黑马程序员

《Java EE 企业级应用开发教程（SSM）（第 2 版）》

教学设计

课程名称：Java EE 企业级应用开发教程

授课年级：XXXX年级

授课学期：第 X 学期

教师姓名：某某老师

2021年6月

课题名称	第 9 章 Spring 的数据库编程	计划课时	6 课时
教学引入	数据库用于处理持久化业务产生的数据，应用程序在运行过程中经常要操作数据库。一般情况下，数据库的操作由持久层来实现。作为扩展性较强的一站式开发框架，Spring 也提供了持久层 Spring JDBC 功能，Spring JDBC 可以管理数据库连接资源，简化传统 JDBC 的操作，进而提升程序数据库操作的效率。本章将对 Spring JDBC 相关知识进行详细讲解。		
教学目标	● 使学生了解 JdbcTemplate 类的作用 ● 使学生熟悉 Spring JDBC 的配置 ● 使学生熟悉 JdbcTemplate 的增删改查操作 ● 使学生理解 Spring 事务管理 ● 使学生理解基于 XML 方式的声明式事务 ● 使学生熟悉基于注解方式的声明式事务		
教学重点	● excute()方法 ● update()方法 ● query()方法 ● 基于 XML 方式的声明式事务 ● 基于注解方式的声明式事务		
教学难点	● 基于 XML 方式的声明式事务		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
教学过程	**第一课时** **（JdbcTemplate 概述、Spring JDBC 的配置、excute()方法）** 一、创设情景，导入新课 数据库像是一个数据的仓库，可以对数据进行组织、存储和管理。涉及到的数据的操作，都要进行数据库的访问。而传统的 JDBC 在操作数据库时，需要先打开数据库连接，执行 SQL 语句，然后封装结果，最后关闭数据库连接等资源。频繁的数据库操作会产生大量重复代码，造成代码冗余，为了解决这些问题，Spring 提供了自己的解决方案，那就是 JdbcTemplate 模板。本节课，将对 JdbcTemplate 概述、Spring JDBC 的配置等内容进行详细讲解。 二、新课讲解 **知识点 1-Jdbctemplate 概述** 教师通过多媒体演示 PPT 内容讲解 JdbcTemplate 概述。		

针对数据库操作，Spring 框架提供了 JdbcTemplate 类，JdbcTemplate 是一个模板类，Spring JDBC 中的更高层次的抽象类均在 JdbcTemplate 模板类的基础上创建。

JdbcTemplate 类继承自抽象类 JdbcAccessor，同时实现了 JdbcTemplate 接口。接下来讲一下抽象类 JdbcAccessor 的属性。

知识点 2-Spring JDBC 的配置

教师讲解 Spring JDBC 的配置。

Spring JDBC 模块主要由 4 个包组成，分别是 core（核心包）、dataSource（数据源包）、object（对象包）和 support（支持包）。

然后再讲一下 dataSource 配置中的 4 个属性，这 4 个属性是 JDBC 连接数据库所必须的。

定义 JdbcTemplate 时，需要将 dataSource 注入到 JdbcTemplate 中，而其他需要使用 JdbcTemplate 的 Bean，也需要将 JdbcTemplate 注入到该 Bean 中。

知识点 3-excute()方法

教师讲解 excute()方法。

execute()方法用于执行 SQL 语句。下面以创建数据表的 SQL 语句为例，来演示此方法的使用，具体步骤如下。

(1) 创建数据库

在 MySQL 中，创建一个名为 spring 的数据库。

(2) 创建项目并引入依赖

在 IDEA 中创建一个名为 chapter09 的 Maven 项目，然后在 pom.xml 文件中加载使用到的 Spring 基础包、Spring 依赖包、MySQL 数据库的驱动 JAR 包、Spring JDBC 的 JAR 包以及 Spring 事务处理的 JAR 包。

(3) 编写配置文件

在 chapter09 项目的 src/main/resources 目录下，创建配置文件 applicationContext.xml，在该文件中配置 id 为 dataSource 的数据源 Bean 和 id 为 jdbcTemplate 的 JDBC 模板 Bean，并将数据源注入到 JDBC 模板中。

(4) 编写测试类

在 src/main/java 目录下创建一个 com.itheima 包，在该包中创建测试类 TestJdbcTemplate，在该类的 main()方法中通过 Spring 容器获取在配置文件中定义的 JdbcTemplate 实例，然后调用 JdbcTemplate 实例的 execute()方法执行创建数据表的 SQL 语句。

三、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

四、布置作业

教师通过高校教辅平台（http://tch.ityxb.com）布置本节课作业以及下节课的预习作业。

第二课时

（update()方法、query()方法）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接导入的方式导入新课

通过上节课的学习，相信我们已经对 Template 模版有了一个初步的了解，并且对 JdbcTemplate 类中的 excute()方法进行了学习，JdbcTemplate 类提供了大量的更新和查询数据库的方法，我们就是使用这些方法来操作数据库的。接下来将学习 JdbcTemplate 类的 update()和 query()方法。

三、新课讲解

知识点 1-update()方法

教师讲解 update()方法。

update()方法可以完成插入、更新和删除数据的操作。下面通过一个案例演示 update()方法的使用，该案例要求添加、更新、删除用户账户，案例具体实现步骤如下所示。

(1) 编写实体类

在 chapter09 项目的 com.itheima 包中，创建 Account 类，在该类中定义 id、username 和 balance 属性，分别表示账户 id、用户名和账户余额，以及其对应的 getter/setter 方法。

(2) 编写 Dao 层接口

在 com.itheima 包中，创建接口 AccountDao，并在接口中定义添加、更新和删除账户的方法。

(3) 实现 Dao 层接口

在 com.itheima 包中，创建 AccountDao 接口的实现类 AccountDaoImpl，并在类中实现添加、更新和删除账户的方法。

(4) 编写配置文件

在 applicationContext.xml 中，定义一个 id 为 accountDao 的 Bean，用于将 jdbcTemplate 注入到 accountDao 实例中。

(5) 测试添加功能

在 com.itheima 包中创建测试类 TestAddAccount，该类主要用于添加用户账户信息。

(6) 测试更新操作

执行完插入操作后，接下来调用 JdbcTemplate 类的 update()方法执行更新操作。在 com.itheima 包中创建测试类 TestUpdateAccount，用于更新用户账户信息。

(7) 测试删除操作

执行完更新操作后，最后调用 JdbcTemplate 类的 delete()方法执行删除操作。在 com.itheima 包中创建测试类 TestDeleteAccount，该类主要用于删除用户账户信息。

知识点 2-query()方法

教师讲解 query()方法。

JdbcTemplate 类中还提供了一系列方法用于处理数据库表的各种查询操作，介绍 JdbcTemplate 类中的常用方法。

接下来通过一个具体的案例演示 query()方法的使用，案例实现步骤如下。

(1) 插入数据

向数据表 account 中插入几条数据。

(2) 编写查询方法

在文件 AccountDao.java 的 AccountDao 接口中，声明 findAccountById()方法，通过 id 查询单个账户信息；声明 findAllAccount()方法，用于查询所有账户信息。

(3) 实现查询方法

在文件 AccountDaoImpl.java 的 AccountDaoImpl 类中，实现 AccountDao 接口中的 findAccountById()方法和 findAllAccount()方法，并调用 query()方法分别进行查询。

(4) 测试条件查询

在 com.itheima 包中创建测试类 FindAccountByIdTest，用于测试条件查询。

(5) 测试查询所有用户信息

在 com.itheima 包中创建测试类 FindAllAccountTest，用于查询所有用户账户信息。

第三课时

（事务管理的核心接口、事务管理的方式、基于 XML 方式的声明式事务）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接导入的方式导入新课

掌握了 JdbcTemplate 类的增删改查操作后，接下来将学习 Spring 事务管理和声明式事务管理。

四、新课讲解

知识点 1-事务管理的核心接口

教师讲解事务管理的核心接口。

Spring 包含一个名为 spring-tx-5.2.8.RELEASE 的 JAR 包，该 JAR 包是 Spring 提供的用于事务管理的依赖包。spring-tx-5.2.8.RELEAS 依赖包提供了 3 个接口实现事务管理。

1. PlatformTransactionManager 接口

PlatformTransactionManager 接口主要用于管理事务。

2. TransactionDefinition 接口

TransactionDefinition 接口中定义了事务描述相关的常量，其中包括了事务的隔离级别、事务的传播行为、事务的超时时间和是否为只读事务。

1) 事务的隔离级别

事务的隔离级别是指事务之间的隔离程度，TransactionDefinition 接口中定义了 5 种隔离级别。

2) 事务的传播行为

事务的传播行为是指处于不同事务中的方法在相互调用时，方法执行期间，事务的维护情况。TransactionDefinition 接口中定义了 7 种事务传播行

为。

3） 事务的超时时间

事务的超时时间是指事务执行的时间界限，超过这个时间界限，事务将会回滚。TransactionDefinition 接口提供了 TIMEOUT_DEFAULT 常量定义事务的超时时间。

4） 事务是否只读

当事务为只读时，该事务不修改任何数据，只读事务有助于提升性能，如果在只读事务中修改数据，会引发异常。

3. TransactionStatus 接口

TransactionStatus 接口主要用于界定事务的状态，通常情况下，编程式事务中使用该接口较多。

知识点 2-事务管理的方式

教师讲解事务管理的方式。

Spring 中的事务管理分为两种方式，一种是传统的编程式事务管理，另一种是声明式事务管理。

知识点 3-基于 XML 方式声明式事务

教师讲解基于 XML 方式的声明式事务。

基于 XML 方式的声明式事务管理是通过在配置文件中配置事务规则的相关声明来实现的。在使用 XML 文件配置声明式事务管理时，首先要引入 tx 命名空间，在引入 tx 命名空间之后，可以使用<tx:advice>元素来配置事务管理的通知，进而通过 Spring AOP 实现事务管理。

接下来通过一个案例演示如何通过 XML 方式实现 Spring 的声明式事务管理。本案例以 9.2 小节的项目代码和数据表为基础，编写一个模拟银行转账的程序，要求在转账时通过 Spring 对事务进行控制。案例具体实现步骤如下。

(1) 导入依赖

在 chapter09 项目的 pom.xml 文件中加入 aspectjweaver 依赖包和 aopalliance 依赖包作为实现切面所需的依赖包。

(2) 定义 Dao 层方法

在 com.itheima 包的 AccountDao 接口中声明转账方法 transfer()。

(3) 实现 Dao 层方法

在 com.itheima 包的 AccountDaoImpl 实现类中实现 AccountDao 接口中的 transfer()方法。

(4) 修改配置文件

修改 chapter09 项目的配置文件 applicationContext.xml，添加命名空间等相关配置代码。

(5) 测试系统

在 chapter09 项目的 com.itheima 包中创建测试类 TransactionTes。

(6) 使用事务管理测试系统

在文件 applicationContext.xml 中添加事务管理的配置。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给

予指导。

五、布置作业

教师通过高校教辅平台（http://tch.ityxb.com）布置本节课作业以及下节课的预习作业。

第三课时

（基于注解方式的声明式事务、案例：实现用户登录）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接导入的方式导入新课

掌握了 JdbcTemplate 类的增删改查操作以及基于 XML 方式的声明式事务后，接下来将学习另一种声明式事务管理和本章案例的实现。

五、新课讲解

知识点 1-基于注解方式的声明式事务

教师讲解基于注解方式的声明式事务。

Spring 提供了@Transactional 注解实现事务管理，@Transactional 注解和 XML 文件中<tx:advice>元素有相同的功能。

接下来对上一小节的案例进行修改，以注解方式来实现项目中的事务管理，具体实现步骤如下。

(1) 创建配置文件

在 chapter09 项目的 src/main/resources 目录下，创建配置文件 applicationContext-annotation.xml，在该文件中声明事务管理器等配置信息。

(2) 修改 Dao 层实现类

在 AccountDaoImpl 类的 transfer()方法上添加事务注解@Transactional。

(3) 编写测试类

在 chapter09 项目的 com.itheima 包中创建测试类 AnnotationTest。

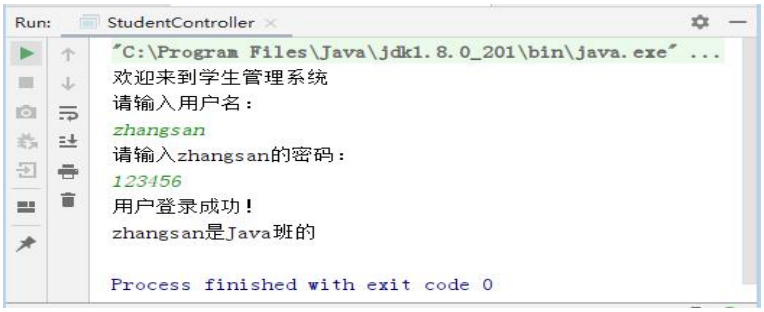
知识点 2-案例：实现用户登录

教师讲解案例：实现用户登录。

案例目标：

通过所学的 Spring 数据库编程知识，实现学生管理系统的登录功能。本案例要求学生在控制台输入用户名密码，如果用户账号密码正确则显示用户所属班级，如果登录失败则显示登录失败。

实现用户登录项目运行成功后控制台效果如图所示。



思路分析：

根据学生管理系统及其登录要求，可以分析案例的实现步骤如下。

(1) 为了存储学生信息，需要创建一个数据库。

(2) 为了程序连接数据库并完成对数据的增删改查操作，需要在 XML 配置文件中配置数据库连接和事务等信息。

(3) 在 Dao 层实现查询用户信息的方法。

(4) 在 Controller 层处理业务逻辑，如判断用户输入的用户名与密码是否正确。

实现步骤：

1. 创建数据库

在 MySQL 中的 spring 数据库中创建一个名为 student 的表，student 表的结构如表所示。

student 表结构				
字段名	类型	长度	是否主键	说明
id	int	11	是	学生编号
username	varchar	255	否	学生姓名
password	varchar	255	否	学生密码
course	varchar	255	否	学生班级

2. 编写实体类

在 chapter09 项目的 com.itheima 包中创建 entity 包，在该包下创建 Student 类，在该类中定义 id、username、password 和 course 属性，以及属性对应的 getter/setter 方法。

3. 编写配置文件

在 chapter09 项目的 resources 文件夹下，创建配置文件 applicationContext-student.xml，在该文件中配置 id 为 dataSource 的数据源 Bean 和 id 为 jdbcTemplate 的 JDBC 模板 Bean，并将数据源注入到 JDBC 模板中。

4. 编写 Dao 层方法

在 com.itheima 包下创建 dao 包，在该包下创建 StudentDao 接口，在 StudentDao 接口中声明查询所有用户信息的方法。

5. 实现 Dao 层方法

在 com.itheima.dao 包下创建 Impl 包，在该包下创建 StudentDaoImpl 实现类，在 StudentDaoImpl 类中实现 StudentDao 接口中的 findAllStudent()方法。

6. 编写 Controller 层

在 com.itheima 包中创建 controller 包，在 controller 包中创建 StudentController 类，用于实现用户登录操作。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过高校教辅平台（http://tch.ityxb.com）布置本节课作业以及下节课的预习作业。

第五、六课时（上机练习）

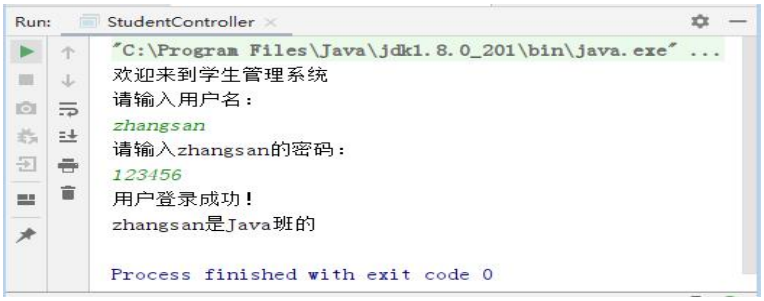
上机一:（考察知识点为 JdbcTemplate 概述、Spring JDBC 的配置、excute()方法、update()方法、query()方法、事务管理的核心接口、事务管理的方式、基于 XML 方式的声明式事务、基于注解方式的声明式事务、案例：实现用户登录）

形式：单独完成

题目：

通过所学的 Spring 数据库编程知识，实现学生管理系统的登录功能。要求学生在控制台输入用户名密码，如果用户账号密码正确则显示用户所属班级，如果登录失败则显示登录失败。

实现用户登录项目运行成功后控制台效果如图所示。



思路分析：

根据学生管理系统及其登录要求，可以分析案例的实现步骤如下。

(1) 为了存储学生信息，需要创建一个数据库。

(2) 为了程序连接数据库并完成对数据的增删改查操作，需要在 XML 配置文件中配置数据库连接和事务等信息。

(3) 在 Dao 层实现查询用户信息的方法。

(4) 在 Controller 层处理业务逻辑，如判断用户输入的用户名与密码是否正确。

教学后记	