefaultTestContext@43b0ade testClass = Pro
 Hibernate: insert into t_product (brand, name, price, store) values (?, ?, ?, ?)
 一月 03, 2019 9:25:07 上午 org.springframework.test.context.transaction.TransactionContext
 信息: Committed transaction for test context [DefaultTestContext@43b0ade testClass = Pro
 一月 03, 2019 9:25:07 上午 org.springframework.context.support.GenericApplicationContext d

```

21
22 @Test
23 @Transactional //现在是更新操作，必须在测试方法中加入事务注解
24 @Rollback(false) //取消测试方法中，事务的自动回滚
25 public void test1() {
26
27     Product product = new Product();
28     product.setName("MI");
29     product.setBrand("小米");
30     product.setPrice(1999.00);
31     product.setStore(150);
32
33     dao.save(product);
34 }
35
36
  
```

t_product @spring_data_jpa (localhost_3306) - 表

文件 编辑 查看 窗口 帮助

导入向导 导出向导 筛选向导 网格查看 表单

id	brand	name	price	store
2	华为	HONOR	2000	100
3	小米	MI	1999	150

Spring Data 整合 JPA:

前提: 在 Spring 框架已经整合了 JPA 框架的基础之上

导入 spring-data-jpa jar 包:

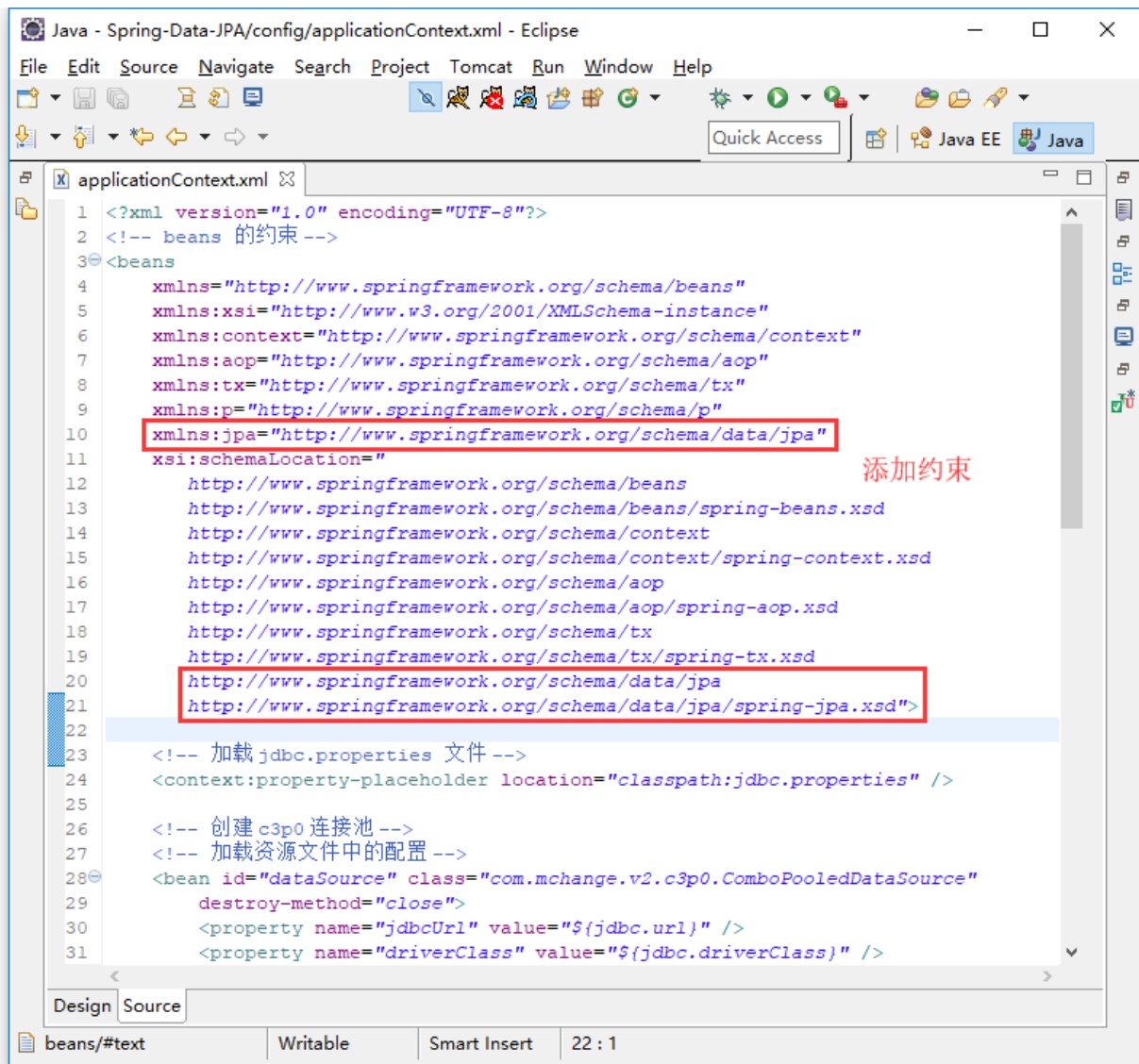
spring-data-commons-1.11.0.RELEASE.jar	2017/6/29 11:36	Executable Jar File	621 KB
spring-data-jpa-1.9.0.RELEASE.jar	2017/6/29 11:36	Executable Jar File	249 KB

导入日志 jar 包:

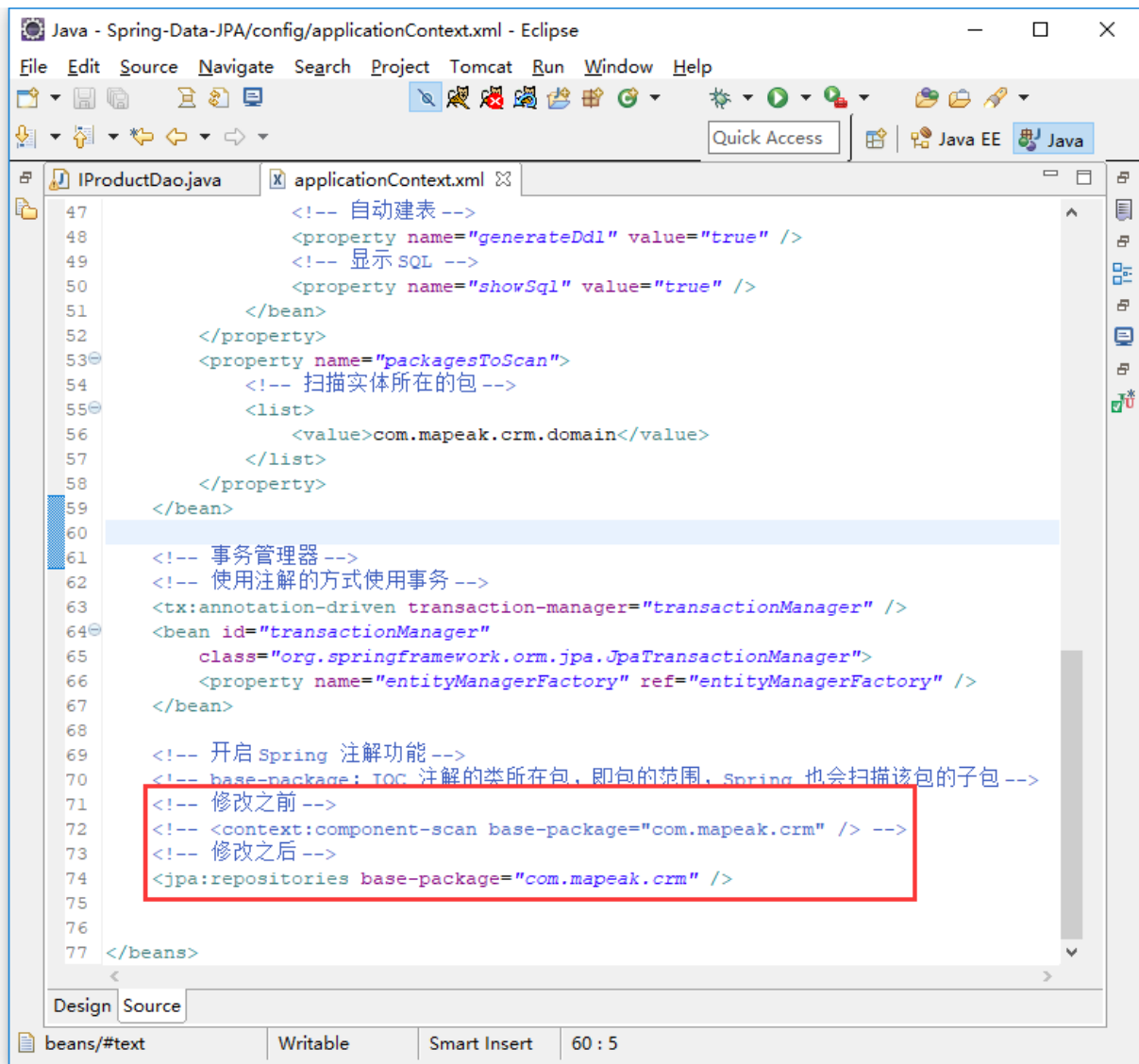
log4j-1.2.16.jar	2017/6/29 11:46	Executable Jar File	471 KB
slf4j-api-1.6.1.jar	2017/6/29 11:46	Executable Jar File	25 KB
slf4j-log4j12-1.7.2.jar	2017/6/29 11:46	Executable Jar File	9 KB

修改 applicationContext.xml 配置:

1、添加约束



2. 包扫描修改



修改 DAO:

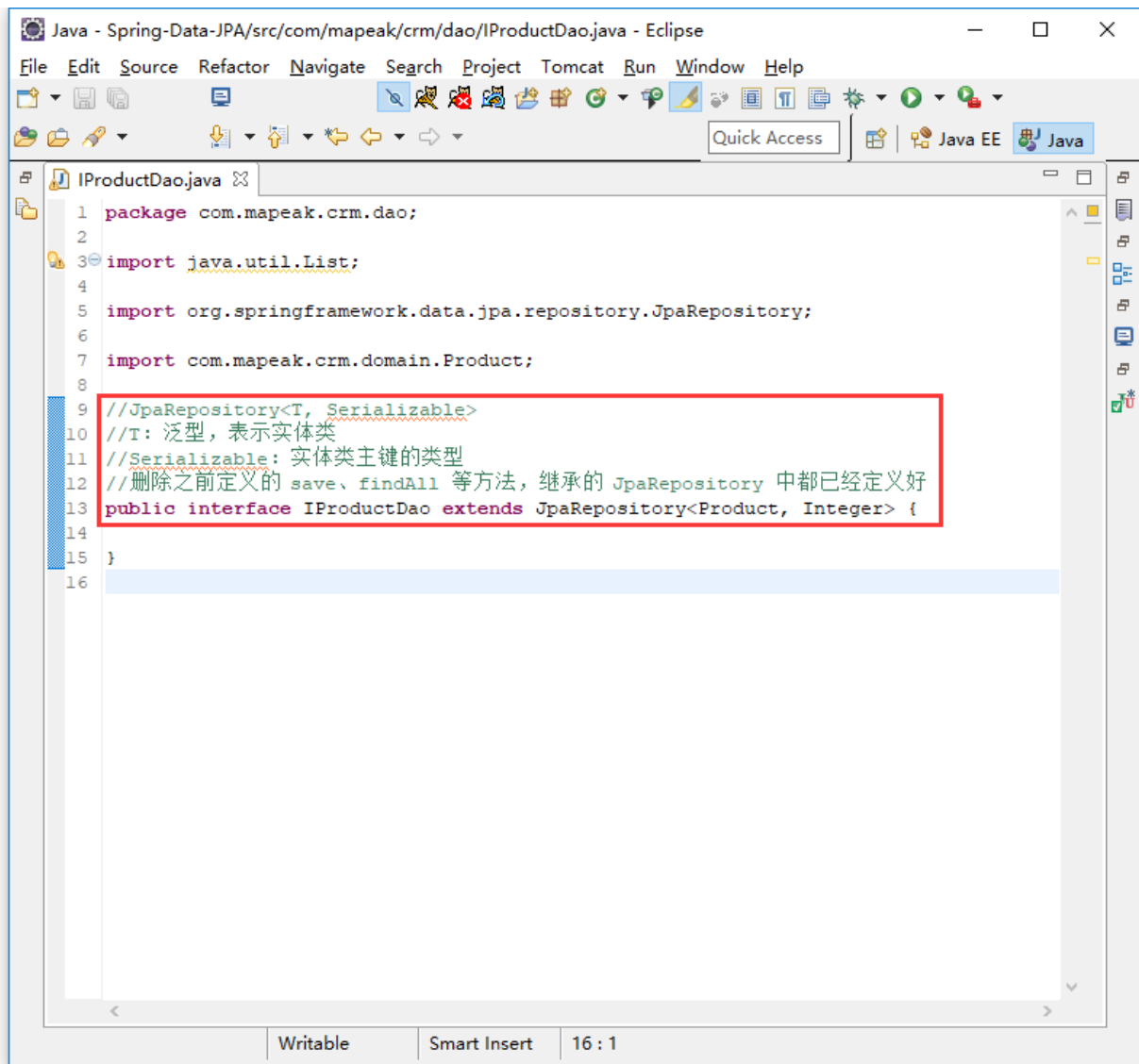
PS: 删除 DAO 实现类

JpaRepository<T, Serializable>

T: 泛型, 表示实体类

Serializable: 实体类主键的类型

删除之前定义的 save、findAll 等方法, 继承的 JpaRepository 中都已经定义



测试:

Java - Spring-Data-JPA/src/com/mapeak/crm/test/ProductTest.java - Eclipse

File Edit Source Refactor Navigate Search Project Tomcat Run Window Help

Quick Access Java EE Java

Console

<terminated> ProductTest (2) [JUnit] D:\JDK\jdk1.8.0_101\bin\javaw.exe (2019年1月3日 上午9:57:12)

log4j:WARN No appenders could be found for logger (org.springframework.test.context.junit4).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Hibernate: insert into t_product (brand, name, price, store) values (?, ?, ?, ?)

```
21  
22 @Test  
23 @Transactional //现在是更新操作，必须在测试方法中加入事务注解  
24 @Rollback(false) //取消测试方法中，事务的自动回滚  
25 public void test1() {  
26  
27     Product product = new Product();  
28     product.setName("OPPO");  
29     product.setBrand("OPPO");  
30     product.setPrice(1099.00);  
31     product.setStore(213);  
32  
33     dao.save(product);  
34 }  
35  
36
```

t_product @spring_data_jpa (localhost_3306) - 表

文件 编辑 查看 窗口 帮助

导入向导 导出向导 筛选向导 网格查看 表单查看

id	brand	name	price	store
2	华为	HONOR	2000	100
4	小米	MI	1999	150
5	OPPO	OPPO	1099	213