# Spring的基本应用

## 习题

1. 填空题
2. 依赖注入的作用就是在使用Spring框架创建对象时，动态的将其所依赖的对象注入到【 】组件中。
3. 在使用Spring框架之后，对象的实例不再由调用者来创建，而是由【 】来创建，Spring容器会负责控制程序之间的关系，而不是由调用者的程序代码直接控制。
4. 在Spring入门程序中只需将Spring的4个基础包以及【 】的JAR包复制到lib目录中，并发布到类路径下即可。
5. 简单来说，BeanFactory就是一个管理Bean的工厂，它主要负责初始化各种Bean，并调用它们的【 】方法。
6. Spring开发所需的JAR包分为两个部分：【 】和【 】。
7. 判断题
8. Spring中基于构造方法的依赖注入通过调用带参数的构造方法来实现，每个参数代表着一个依赖。（ ）
9. 通常在Java项目中，会采用通过FileSystemXmlApplicationContext类来实例化ApplicationContext容器的方式。（ ）
10. 初学者学习Spring框架时，只需将Spring的4个基础包以及commons-logging.jar复制到项目的lib目录，并发布到类路径中即可。（ ）
11. 依赖注入的作用就是在使用Spring框架创建对象时，动态的将其所依赖的对象注入到Bean组件中。（ ）
12. Spring框架采用的是分层架构，它一系列的功能要素被分成20个模块。（ ）
13. 选择题
14. 以下关于Spring核心容器相关说法错误的是。（ ）

A: Spring框架的所有功能都是通过其核心容器来实现的。

B: 创建BeanFactory实例时，需要提供Spring所管理容器的详细配置信息，这些信息通常采用XML文件形式来管理。

C: ApplicationContext不仅包含了BeanFactory的所有功能，还添加了对国际化、资源访问、事件传播等方面的支持。

D: 通常在Java项目中，会采用通过ClassPathXmlApplicationContext类来实例化ApplicationContext容器的方式，而在Web项目中，ApplicationContext容器的实例化工作会交由Web服务器来完成。

1. 以下有关Spring的四个基础包说法正确的是。（ ）

A: Spring的四个基础包，它们分别对应Spring Web容器的四个模块。

B: Spring的四个基础包有spring-core.RELEASE.jar、spring-beans-.RELEASE.jar、spring-context-.RELEASE.jar和spring-aop-.RELEASE.jar。

C: spring-context-.RELEASE.jar是所有应用都要用到的JAR包，它包含访问配置文件以及进行IoC或者DI操作相关的所有类。

D: spring-core.RELEASE.jar包含Spring框架基本的核心工具类，Spring其它组件都要用到这个包里的类，是其它组件的基本核心。

1. 以下有关Spring框架优点的说法正确的是。（ ）

A: Spring具有简单、可测试和松耦合等特点，从这个角度出发，Spring就是应用于任何Java应用的开发中；

B: Spring提供了对AOP的支持，它允许将一些通用任务，如安全、事务、日志等进行集中式处理，从而提高了程序的复用性

C: Spring就是一个大工厂，可以将所有对象的创建和依赖关系的维护工作都交给Spring容器管理，杜绝了组件之间的耦合性

D: Spring增加了Java EE开发中一些API的使用难度

1. Spring的核心容器是其他模块建立的基础，以下哪个不是该容器的组成模块。（ ）

A: Beans模块

B: Core模块

C: Context模块

D: AOP模块

1. 下列选项中，不属于Spring框架优点的是。（ ）

A: 提供强大的、可以有效减少页面代码的标签

B: 声明式事务的支持。

C: 方便解耦、简化开发

D: 方便集成各种优秀框架

1. 简答题
2. Spring依赖注入通常有哪些实现方式，列举并分别进行说明。

## 答案

1. 填空题
2. Bean
3. Spring容器
4. commons-logging
5. 生命周期
6. Spring框架包、第三方依赖包
7. 判断题
8. 对
9. 错
10. 对
11. 对
12. 对
13. 选择题
14. A
15. D
16. B
17. D
18. A
19. 简答题

Spring的实现方式通常有两种，一种是属性setter方法注入，另一种是构造方法注入。属性setter方法注入是指IoC容器使用setter方法来注入被依赖的实例。通过调用无参构造器或无参静态工厂方法实例化Bean后，调用该Bean的setter方法，即可实现基于setter方法的依赖注入。构造方法注入是指IoC容器使用构造方法来注入被依赖的实例。基于构造方法的依赖注入通过调用带参数的构造方法来实现，每个参数代表着一个依赖。

# Spring中的Bean

## 习题

1. 填空题
2. 所谓自动装配，就是将一个Bean自动的注入到到其他Bean的【 】中。
3. @Controller通常作用在控制层，如Spring MVC的【 】，用于将控制层的类标识为Spring中的Bean，其功能与@Component 相同。
4. 【 】注解用于将数据访问层（DAO层）的类标识为Spring中的Bean，其功能与@Component 相同。
5. 对于【 】作用域的Bean，Spring只负责创建，当容器创建了Bean实例后，Bean的实例就交给客户端代码来管理，Spring容器将不再跟踪其生命周期。
6. 实例工厂方式采用直接创建Bean实例的方式，在配置文件中，需要实例化的Bean是通过【 】属性指向配置的实例工厂，然后使用factory-method属性确定使用工厂中的哪个方法。
7. 判断题
8. @Component注解用于描述Spring中的Bean，它是一个泛化的概念，仅仅表示一个组件，并且可以作用在任何层次，使用时只需将该注解标注在相应方法上即可。（ ）
9. Spring提供了3种基于XML的装配方式：设值注入、构造注入和属性注入。（ ）
10. 每次客户端请求singleton作用域的Bean时，Spring容器都会创建一个新的实例，并且不会管那些被配置成singleton作用域的Bean的生命周期。（ ）
11. Spring中使用prototype定义的Bean在Spring容器中将只有一个实例，也就是说，无论有多少个Bean引用它，始终将指向同一个对象。（ ）
12. 对于prototype作用域的Bean，Spring只负责创建，当容器创建了Bean实例后，Bean的实例就交给客户端代码来管理，Spring容器将不再跟踪其生命周期。（ ）
13. 选择题
14. Spring的<bean>元素中的autowire属性取值不包括以下。（ ）

A: default B: byName C: byType D: byId

1. 以下有关Bean的装配方式说法正确的是。（ ）

A: Spring容器支持多种形式的Bean的装配方式，如基于XML的装配、基于注解（Annotation）的装配和自动装配（其中最常用的是基于XML的装配）；

B: Spring提供了3种基于XML的装配方式：设值注入、构造注入和属性注入；

C: 在Spring实例化Bean的过程中，Spring首先会调用Bean的默认构造方法来实例化Bean对象，然后通过反射的方式调用setter方法来注入属性值；

D: 设值注入要求一个Bean必须提供一个有参构造方法并且为需要注入的属性提供对应的setter方法。

1. 以下哪些不属于Spring 4.3版本中Bean的作用域。（ ）

A: application B: request C: response D: globalSession

1. Spring中定义了一系列的注解，以下有关其常用的注解说明错误的是。（ ）

A: @Autowired用于对Bean的属性变量、属性的setter方法及构造方法进行标注，配合对应的注解处理器完成Bean的自动配置工作，默认按照Bean的名称进行装配。

B: @Repository用于将数据访问层（DAO层）的类标识为Spring中的Bean。

C: @Service通常作用在业务层（Service层），用于将业务层的类标识为Spring中的Bean。

D: @Controller通常作用在控制层（如Spring MVC的Controller），用于将控制层的类标识为Spring中的Bean。

1. 下列选项中，不属于Spring中实例化Bean的方式的是。（ ）

A: 构造器实例化 B: 静态工厂方式实例化

C: 实例工厂方式实例化 D: 抽象方法实例化

1. 简答题
2. 请简述基于XML的装配方式中设值注入的要求。

## 答案

1. 填空题
2. Property
3. Controller
4. @Repository
5. prototype
6. factory-bean
7. 判断题
8. 错
9. 错
10. 错
11. 错
12. 对
13. 选择题
14. D
15. C
16. C
17. A
18. D
19. 简答题

Spring中基于XML的装配方式中设值注入需要满足以下两点要求：​

1、Bean类必须提供一个默认的无参构造方法。

2、Bean类必须为需要注入的属性提供对应的setter方法。

# Spring AOP

## 习题

1. 填空题
2. AspectJ框架中注解【 】用于定义切入点表达式，在使用时还需定义一个包含名字和任意参数的方法签名来表示切入点名称。
3. 在Spring配置文件中，<aop:aspect>子元素的pointcut-ref属性用于指定一个已经存在的【 】。
4. 在Spring的配置文件中，配置切面使用的是【 】元素。
5. 在Spring中，使用【 】是创建AOP代理的基本方式。
6. AOP术语中【 】表示AOP框架在特定的切入点执行的增强处理，即在定义好的切入点处所要执行的程序代码。
7. 判断题
8. 如果在同一个连接点有多个通知需要执行，那么在同一切面中，目标方法之前的前置通知和环绕通知的执行顺序是未知的，目标方法之后的后置通知和环绕通知的执行顺序也是未知的。（ ）
9. AspectJ框架中的注解@After用于定义最终final通知，不管是否异常，该通知都会执行。（ ）
10. Spring配置文件中的<beans>元素下可以包含多个<aop:config>元素，一个<aop:config>元素中又可以包含属性和子元素，其子元素包括<aop:pointcut>、<aop:advisor>和<aop:aspect>。（ ）
11. Spring中的AOP代理默认就是使用CGLIB代理的方式来实现的。（ ）
12. Spring 3.0以后，Spring AOP引入了对AspectJ的支持，并允许直接使用AspectJ进行编程，而Spring自身的AOP API也尽量与AspectJ保持一致。（ ）
13. 选择题
14. 以下不属于ProxyFactoryBean类中的常用可配置属性的是。（ ）

A: target

B: proxyInterfaces

C: targetClass

D: interceptorNames

1. 以下哪种类型不是Spring中的通知类型。（ ）

A: 异常通知

B: 前置通知

C: 后置通知

D: 最终通知

1. 关于AspectJ注解的介绍，说法错误的是。（ ）

A: @Aspect用于定义一个切面

B: @Pointcut用于定义切入点表达式

C: @Before用于定义前置通知，相当于BeforeAdvice

D: @After用于定义后置通知，相当于AfterReturningAdvice

1. 以下有关CGLIB代理相关说法正确的是。（ ）

A: CGLIB代理的使用非常简单，但它还有一定的局限性——使用动态代理的对象必须实现一个或多个接口。

B: 如果要对没有实现接口的类进行代理，那么可以使用CGLIB代理。

C: CGLIB是一个高性能开源的代码生成包，在使用时需要另外导入CGLIB所需要的包。

D: Spring中的AOP代理，可以是JDK动态代理，也可以是CGLIB代理。

1. 以下关于Spring AOP 的介绍错误的是。（ ）

A: AOP的全称是Aspect-Oriented Programming，即面向切面编程（也称面向方面编程）。

B: AOP采取横向抽取机制，将分散在各个方法中的重复代码提取出来，这种采用横向抽取机制的方式，采用OOP思想是无法办到的。

C: 虽然AOP是一种新的编程思想，采取横向抽取机制，是OOP的升级替代品。

D: 目前最流行的AOP框架有两个，分别为Spring AOP和AspectJ。

1. 简答题
2. 请对Spring的通知类型进行简要说明。

## 答案

1. 填空题
2. @Pointcut
3. 切入点名称
4. <aop:aspect>
5. ProxyFactoryBean
6. Advice
7. 判断题
8. 对
9. 对
10. 对
11. 错
12. 错
13. 选择题
14. C
15. D
16. D
17. C
18. C
19. 简答题

Spring中主要包含5种通知类型，分别是环绕通知，前置通知，后置通知，异常通知和引介通知。其中，环绕通知表示在目标方法执行前后实施增强，可以应用于日志、事务管理等功能。前置通知表示在目标方法执行前实施增强，可以应用于权限管理等功能。后置通知表示在目标方法执行后实施增强，可以应用于关闭流、上传文件、删除临时文件等功能。异常通知表示在方法抛出异常后实施增强，可以应用于处理异常记录日志等功能。引介通知表示在目标类中添加一些新的方法和属性，可以应用于修改老版本程序（增强类）。

# Spring的数据库开发

## 习题

1. 填空题
2. JdbcTemplate类中还提供了大量的【 】方法来处理各种对数据库表的查询操作。
3. JdbcTemplate类中的【 】方法可以完成插入、更新和删除数据的操作。
4. 【 】就是Junit4用来测试的注解，要测试哪个方法，只需要在相应测试的方法上添加此注解即可。
5. JDBC连接数据库时需要4个基本属性，包括有【 】、url、username和password。
6. JdbcTemplate类的继承关系十分简单。它继承自抽象类【 】，同时实现了JdbcOperations接口。
7. 判断题
8. 在使用Junit进行单一测试时，JUnit视图窗口的进度条为绿色表明运行结果正确；如果进度条为红色则表示有错误，并且会在窗口中显示所报的错误信息。（ ）
9. 定义jdbcTemplate时，需要将dataSource注入到jdbcTemplate中。（ ）
10. JdbcOperations接口定义了在JdbcTemplate类中可以使用的操作集合，包括添加、修改、查询和删除等操作。（ ）
11. JdbcTemplate类中还提供了大量的query()方法来处理各种对数据库表的查询操作。（ ）
12. 在JdbcTemplate类中，提供了大量的更新和查询数据库的方法，我们就是使用的这些方法来操作数据库的。（ ）
13. 选择题
14. Spring JDBC模块主要由4个包组成，其中不包括。（ ）

A: core（核心包）

B: dataSource（数据源包）

C: driverClass（数据库驱动包）

D: support（支持包）

1. JdbcTemplate的直接父类是。（ ）

A: JdbcAccessor

B: JdbcOperations

C: JdbcSupper

D: Object

1. JdbcTemplate类包含在Spring JDBC模块的哪个包中。（ ）

A: 核心包

B: 数据源包

C: 对象包

D: 支持包

1. 下面关于update()方法描述错误的是。（ ）

A: update()方法可以完成插入、更新、删除和查询数据的操作

B: 在JdbcTemplate类中，提供了一系列的update()方法

C: update()方法执行后，会返回受影响的行数

D: update()方法返回的参数是int类型

1. 下面描述中，关于query()方法说法错误的是。（ ）

A: List query(String sql, RowMapper rowMapper)会执行String类型参数提供的SQL语句，并通过RowMapper返回一个List类型的结果。

B: List query（String sql, PreparedStatementSetter pss, RowMapper rowMapper）会根据String类型参数提供的SQL语句创建PreparedStatement对象，通过RowMapper将结果返回到List中。

C: List query（String sql, Object[] args, RowMapper rowMapper）会将args参数绑定到SQL语句中，并通过RowMapper返回一个Object类型的单行记录。

D: queryForList（String sql,Object[] args, class<T> elementType）可以返回多行数据的结果，但必须是返回列表，elementType参数返回的是List元素类型。

1. 简答题
2. 请简述如何使用Junit对程序中的方法进行测试。

## 答案

1. 填空题
2. query()
3. update()
4. @Test
5. driverClassName
6. JdbcAccessor
7. 判断题
8. 对
9. 对
10. 对
11. 对
12. 对
13. 选择题
14. C
15. A
16. A
17. A
18. C
19. 简答题

使用JUnit对程序中的方法进行测试时的步骤如下：

1、先创建一个普通测试方法，并在方法上添加单元测试的注解@Test；

2、添加@Test注解后，在注解处会报错，将鼠标移动至@Test注解处，会弹出错误提示框，单击提示框中的Add JUnit4 library to the build path后，Eclipse会自动将JUnit4的支持包加入到项目中；

3、在执行程序时，只需使用鼠标右键单击测试方法，在弹出的快捷菜单中选择Run As中的JUnit Test选项来运行测试方法即可。

# Spring的事务管理

## 习题

1. 填空题
2. 使用基于注解方式的事务管理需要在使用事务的Spring Bean类或者Bean类的方法上添加注解【 】。
3. Spring的声明式事务管理可以通过两种方式来实现，一种是基于XML的方式，另一种是基于【 】的方式。
4. Spring2.0以后，提供了tx命名空间来配置事务，tx命名空间下提供了【 】元素来配置事务的通知（增强处理）。
5. 在Spring的所有JAR包中，包含一个名为【 】的JAR包，该包就是Spring提供的用于事务管理的依赖包。
6. 声明式事务管理：是通过【 】实现的事务管理。
7. 判断题
8. 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。（ ）
9. 编程式事务管理是通过编写代码实现的事务管理，包括定义事务的开始、正常执行后的事务提交和异常时的事务回滚。（ ）
10. Spring的事务管理简化了传统的事务管理流程，并且在一定程度上减少了开发者的工作量。（ ）
11. Spring的声明式事务管理可以通过两种方式来实现，一种是基于XML的方式，另一种是基于Annotation的方式。（ ）
12. @Responsebody注解与@RequestMapping注解配合使用时，页面中将可以获取到方法的返回值。（ ）
13. 选择题
14. 以下关于@Transactional注解可配置的参数信息及秒正确的是。（ ）

A: value用于指定需要使用的事务管理器，默认为""；

B: read-only用于指定事务是否只读，默认为true；

C: isolation用于指定事务的隔离级别，默认为Isolation.READ\_COMMITTED；

D: propagation用于指定事务的传播行为，默认为Propagation. SUPPORTS；

1. 以下基于XML方式的声明式事务管理配置文件中<tx:advice>元素的子元素<tx:method>属性描述错误的是。（ ）

A: name：该属性为必选属性，它指定了与事务属性相关的方法名

B: propagation：用于指定事务的传播行为，它的默认值为SUPPORTS

C: isolation：该属性用于指定事务的隔离级别，其默认值为DEFAULT

D: read-only：该属性用于指定事务是否只读，其默认值为false

1. 以下有关事务管理方式相关说法错误的是。（ ）

A: 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。

B: 编程式事务管理：是通过AOP技术实现的事务管理，就是通过编写代码实现的事务管理，包括定义事务的开始、正常执行后的事务提交和异常时的事务回滚。

C: 声明式事务管理：其主要思想是将事务管理作为一个“切面”代码单独编写，然后通过AOP技术将事务管理的“切面”代码植入到业务目标类中。

D: 声明式事务管理最大的优点在于开发者无需通过编程的方式来管理事务，只需在配置文件中进行相关的事务规则声明，就可以将事务规则应用到业务逻辑中。

1. 以下有关Spring事务管理及核心接口说法错误的是。（ ）

A: PlatformTransactionManager接口是Spring提供的平台事务管理器，主要用于管理事务。

B: TransactionDefinition接口是事务定义（描述）的对象，该对象中定义了事务规则，并提供了获取事务相关信息的方法。

C: TransactionStatus接口是事务的状态，它描述了某一时间点上事务的状态信息。

D: 在事务管理过程中，传播行为可以控制是否需要创建事务以及如何创建事务，通常情况下，对于数据的查询、插入、更新和删除操作，必须进行事务管理。

1. 下列选项中，哪一个不是Spring中事务管理的核心接口。（ ）

A: PlatformTransactionManager

B: TransactionDefinition

C: TransactionStatus

D: TransactionManager

1. 简答题
2. 请列举PlatformTransactionManager接口提供了有哪些事务操作方法，并对这些方法分别进行简单说明。

## 答案

1. 填空题
2. @Transactional
3. Annotation
4. <tx:advice>
5. spring-tx.RELEASE.jar
6. AOP技术
7. 判断题
8. 对
9. 对
10. 对
11. 对
12. 对
13. 选择题
14. A
15. B
16. B
17. D
18. D
19. 简答题

PlatformTransactionManager接口中提供了三个事务操作方法，具体如下所示：

（1）TransactionStatus getTransaction(TransactionDefinition definition )：用于获取事务状态信息。

（2）void commit(TransactionStatus status)：用于提交事务。

（3）void rollback(TransactionStatus status)：用于回滚事务。

# 初识MyBatis

## 习题

1. 填空题
2. MyBatis的删除操作在映射文件中是通过配置【 】元素来实现的。
3. MyBatis映射文件中<mapper>元素是配置文件的根元素，它包含一个【 】属性，该属性为这个<mapper>指定了唯一的命名空间。
4. 使用MySQL中的【 】函数进行字符串拼接，可以有效防止模糊查询时可能出现的SQL注入问题。
5. mybatis-config.xml作为MyBatis的全局配置文件，配置了MyBatis的运行环境等信息，其中主要内容是获取【 】。
6. 【 】是当前主流的Java持久层框架之一，它与Hibernate一样，也是一种ORM框架。
7. 判断题
8. MyBatis映射文件中<mappers>元素是配置文件的根元素，它包含一个namespace属性，该属性为这个<mapper>指定了唯一的命名空间。（ ）
9. 使用MyBatis框架非常简单，只需在应用程序中引入MyBatis框架lib目录中的全部JAR包以及数据库驱动包即可进行项目运行。（ ）
10. ORM就是一种为了解决面向对象与关系型数据库中数据类型不匹配的技术，它通过描述Java对象与数据库表之间的映射关系，自动将Java应用程序中的对象持久化到关系型数据库的表中。（ ）
11. MyBatis的更新操作也需要进行事务提交。（ ）
12. 创建的映射文件中的SQL语句要以面向对象的写法进行书写。（ ）
13. 选择题
14. MyBatis的删除操作有关说发错误的是。（ ）

A: MyBatis的删除操作在映射文件中是通过配置<delete>元素来实现的。

B: MyBatis的删除操作也需要进行事务提交。

C: MyBatis的删除操作执行了SqlSession的delete()方法。

D: MyBatis的删除操作和添加操作时，也需要封装整个实体类。

1. MyBatis的更新操作有关说发错误的是。（ ）

A: MyBatis的更新操作在映射文件中是通过配置<update>元素来实现的 。

B: MyBatis的更新操作也需要进行事务提交。

C: MyBatis的更新操作执行了SqlSession的update()方法 。

D: MyBatis的更新操作和添加操作时，只需要将insert()方法改为updat()方法即可。

1. 关于MyBatis模糊查询中进行SQL字符串拼接时，说法错误的是。（ ）

A: 使用“${}”进行SQL字符串拼接时，无法防止SQL注入问题。

B: 可以使用MySQL中的concat()函数进行字符串拼接。

C: 使用MySQL中的concat()函数进行字符串拼接，也无法防止SQL注入。

D: 使用MySQL中的concat()函数进行字符串拼接，导致数据库移植性变差。

1. 有关MyBatis工作原理说法错误的是。（ ）

A: MyBatis的全局配置文件配置了MyBatis的运行环境等信息，其中主要内容是获取数据库连接

B: MyBatis映射文件中配置了操作数据库的SQL语句，需要在MyBatis的全局配置文件中加载才能执行

C: 可以通过MyBatis的环境等配置信息构建会话对象SqlSession

D: SqlSession对象，该对象中包含了执行SQL的所有方法

1. 使用ORM框架后，应用程序不再直接访问底层数据库，而是以（）的方式来操作持久化对象（PO，即Persisent Object）。（ ）

A: 面向业务B: Hibernate C: 面向对象 D: MyBatis

1. 简答题
2. 请简述Hibernate和MyBatis这两个框架的主要区别。

## 答案

1. 填空题
2. <delete>
3. namespace
4. concat()
5. 数据库连接
6. MyBatis
7. 判断题
8. 错
9. 错
10. 对
11. 对
12. 错
13. 选择题
14. D
15. D
16. C
17. C
18. C
19. 简答题

Hibernate是一个全表映射的框架。通常开发者只需定义好持久化对象到数据库表的映射关系，就可以通过Hibernate提供的方法完成持久层操作，开发者并不需要熟练的掌握SQL语句的编写，所以其开发效率会高于MyBatis。而Hibernate自身也存在着一些缺点，例如它在多表关联时，对SQL查询的支持较差；更新数据时，需要发送所有字段；不支持存储过程；不能通过优化SQL来优化性能等。这些问题导致其只适合在场景不太复杂且对性能要求不高的项目中使用。

MyBatis是一个半自动映射的框架。这里所谓的“半自动”是相对于Hibernate全表映射而言的。与Hibernate相比，虽然使用MyBatis手动编写SQL要比使用Hibernate的工作量大，但MyBatis可以配置动态SQL并优化SQL，可以通过配置决定SQL的映射规则，它还支持存储过程等。对于一些复杂的和需要优化性能的项目来说，显然使用MyBatis更加合适。

# MyBatis的核心配置

## 习题

1. 填空题
2. 当数据表中的列和需要返回的对象的属性不完全一致， MyBatis是不会自动赋值的。此时，就可以使用【 】元素进行处理。
3. <resultMap>元素的子元素<id>用于表示哪个列是【 】，而【 】元素用于表示POJO和数据表中普通列的映射关系。
4. MyBaits映射文件中使用<include>元素的【 】属性可以引用自定义的代码片段，其属性值为自定义代码片段的id。
5. MyBatis映射文件中<insert>的<selectKey>子元素的order属性可以被设置为【 】或【 】。
6. 对于数据源的配置，MyBatis框架提供了UNPOOLED、【 】和JNDI三种数据源类型。
7. 判断题
8. <resultMap>的子元素<association>和<collection>用于处理多表时的关联关系。（ ）
9. MyBatis映射文件的<insert>元素的属性与<select>元素的属性大部分相同，但还包含了3个特有属性。（ ）
10. MyBatis框架提供了许多默认别名，由于别名不区分大小写，所以在使用时要注意重复定义的覆盖问题。（ ）
11. 映射文件中可以包含多个id值相同的<resultMap>元素。（ ）
12. 与<insert>元素一样，<update>和<delete>元素在执行完之后，也会返回一个表示影响记录条数的整数。（ ）
13. 选择题
14. 以下有关<sql>元素说法错误的是。（ ）

A: <sql>元素的作用就是定义可重用的SQL代码片段，然后在其他语句中引用这一代码片段。

B: 使用<include>元素的refid属性可以引用自定义的代码片段。

C: 使用<include>元素refid的属性值为自定义代码片段的name。

D: <sql>元素是<mapper>元素的子元素。

1. 以下有关MyBatis映射文件中<insert>元素说法正确的是。（ ）

A: <insert>元素用于映射插入语句，在执行完元素中定义的SQL语句后，没有返回结果。

B: <insert>元素的属性与<select>元素的属性相同。

C: keyColumn属性用于设置第几列是主键，当主键列不是表中的第一列时需要设置。

D: useGeneratedKeys（仅对insert有用）此属性会使MyBatis使用JDBC的getGeneratedKeys()方法来获取由数据库内部生产的主键。

1. 以下关于<select>元素及其属性说法错误的是。（ ）

A: <select>元素用来映射查询语句，它可以帮助我们从数据库中读取出数据，并组装数据给业务开发人员。

B: parameterType属性表示传入SQL语句的参数类的全限定名或者别名。

C: resultMap表示外部resultMap的命名引用，返回时可以同时使用resultType和resultMap

D: 在同一个映射文件中可以配置多个<select>元素。

1. 关于<typeHandlers>元素说法错误的是。（ ）

A: <typeHandler>元素就是用来在配置文件中注册自定义的类型处理器的，它的使用方式有两种。

B: 通过<typeHandlers>元素的子元素<typeHandler>就可以配置单个类型处理器。

C: 通过<typeHandlers>元素的子元素<package>可以配置包下的所有类型处理器。

D: <typeHandlers>元素的子元素<package>的handler属性用于指定类型处理器所在的包名，系统会在启动时自动的扫描com.itheima.type包下所有的文件，并把它们作为类型处理器。

1. 有关MyBatis配置文件中<settings>元素的说法错误的是。（ ）

A: <settings>元素主要用于改变MyBatis运行时的行为，例如开启二级缓存、开启延迟加载等。

B: 虽然不配置<settings>元素，也可以正常运行MyBatis，但是熟悉<settings>的配置内容以及它们的作用还是十分必要的。

C: <settings>元素中延迟加载的全局开关的参数lazyLoadingEnabled默认开启。

D: <settings>元素的配置内容大多数都不需要开发人员去配置它，通常在需要时只配置少数几项即可。

1. 简答题
2. 请简述<sql>元素的作用及使用。

## 答案

1. 填空题
2. <resultMap>
3. 主键 、 <result>
4. refid
5. BEFORE 、AFTER
6. POOLED
7. 判断题
8. 对
9. 对
10. 对
11. 错
12. 对
13. 选择题
14. C
15. C
16. C
17. D
18. C
19. 简答题

<sql>元素的作用就是定义可重用的SQL代码片段。使用时，先通过<sql>元素定义重用的代码片段，然后在需要使用的地方使用<include>元素的refid属性引用了自定义的代码片段。

# 动态SQL

## 习题

1. 填空题
2. 当在MyBaits文件中使用了<bind>元素，在SQL语句中可以直接引用<bind>元素的【 】属性值即可进行动态SQL组装。
3. 在使用MyBaits的<foreach>时，如果传入的参数是多个的时候，就需要把它们封装成一个Map了，这时候collection属性值就为Map的【 】。
4. 在MyBaits映射文件中使用<set>和<if>元素组合进行update语句动态SQL组装时，如果<set>元素内包含的内容都为空，则会出现【 】。
5. MyBaits中<trim>元素的作用是去除一些特殊的字符串，它的【 】属性代表的是语句的前缀，而prefixOverrides属性代表的是需要去除的哪些特殊字符串。
6. 使用<set>和<if>元素相结合的方式来组装update语句时，<set>元素就会动态前置【 】关键字，同时也会消除SQL语句中最后一个多余的【 】。
7. 判断题
8. 可以将任何可迭代对象（如列表、集合等）和任何的字典或者数组对象传递给<foreach>作为集合参数。（ ）
9. 在MyBaits映射文件中使用<set>和<if>元素组合进行update语句动态SQL组装时，如果<set>元素内包含的内容都为空，则会出现SQL语法错误。（ ）
10. MyBaits中对原始SQL中的“where 1=1”的条件可以使用<where>与<if>元素组合进行替换。（ ）
11. 在MyBatis中<if>元素用于单条件分支判断，当某个<if>的test属性判断成立，然后跳过其他<if>元素进行动态SQL组装。（ ）
12. MyBatis中的<foreach>元素可以用于对数组和集合循环遍历，批量执行SQL操作。（ ）
13. 选择题
14. 以下不属于<foreach>元素中使用的属性的是。（ ）

A: separator

B: collection

C: current

D: item

1. 以下关于<foreach>元素中使用的几种属性的描述错误的是。（ ）

A: item：配置的是循环中当前的元素。

B: index：配置的是当前元素在集合的位置下标。

C: collection：配置的是传递过来的参数类型，它可以是一个array、list（或collection）、Map集合的键、POJO包装类中数组或集合类型的属性名等。

D: separator：配置的是各个元素的间隔符。

1. 以下关于MyBatis的<set>元素的使用及说法正确的是。（ ）

A: <set>元素主要用于更新操作，其主要作用是在动态包含的SQL语句前输出一个SET关键字，并将SQL语句中最后一个多余的逗号去除。

B: 使用MyBatis的<set>元素来更新操作时，前端需要传入所有参数字段，否则未传入字段会默认设置为空。

C: 在映射文件中使用<set>和<if>元素组合进行update语句动态SQL组装时， <set>元素内包含的内容可以都为空，<if>元素会进行判断处理。

D: 在映射文件进行更新操作时，只需要使用<set>元素就可以进行动态SQL组装。

1. 以下有关MyBatis动态SQL中的主要元素说法错误的是。（ ）

A: <if>用于单条件分支判断。

B: <choose>（<when>、<otherwise>）用于多条件分支判断。

C: <foreach>循环语句，常用于in语句等列举条件中。

D: <bind>从OGNL表达式中创建一个变量，并将其绑定到上下文，只于模糊查询的sql中。

1. 当有多个选择的情况是，值需要从多个选项中选择一个去执行时，可以使用的动态SQL元素是。（ ）

A: <if>

B: <choose>、<when>、<otherwise>

C: <when>

D: <set>

1. 简答题
2. 请简述MyBatis动态SQL中的主要元素及说明。

## 答案

1. 填空题
2. name
3. 键
4. SQL语法错误
5. prefix
6. SET 、逗号
7. 判断题
8. 对
9. 对
10. 对
11. 错
12. 对
13. 选择题
14. C
15. C
16. A
17. D
18. B
19. 简答题

MyBatis动态SQL中的主要元素及说明如下：​

<if>：判断语句，用于单条件分支判断；

<choose>（<when>、<otherwise>）：相当于Java中的switch...case...default语句，用于多条件分支判断；

<where>、<trim>、<set>：辅助元素，用于处理一些SQL拼装、特殊字符问题；

<foreach>：循环语句，常用于in语句等列举条件中；

<bind>：从OGNL表达式中创建一个变量，并将其绑定到上下文，常用于模糊查询的sql中。

# MyBatis的关联映射

## 习题

1. 填空题
2. MyBatis是通过<resultMap>元素的【 】子元素该元素来处理多对多关联关系的。
3. MyBatis是通过<resultMap>元素的【 】子元素该元素来处理一对一关联关系的。
4. 在关系型数据库中，多对多关系会产生【 】表，引入两张表的主键作为外键，两个主键成为联合主键或使用新的字段作为主键。
5. 在关系型数据库中，一对一可以在【 】引入对方主键作为外键。
6. <collection>子元素的ofType属性与javaType属性对应，它用于指定实体对象中集合类属性所包含的【 】。
7. 判断题
8. MyBatis是同样是通过<resultMap>元素的<collection>子元素该元素来处理多对多关联关系的。（ ）
9. MyBatis在映射文件中加载关联关系对象主要通过两种方式：嵌套查询和嵌套结果。（ ）
10. 在关系型数据库中，一对多就是在“一”的一方，添加“多”的一方的主键作为外键。（ ）
11. MyBatis中<collection>元素的属性与<association>元素相同，且使用也很简单，同样有嵌套查询和嵌套结果两种关联方式。（ ）
12. MyBatis是通过<resultMap>元素的<association>子元素该元素来处理一对一关联关系的。（ ）
13. 选择题
14. 以下关于MyBatis映射文件中<association>元素属性的说明错误的是。（ ）

A: property：指定映射到的实体类对象属性，与表字段一一对应。

B: column：指定表中对应的字段。

C: javaType：指定映射到实体对象属性的类型。

D: fetchType：指定在关联查询时是否启用延迟加载。fetchType属性有lazy和eager两个属性值，默认值为eager。

1. 下面关于数据库中多表之间关联关系说法错误的是。（ ）

A: 一对一关联关系可以在任意一方引入对方主键作为外键。

B: 一对多关联关系在“一”的一方，添加“多”的一方的主键作为外键。

C: 多对多关联关系会产生中间关系表，引入两张表的主键作为外键。

D: 多对多关联关系的两个表的主键成可以为联合主键或使用新的字段作为主键。

1. 下面关于Java对象之间的关联关系描述正确的是。（ ）

A: 一对一的关系就是在本类和对方类中定义同一个类型的对象。

B: 一对多的关系就是一个A类类型对应多个B类类型的情况。

C: 多对多的关系只需要在一方的类中引入另一方类型的集合。

D: 多对多关联关系需要在本类中引入本类的集合。

1. 下面属性中不属于<association>元素属性的是。（ ）

A: property

B: column

C: ofType

D: javaType

1. 下面关于<collection>元素的描述正确的是。（ ）

A: MyBatis就是通过<collection>元素来处理一对多关联关系的。

B: <collection>元素的属性与<association>元素完全相同。

C: ofType属性与javaType属性对应，它用于指定实体对象中所有属性所包含的元素类型。

D: <collection >元素只能使用嵌套查询方式。

1. 简答题
2. 请简要说明MyBatis在映射文件中加载关联关系对象的方式。

## 答案

1. 填空题
2. <collection>
3. <association>
4. 中间关系
5. 任意一方
6. 元素类型
7. 判断题
8. 对
9. 对
10. 错
11. 错
12. 对
13. 选择题
14. D
15. B
16. B
17. C
18. A
19. 简答题

MyBatis在映射文件中加载关联关系对象主要通过两种方式：嵌套查询和嵌套结果。嵌套查询是指通过执行另外一条SQL映射语句来返回预期的复杂类型；嵌套结果是使用嵌套结果映射来处理重复的联合结果的子集。

# MyBatis与Spring的整合

## 习题

1. 填空题
2. 在MyBatis+Spring的项目中，事务是由【 】来管理的。
3. MyBatis-Spring团队提供了一种自动扫描的形式来配置MyBatis中的映射器——采用【 】类。
4. 【 】是MyBatis-Spring团队提供的一个用于根据Mapper接口生成Mapper对象的类。
5. SqlSessionDaoSupport是一个抽象支持类，可以通过SqlSessionDaoSupport类的【 】方法来获取所需的SqlSession。
6. 在进行Spring与MyBatis整合时，Spring框架所需要准备的JAR包共10个，其中包括：4个核心模块JAR，AOP开发使用的JAR，【 】和事务的JAR。
7. 判断题
8. MapperFactoryBean的参数SqlSessionTemplate用于指定SqlSessionTemplate。如果与SqlSessionFactory同时设定，则只会启用SqlSessionFactory。（ ）
9. MyBaits与Spring进行整合时，Dao层开发可以使用传统的DAO方式的开发整合，以及Mapper接口方式的开发整合。（ ）
10. 可以使用mybatis-spring包中所提供的SqlSessionTemplate类或SqlSessionDaoSupport类来实现向DAO实现类中注入SqlSessionFactory。（ ）
11. 在实际的项目开发中，Spring与MyBatis都是整合在一起使用的。（ ）
12. MapperFactoryBean是MyBatis-Spring团队提供的一个用于根据Mapper接口生成Mapper对象的类。（ ）
13. 选择题
14. MapperFactoryBean是MyBatis-Spring团队提供的用于根据Mapper接口生成Mapper对象的类，该类在Spring配置文件中可以配置的参数不包括。（ ）

A: mapperInterface

B: SqlSessionFactory

C: SqlSessionTemplate

D: basePackage

1. MapperScannerConfigurer类在Spring配置文件中使用时，可以配置的属性及说明错误的是。（ ）

A: basePackage：指定映射接口文件所在的包路径，当需要扫描多个包时可以使用分号或逗号作为分隔符。

B: annotationClass：指定了要扫描的注解名称，只有被注解标识的类才会被配置为映射器。

C: sqlSessionFactoryBeanName：指定在Spring中定义的SqlSessionFactory的Bean名称。

D: sqlSessionTemplateBeanName：指定在Spring中定义的SqlSessionTemplate的Bean名称。如果定义此属性，则sqlSessionFactoryBeanName将起作用

1. 在MyBatis+Spring的项目中，以下有关事务的相关说法正确的是。（ ）

A: 在MyBatis+Spring的项目中，事务是由MyBatis来管理的。

B: 在项目中，数据访问层既是处理业务的地方，又是管理数据库事务的地方。

C: 进行注解开发时，需要在配置文件中配置事务管理器并开启事务注解。

D: 进行注解开发时，需要使用@Transactional注解来标识表现层中的类。

1. 以下不属于MapperScannerConfigurer类，在Spring配置文件中使用时需要配置的属性的是。（ ）

A: basePackage

B: annotationClass

C: sqlSessionFactoryBeanName

D: mapperInterface

1. 以下有关采用传统DAO开发方式进行MyBatis与Spring框架的整合的说法错误的是。（ ）

A: 采用传统DAO开发方式进行MyBatis与Spring框架的整合时，只需要编写DAO接口。

B: 采用传统DAO开发方式进行MyBatis与Spring框架的整合时，需要向DAO实现类中注入SqlSessionFactory，然后在方法体内通过SqlSessionFactory创建SqlSession。

C: 可以使用mybatis-spring包中所提供的SqlSessionTemplate类或SqlSessionDaoSupport类来实现在类中注入SqlSessionFactory。

D: SqlSessionDaoSupport是一个抽象支持类，它继承了DaoSupport类，主要是作为DAO的基类来使用。可以通过SqlSessionDaoSupport类的getSqlSession()方法来获取所需的SqlSession。

1. 简答题
2. 请对mybatis-spring包中所提供的两个类SqlSessionTemplate和SqlSessionDaoSupport进简要介绍。

## 答案

1. 填空题
2. Spring
3. MapperScannerConfigurer
4. MapperFactoryBean
5. getSqlSession()
6. JDBC
7. 判断题
8. 错
9. 对
10. 对
11. 对
12. 对
13. 选择题
14. D
15. D
16. C
17. D
18. A
19. 简答题

SqlSessionTemplate是mybatis-spring的核心类，它负责管理MyBatis的SqlSession，调用MyBatis的SQL方法，当调用SQL方法时，SqlSessionTemplate将会保证使用的SqlSession和当前Spring的事务是相关的；SqlSessionDaoSupport是一个抽象支持类，它继承了DaoSupport类，主要是作为DAO的基类来使用。

# Spring MVC入门

## 习题

1. 填空题
2. 前端控制器拦截请求后，会调用【 】。
3. Spring MVC支持多种视图技术，包括【 】、Velocity和FreeMarker等。
4. Spring MVC的配置文件中，可以配置处理器映射、处理器映射器、处理器适配器和【 】。
5. Spring MVC提供了一个前端控制器【 】，使开发人员无需额外开发控制器对象。
6. 在Spring MVC的执行流程中，Controller执行完成后，会返回一个【 】对象。
7. 判断题
8. HandlerAdapter将ModelAndView对象返回给ViewReslover。（ ）
9. Spring4.3版本的配置文件中，必须要配置处理器映射器、处理器适配器和视图解析器，否则程序将无法运行。（ ）
10. <load-on-startup>元素中的1表示容器在启动时立即加载这个Servlet。（ ）
11. Spring MVC支持JSP、Velocity、XML和FreeMarker等视图技术。（ ）
12. Spring MVC的灵活性比较弱，易于与其他框架集成。（ ）
13. 选择题
14. 下面关于Spring MVC特点说法错误的是。（ ）

A: 灵活性强，但不易于与其他框架集成

B: 可自动绑定用户输入，并能正确的转换数据类型

C: 支持国际化

D: 使用基于XML的配置文件，在编辑后，不需要重新编译应用程序

1. Spring MVC中的后端控制器是指。（ ）

A: HandlerAdapter

B: DispatcherServlet

C: ViewReslover

D: Handler

1. 用户通过浏览器向服务器发送请求时，负责拦截用户请求的是。（ ）

A: 处理器

B: 处理器映射器

C: 处理器适配器

D: 前端控制器

1. 入门程序中，不是必须引入的JAR包是。（ ）

A: Spring的4个核心JAR包

B: commons-logging的JAR包

C: spring-web和spring-webmvc的JAR包

D: log4j的JAR

1. Spring MVC是Spring提供的一个实现了（ ）设计模式的轻量级Web框架。

A: Web MVC

B: Web

C: 单例

D: 工厂

1. 简答题
2. 请简述Spring MVC的特点。

## 答案

1. 填空题
2. HandlerMapping
3. JSP
4. 视图解析器
5. DispatcherServlet
6. ModelAndView
7. 判断题
8. 错
9. 错
10. 对
11. 错
12. 错
13. 选择题
14. A
15. D
16. D
17. D
18. A
19. 简答题

1.是Spring框架的一部分，可以方便的利用Spring所提供的其他功能。

2.灵活性强，易于与其他框架集成。

3.提供了一个前端控制器DispatcherServlet，使开发人员无需额外开发控制器对象。

4.可自动绑定用户输入，并能正确的转换数据类型。

5.内置了常见的校验器，可以校验用户输入。如果校验不能通过，那么就会重定向到输入表单。

6.支持国际化。可以根据用户区域显示多国语言。

# Spring MVC的核心类和注解

## 习题

1. 填空题
2. RequestMapping注解类型用于映射【 】。
3. 在使用Spring MVC的注解开发时，除了需要引入Spring的核心JAR包、Commons-logging的JAR包以及Spring MVC的2个JAR包外，还需要引入【 】的JAR包。
4. 在视图解析器配置中，可以设置视图的【 】。
5. 如果没有通过<init-param>元素配置，则应用程序会默认去【 】寻找配置文件。
6. 如果<load-on-startup>元素不存在，则应用程序会【 】加载该Servlet。
7. 判断题
8. 在控制器类中，每一个请求处理方法都可以有多个不同类型的参数，以及一个多种类型的返回结果。（ ）
9. @RequestMapping的method属性必须使用。（ ）
10. @RequestMapping(method = RequestMethod.GET)可以缩写为@GetMapping。（ ）
11. @RequestMapping注解的属性都是可选属性。（ ）
12. Web.xml文件中必须使用<load-on-startup>元素和<init-param>元素，否则文件会报错。（ ）
13. 选择题
14. 下面关于请求处理方法返回类型说法错误的是。（ ）

A: 常见的返回类型是ModelAndView、String和void

B: ModelAndView类型中可以添加Model数据，并指定视图

C: String类型的返回值可以跳转视图，但不能携带数据

D: void类型主要在异步请求时使用，它既返回数据，又跳转视图

1. 下面类型中，不属于请求处理方法参数类型的是。（ ）

A: javax.servlet.http.HttpSession

B: @MatrixVariable

C: org.springframework.ui.Model

D: void

1. 下面关于组合注解的说法正确的是。（ ）

A: 组合注解是Spring3.x版本中的新特性

B: 组合注解可以简化常用的HTTP方法的映射

C: 在所有的Spring项目中，使用组合注解可以替代@RequestMapping注解

D: @GetMapping注解可以用来匹配GET和POST方式的请求

1. 下面关于@RequestMapping注解说法错误的是。（ ）

A: @RequestMapping注解的默认属性是value

B: @RequestMapping注解的value属性值可以省略

C: @RequestMapping注解的value属性名可以省略

D: @RequestMapping注解的value属性必须标注

1. 下面关于<load-on-startup>元素说法错误的是。（ ）

A: 如果<load-on-startup>元素的值为1，则在应用程序启动时会立即加载该Servlet

B: 如果<load-on-startup>元素不存在，则应用程序会在第一个Servlet请求时加载该Servlet

C: 如果<load-on-startup>元素的值为1，则在应用程序启动时会延迟加载该Servlet

D: <load-on-startup>元素是可选的

1. 简答题
2. 请简述ModelAndView、String和void三种返回类型的作用。

## 答案

1. 填空题
2. 一个请求或一个方法
3. Spring AOP
4. 前缀和后缀
5. WEB-INF目录下
6. 在第一个Servlet请求时
7. 判断题
8. 对
9. 错
10. 对
11. 对
12. 错
13. 选择题
14. D
15. D
16. B
17. D
18. C
19. 简答题

ModelAndView类型中可以添加Model数据，并指定视图；String类型的返回值可以跳转视图，但不能携带数据；而void类型主要在异步请求时使用，它只返回数据，而不会跳转视图。

# 数据绑定

## 习题

1. 填空题
2. 在使用集合数据绑定时，后台方法中不支持直接使用集合形参进行数据绑定，所以需要使用包装POJO作为形参，然后在包装POJO中包装一个【 】属性。
3. 使用包装POJO类型数据绑定时，如果前端条件参数是包装类中POJO的子属性，则参数名必须为【 】。
4. 当前端请求的参数比较简单时，可以在后台方法的形参中直接使用Spring MVC提供的【 】类型进行数据绑定。
5. 如果将订单和用户的所有查询条件都封装在一个简单POJO中，显然会比较混乱，这时就可以考虑使用【 】类型的数据绑定。
6. 在使用POJO类型数据绑定时，前端请求的参数名必须与要绑定的POJO类中的【 】一样，这样才会自动将请求数据绑定到POJO对象中，否则后台接收的参数值为【 】。
7. 判断题
8. 绑定数组与绑定集合页面传递的参数相同，只是后台接收方法的参数不同。（ ）
9. 要使用集合数据绑定，需要在包装类中定义一个包含用户信息类的集合，然后在接收方法中将参数类型定义为该包装类的集合。（ ）
10. 在执行绑定数组操作中，前台页面中复选框中传递的name属性值必须相同。（ ）
11. 注册自定义的Formatter转换器类时，Bean的类名必须是org.springframework.format.support.FormattingConversionServiceFactoryBean，并且其属性为formatters。（ ）
12. Formatter与Converter的作用相同，只是Formatter的源类型可以是任意类型，而Converter必须是一个String类型。（ ）
13. 选择题
14. 以下有关Spring MVC数据绑定中集合数据绑定的说法正确的是。（ ）

A:批量删除用户操作时，前端请求传递过来的参数就会包含多个相同类型的数据，此时可以采用数组类型数据绑定的形式

B:使用集合数据绑定需要后台方法中定义一个集合类型参数介绍绑定前端请求参数

C:绑定数组与绑定集合页面传递的参数相同，只是后台接收方法的参数不同

D:在使用集合数据绑定时，后台方法中不支持直接使用集合形参进行数据绑定

1. 下面选项中，哪一个是Spring的编码过滤器类。（ ）

A: org.springframework.web.filter.EncodingFilter

B: org.springframework.web.filter.CharacterEncodingFilter

C: org.springframework.web.filter.CharacterEncoding

D: org.springframework.web.filter.CharacterFilter

1. @RequestParam注解中可以省略属性名称的是。（ ）

A: defaultValue

B: value

C: name

D: required

1. 下面关于包装POJO类型数据绑定的说法正确的是。（ ）

A: 如果查询条件参数是包装类的直接基本属性，则参数名直接用对应的属性名

B: 如果查询条件参数是包装类的直接基本属性，则参数名必须使用对应的“对象.属性名”

C: 如果查询条件参数是包装类中POJO的子属性，则参数名必须为属性名

D: 如果查询条件参数是包装类中POJO的子属性，则参数名必须为“对象.子属性.属性值”的形式

1. 下面不属于@RequestParam注解类型属性的是。（ ）

A: id

B: value

C: name

D: required

1. 简答题
2. 为什么需要自定义数据绑定？

## 答案

1. 填空题
2. 集合
3. 对象.属性
4. 默认参数
5. 包装POJO
6. 属性名 null
7. 判断题
8. 错
9. 对
10. 对
11. 对
12. 错
13. 选择题
14. D
15. B
16. B
17. A
18. A
19. 简答题

一般情况下，使用基本数据类型和POJO类型的参数数据已经能够满足需求，然而有些特殊类型的参数是无法在后台进行直接转换的，例如日期数据就需要开发者自定义转换器（Converter）或格式化（Formatter）来进行数据绑定。

# JSON数据绑定和RESTful支持

## 习题

1. 填空题
2. RESTful风格在HTTP请求中，使用【 】、delete、post和get方式分别对应添加、删除、修改和查询的操作。
3. 使用<bean>标签配置方式配置JSON转换器时，需要同时配置处理器映射器和处理器适配器，并且JSON转换器是配置在【 】中。
4. JSON对象的数组结构以“[”开始，以“]”结束，中间部分由【 】以英文“,”分隔的值的列表组成。
5. @PathVariable注解用于接收并绑定请求参数，它可以将请求URL中的【 】到方法的形参上。
6. RESTful风格在HTTP请求中，不能够出现【 】。
7. 判断题
8. @PathVariable("id")注解用于接收并绑定请求参数，如果请求路径中的参数名方法形参名称一样，则@PathVariable后面的“(“id”)”可以省略。（ ）
9. JSON对象数据结构的关键字（key）必须为String类型，值（value）可以是String、Number、Object、Array等数据类型。（ ）
10. RESTful风格在HTTP请求中，使用put、delete、post和get方式分别对应添加、删除、修改和查询的操作。（ ）
11. RESTful也称之为REST，是英文“Representational State Transfer”的简称。（ ）
12. MappingJackson2HttpMessageConverter类只能将Java对象转换为JSON对象和XML文档，不能将JSON对象和XML文档转换为Java对象。（ ）
13. 选择题
14. 针对GET方式RESTful风格的请求http://localhost:8080/chapter14/user/1，后台方法接收参数的路径映射写法可行的是。（ ）

A: @RequestMapping(&quot;/user/{id}&quot;，method=RequestMethod.GET)。

B: @RequestMapping(value="/user/ids")。

C: @GetMapping(value="/user/{ids}")。

D: @GetMapping(value="/user/{id}"，method=RequestMethod.GET)

1. 下面属于RESTful风格请求的是。（ ）

A: http://.../queryItems?id=1

B: http://.../queryItems?id=1&amp;name=zhangsan

C: http://.../items/1

D: http://.../queryitems/1

1. 下面不属于AJAX中的属性的是。（ ）

A: data

B: contentType

C: dataType

D: database

1. 下面选项中不属于Jackson的开源包的是。（ ）

A: jackson-annoations-2.8.8.jar

B: jackson-core-2.8.8.jar

C: jackson-databind-2.8.8.jar

D: jackson-data-2.8.8.jar

1. JSON对象结构中，关键字key必须为。（ ）类型

A: Object

B: Array

C: String

D: Number

1. 简答题
2. 请简述Spring中HttpMessageConverter<T>接口的主要作用。

## 答案

1. 填空题
2. put
3. 适配器
4. 0个或多个
5. 变量映射
6. 动词
7. 判断题
8. 对
9. 对
10. 对
11. 对
12. 错
13. 选择题
14. C
15. C
16. D
17. D
18. C
19. 简答题

1.为了实现浏览器与控制器类（Controller）之间的数据交互，Spring提供了一个HttpMessageConverter<T>接口来完成此项工作。该接口主要用于将请求信息中的数据转换为一个类型为T的对象，并将类型为T的对象绑定到请求方法的参数中，或者将对象转换为响应信息传递给浏览器显示。

# 拦截器

## 习题

1. 填空题
2. Spring MVC单个拦截器执行顺序中，在【 】处理完请求后，才会执行afterCompletion()方法。
3. Spring MVC中的拦截器（Interceptor）类似于Servlet中的【 】，它主要用于拦截用户请求并作相应的处理。
4. 用于如果没有登录系统而直接访问主页面，拦截器会将请求拦截，并转发到【 】。
5. 当有多个拦截器同时工作时，它们的【 】方法会按照配置文件中拦截器的配置顺序执行。
6. 如果拦截器类中的preHandle()方法的返回值为【 】，则程序会继续向下执行处理器中的方法。
7. 判断题
8. 在实现用户登录权限验证中，需要定义一个登录拦截器，并在拦截器的postHandle()方法中编写业务逻辑进行登录控制。（ ）
9. 配置拦截器时，必须将path的属性值设置为/\*\*。（ ）
10. 多个拦截器时，postHandle()方法和afterCompletion()方法则会按照配置顺序执行。（ ）
11. 全局拦截器和指定路径下的拦截器不能够同时配置，否则运行时会报错。（ ）
12. <mvc:interceptor>中的子元素可以按照任意位置编写。（ ）
13. 选择题
14. 以下有关Spring MVC配置文件中拦截器的配置说法错误的是。（ ）

A: 要使用Spring MVC中拦截器，要先自定义拦截器还需要在配置文件中进行配置。

B: <mvc:interceptors>元素用于配置一组拦截器，其子元素<bean>中定义的是指定路径的拦截器。

C: <mvc:interceptors>元素中可以同时配置多个<mvc:interceptor>子元素。

D: <mvc:exclude-mapping>元素用于配置不需要拦截的路径请求。

1. 下面不属于拦截器类中的方法的是。（ ）

A: preHandler()

B: postHandle()

C: afterCompletion()

D: afterpletion()

1. 以下哪个方法可以定义Spring MVC中的拦截器。（ ）

A: 继承HandlerInterceptor

B: 实现WebRequestInterceptor

C: 实现HandlerInterceptorAdapter

D: 继承WebRequestInterceptor

1. 下列关于拦截器的执行流程说法错误的是。（ ）

A：程序首先会执行拦截器类中的preHandle()方法。

B：如果preHandle()方法的返回值为true，则程序会继续向下执行处理器中的方法，否则将不再向下执行。

C：在业务处理器（即控制器Controller类）处理完请求后，会执行preHandle()方法。

D：在DispatcherServlet处理完请求后，才会执行afterCompletion()方法。

1. 关于用户权限验证的执行流程，说法错误的是。（ ）

A：只有登录后的用户才能访问系统中的主页面。

B：如果没有登录系统而直接访问主页面，则拦截器会将请求拦截，并转发到登录页面。

C：如果用户名或密码错误，会在登录页面给出相应的提示信息。

D：当已登录的用户在系统主页中单击“退出”链接时，系统会回到主页面。

1. 简答题
2. 请简述拦截器HandlerInterceptor接口中的方法并做说明。

## 答案

1. 填空题
2. DispatcherServlet
3. 过滤器或Filter
4. 登录页面
5. preHandle()
6. true
7. 判断题
8. 错
9. 错
10. 错
11. 错
12. 错
13. 选择题
14. B
15. D
16. D
17. C
18. D
19. 简答题

HandlerInterceptor接口中有3个方法：

1、preHandler()方法：该方法会在控制器方法前执行，其返回值表示是否中断后续操作。当其返回值为true时，表示继续向下执行；当其返回值为false时，会中断后续的所有操作。

2、postHandle()方法：该方法会在控制器方法调用之后，且解析视图之前执行。可以通过此方法对请求域中的模型和视图做出进一步的修改。

3、afterCompletion()方法：该方法会在整个请求完成，即视图渲染结束之后执行。可以通过此方法实现一些资源清理、记录日志信息等工作。

# 文件上传和下载

## 习题

1. 填空题
2. 由于不同版本的IE浏览器，请求代理【 】中的关键字也略有不同。
3. ResponseEntity对象的作用有些类似于@ResponseBody注解，它用于直接返回【 】。
4. 由于各个浏览器【 】的不同，就会出现不同的乱码以及解析异常问题。
5. 使用Servlet API中提供的URLEncoder类中的【 】方法将中文转为UTF-8编码。
6. HttpStatus类型代表的是Http协议中的状态，示例中的HttpStatus.OK表示【 】，即服务器已成功处理了请求。
7. 判断题
8. 上传文件时，必须要保证所上传的文件不重名，为此可以通过UUID等方式对上传的文件名称进行重命名。（ ）
9. 上传的文件，可以在项目目录中找到。（ ）
10. 通过maxUploadSize属性可以对上传文件缓存中的最大尺寸进行设置。（ ）
11. Spring MVC的文件上传是通过MultipartResolver对象实现的。（ ）
12. Spring MVC需要通过第三方组件来实现文件上传。（ ）
13. 选择题
14. 下面关于MultipartFile接口中说法错误的是。（ ）

A: getOriginalFilename()用于获取上传文件的初始化名。

B: getSize()用于获取上传文件的大小，单位是KB。

C: getInputStream()用于读取文件内容，返回一个InputStream流。

D: transferTo(File file)用于将上传文件保存到目标目录下。

1. 下面属于CommonsMultipartResolver属性的是。（ ）

A: getContentType

B: getInputStream

C: isEmpty

D: defaultEncoding

1. 下面关于文件上传表单说法错误的是。（ ）

A: form表单的method属性设置为post

B: form表单的method属性设置为get

C: form表单的enctype属性设置为multipart/form-data

D: 提供<input type="file" name="filename" />的文件上传输入框

1. 下面关于文件下载方法内容描述错误的是。（ ）

A：响应头信息中的MediaType代表的是Interner Media Type（即互联网媒体类型），也叫做MIME类型。

B：MediaType.APPLICATION\_OCTET\_STREAM的值为application/octet-stream，即表示以二进制流的形式下载数据。

C：HttpStatus类型代表的是Http协议中的状态。

D：HttpStatus.OK表示500，即服务器已成功处理了请求。

1. 下面不属于文件解析器类CommonsMultipartResolver属性的是。（ ）

A：maxUploadSize

B：maxInMemorySize

C：defaultEncoding

D：lazy

1. 简答题
2. 请简述文件上传时表单需要满足的3个条件。

## 答案

1. 填空题
2. User-Agent
3. 结果对象
4. 内部转码机制
5. encoder(String s, String enc)
6. 200
7. 判断题
8. 对
9. 错
10. 错
11. 对
12. 错
13. 选择题
14. B
15. D
16. B
17. D
18. D
19. 简答题

文件上传表单需要满足的3个条件如下：

1.form表单的method属性设置为post；

2.form表单的enctype属性设置为multipart/form-data；

3.提供<input type="file" name="filename" />的文件上传输入框。

# SSM框架整合

## 习题

1. 填空题
2. SSM框架整合主要是【 】的整合，以及【 】的整合。
3. 为了避免Spring配置文件中的信息过于臃肿，通常会将Spring配置文件中的信息按照【 】分散在多个配置文件中。
4. @Transactional注解主要是针对数据的【 】、【 】、【 】进行事务管理。
5. 在整合项目中，db.properties文件主要用于【 】。
6. Spring与MyBatis框架的整合时，可以通过Spring【 】，然后调用实例对象中的查询方法来执行MyBatis映射文件中的【 】，如果能够正确查询出数据库中的数据，就可以认为Spring与MyBatis框架整合成功。
7. 判断题
8. 在实际开发时，为了避免Spring配置文件中的信息过于臃肿，通常会将Spring配置文件中的信息按照不同的功能分散在多个配置文件中。（ ）
9. @Autowired注解需要标注在Service层的实现类上，这样才能实现依赖注入。（ ）
10. @Transactional注解主要是针对数据的增加、修改、删除和查询进行事务管理。（ ）
11. Spring与Spring MVC，Spring MVC与MyBatis需要相互整合。（ ）
12. 在Spring MVC的配置文件中，视图解析器是必须配置的。（ ）
13. 选择题
14. 下列选项中，不需要配置在web.xml中的是。（ ）

A: Spring的监听器

B: 编码过滤器

C: 视图解析器

D: 前端控制器

1. 下列选项中，属于Spring MVC所必须的JAR包的是。（ ）

A: spring-web-4.3.6.RELEASE.jar

B: spring-webmvc-portlet-4.3.6.RELEASE.jar

C: spring-webmvc-4.3.6.RELEASE-javadoc.jar

D: spring-websocket-4.3.6.RELEASE.jar

1. 下列选项中，不属于SSM整合时所需的JAR包的是。（ ）

A: spring-web-4.3.6.RELEASE.jar

B: spring-webmvc-4.3.6.RELEASE.jar

C: ant-1.9.6.jar

D: xwork-core-2.3.24.jar

1. 下列关于SSM框架的整合说法错误的是。（ ）

A：Spring MVC与Spring之间不存在整合的问题。

B：SSM框架的整合就涉及到Spring与MyBatis的整合。

C：SSM框架的整合就涉及到Spring MVC与MyBatis的整合。

D：SSM框架的整合就涉及到Spring MVC与Spring之间的整合。

1. 下面选项中，不属于整合SSM框架所编写的配置文件的是。（ ）

A：db.properties

B：applicationContext.xml

C：mybatis-config.xml

D：struts.xml

1. 简答题
2. 请简述SSM框架整合思路。

## 答案

1. 填空题
2. Spring与MyBatis、Spring MVC与MyBatis
3. 不同的功能
4. 增加、修改、删除
5. 配置数据库常量
6. 实例化Bean、 SQL语句
7. 判断题
8. 对
9. 错
10. 错
11. 错
12. 错
13. 选择题
14. C
15. A
16. D
17. D
18. D
19. 简答题

由于Spring MVC是Spring框架中的一个模块，所以Spring MVC与Spring之间不存在整合的问题，只要引入相应JAR包就可以直接使用。因此SSM框架的整合就只涉及到了Spring与MyBatis的整合，以及Spring MVC与MyBatis的整合。Spring与MyBatis框架的整合时，通过Spring实例化Bean，然后调用实例对象中的查询方法来执行MyBatis映射文件中的SQL语句的，如果能够正确查询出数据库中的数据，那么就可以认为Spring与MyBatis框架整合成功。加入Spring MVC后，如果可以通过前台页面来执行查询方法，并且查询出的数据能够在页面中正确显示，那么就可以认为三大框架整合成功。

# BOOT客户管理系统

## 习题

1. 填空题
2. 系统中主要实现了两大功能模块：【 】和【 】。
3. 【 】注解一般在异步获取数据时使用。
4. 在BOOT客户管理系统中，新建客户信息窗口是通过【 】的模态框代码实现的。
5. 查询操作通常可以分为按条件查询和【 】。
6. 在实际应用中，无论是企业级项目，还是互联网项目，使用最多的一定是【 】。
7. 判断题
8. 删除操作只需将所需删除客户的id信息传递给后台删除方法即可。（ ）
9. 修改操作时，修改信息窗口中的数据需要清空。（ ）
10. @Responsebody注解与@RequestMapping注解配合使用时，页面中将可以获取到方法的返回值。（ ）
11. 通常在系统中，查询出的数据都会分页显示。（ ）
12. 在设置事务的传播行为时，需要将与查询相关方法的read-only属性值设置为true。（ ）
13. 选择题
14. 拦截用户请求是在哪一层次中实现的。（ ）

A: 持久对象层

B: 数据访问层

C: 业务逻辑层

D: Web表现层

1. 下列选项中，不属于BOOT客户管理系统中功能的是。（ ）

A: 用户登录

B: 用户管理

C: 查询客户

D: 修改客户

1. 下面选项中，关于Web表现层说法错误的是。（ ）

A: 该层主要包括Spring MVC中的Controller类和JSP页面。

B: Controller类主要负责拦截用户请求。

C: Controller类会调用业务逻辑层中相应组件的业务逻辑方法来处理用户响应。

D: Controller类会将请求的结果返回给JSP页面。

1. 下列关于BOOT客户管理系统用户登录模块说法错误的是。（ ）

A：用户登录过程中首先要验证用户名和密码是否正确，如果正确，可以成功登录系统，系统会自动跳转到主页。

B：用户登录过程中首先要验证用户名和密码是否正确，如果错误，则在登录页面给出错误提示信息。

C：为了保证系统的安全性，用户登录时必须实现登录验证。

D：为了保证系统的稳定性，用户登录时必须实现登录验证。

1. 下列关于客户管理模块说法，正确的是。（ ）

A：查询操作通常可以分为按条件查询和查询所有。

B：添加客户操作时，新建客户信息窗口中需要回显出客户信息。

C：修改客户时，必须修改所有的客户信息才可以执行保存修改。

D：删除客户时，只是更改了数据表中数据的状态，并没有删除数据库中的数据。

1. 简答题
2. 请简述如何实现登录验证。

## 答案

1. 填空题
2. 用户登录模块，客户管理模块
3. @Responsebody
4. Bootstrap
5. 查询所有
6. 查询操作
7. 判断题
8. 对
9. 错
10. 对
11. 对
12. 对
13. 选择题
14. D
15. B
16. C
17. D
18. A
19. 简答题

要实现登录验证，首先需要创建一个登录拦截器类，该类中需要拦截除用户登录请求之外的所有请求，并对用户的登录状态进行判断，只有已登录的用户请求才能够继续执行，否则会将用户请求转发到系统登录页面，并给出登录提示。编写完登录拦截器类后，还需要在Spring MVC的配置文件中对该拦截器类进行配置，配置完成后，即可使用。