|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程名称 | 基于JAVAEE的高级程序设计 | | | | |
| 实验名称 | 阶段实验二：MyBatis框架与Spring框架整合 | | | | |
| 实验时间 |  | | 实验地点 |  | |
| 学 院 | 信息科学与工程学院 | | 专 业 | 计算机科学与技术 | |
| 姓 名 | 蔡涛 | 班 级 | 计科1702 | 学 号 | 20170609037 |
| 同组实验者  姓名 |  | | | | |
| 实验成绩 |  | | 指导教师  （签字） |  | |
| **实验报告内容基本要求参考格式**  一、实验目的  二、实验环境  三、实验步骤/过程  四、实验结果  五、实验分析及反馈 | | | | | |
| 一、实验目的   1. 掌握 Spring 与 MyBatis 整合的方法，理解整合的思路。 2. 掌握 MyBatis 的核心配置。 3. 掌握 MyBatis 的关联映射。 4. 掌握 Spring AOP 实现事务管理。 5. 理解 Spring 框架的依赖注入。   二、实验环境  Eclipse、MySQL  三、实验步骤/过程   1. 准备工作：初始化数据库，创建mybatis数据库，创建user、orders、product和orderitem表，或者导入lab2.sql。 | | | | | |

|  |
| --- |
| 1. 创建项目。 2. 添加Spring框架类库的依赖和MyBatis框架类库的依赖，添加MyBatis与Spring的整合中间件mybatis-spring，添加MySQL JDBC驱动。 3. 编写数据库配置文件：在src/main/java下创建db.properties文件。 4. 编写Spring的配置文件。 5. 利用MyBatis Generator组件生成POJO类和映射文件。拷贝generatorConfig.xml文件至src/main/java路径下，拷贝MBGTest.java至默认的package中。 6. 配置关联映射。 7. 配置User对象与Orders对象之间的一对多关系：   User.java中增加Orders的集合：private List<Orders> ordersList;  UserMapper.xml中增加collection配置。  OrdersMapper.xml中增加一个根据user\_id查询订单的方法。   1. 配置Orders（订单）与Product（商品）之间的多对多关系   Orders.java中增加Product的集合：private List<Product> productList;  OrdersMapper.xml中增加collection配置。  ProductMapper.xml中增加一个根据orders\_id查询订单中对应商品的方法。   1. 实现Service层，在cn.edu.ujn.liukun.lab2.service包中，添加IUserService接口和UserServiceImpl实现类。 2. 整合测试，在src/test/java中，包中创建测试类UserServiceImplTest，并在类中创建findTest方法。 3. 实现订单增加功能。 4. 添加IOrdersService接口。 5. 创建IOrdersService接口的实现类OrdersServiceImpl.java。 6. 修改OrdersMapper.xml文件，增加返回插入Orders对象后的主键。 7. OrdersMapper.xml中增加addOrdersItem的定义。 8. OrdersMapper.java中增加方法。 9. 添加测试类OrdersServiceImplTest.java方法。 |

|  |
| --- |
| 四、实验结果   1. 添加用户：        1. 删除用户（蔡涛1）： |

|  |
| --- |
| 1. 修改用户信息（将蔡涛1修改为蔡涛8）：        1. 查询用户信息：      1. 查询订单：     五、实验分析及反馈  本次实验使用Spring框架和MyBatis框架实现数据表的操作。重点在于Spring和MyBatis的整合过程与方法。使用这两个框架，很好的方便我们对类和数据库的操作，简化了我们的编程过程，使用起来很方便。就比如配置文件书写之后不需要书写sql语句，这有一种全自动的感觉在里边。SQL语句写在xml里，便于统一管理和优化，显得十分灵活。解除sql与程序代码的耦合，通过提供DAO层，将业务逻辑和数据访问逻辑分离，使系统的设计更清晰，更易维护，更易单元测试。SQL和代码的分离，提高了可维护性。 |