**考试题型：**

1、判断题（10道题，每小题1分，共计10分）

2、单选题（16道题，每小题1分，共计16分）

3、多选题（7道题，每小题2分，漏选得1分，错选、多选均不得分，共计14分）

4、问答题（1道题，每小题10分，共计10分）

5、编程题（6道题，每小题5~10分，共计50分）

**复习重点**

下列每个题表述并非全部正确，个别题存在误导信息，请自行判断。

1. Web应用开发流程

浏览器通过http协议发送请求到服务端，tomcat通过请求（网址解析）到指定位置找到项目目录，找到对应的servlet对象，根据请求的方式调用servlet对象的方法，然后由控制层 接收请求并对请求进行分析，根据分析结果把请求分发给业务层 处理，持久层 是与外界数据存储进行交互的封装。视图层 将返回数据渲染通过response返回给浏览器进行显示

2、静态网页的显示可以不依赖于Web服务器。（T）

3、表格布局比DIV+CSS布局的兼容性更好。（T）

4、表格布局比DIV+CSS布局更灵活多样。（F）

5、表单标记通过method属性设置后台的处理程序。（F）action

6、网页中表单的主要作用是发布信息。（F）数据采集

7、表单中的输入组件都是通过<input>标记定义的。（T）

8、表单中的密码框的输入回显字符只能是“\*”。（T）

9、表单中一组单选按钮的name属性要相同。（T）

10、表单中一组复选按钮的name属性不需要相同。（F）

11、表单中的列表默认只显示1个选项。（T）

12、表单中的列表只能选择1个选项。（F）支持多选multiple

13、通过CSS的应用可以同时更新多个网页的样式。（T）

14、CSS可以在网页中定义也可以在CSS文件中定义。（T）内嵌式，外部导入

15、表单关键元素和相关设置

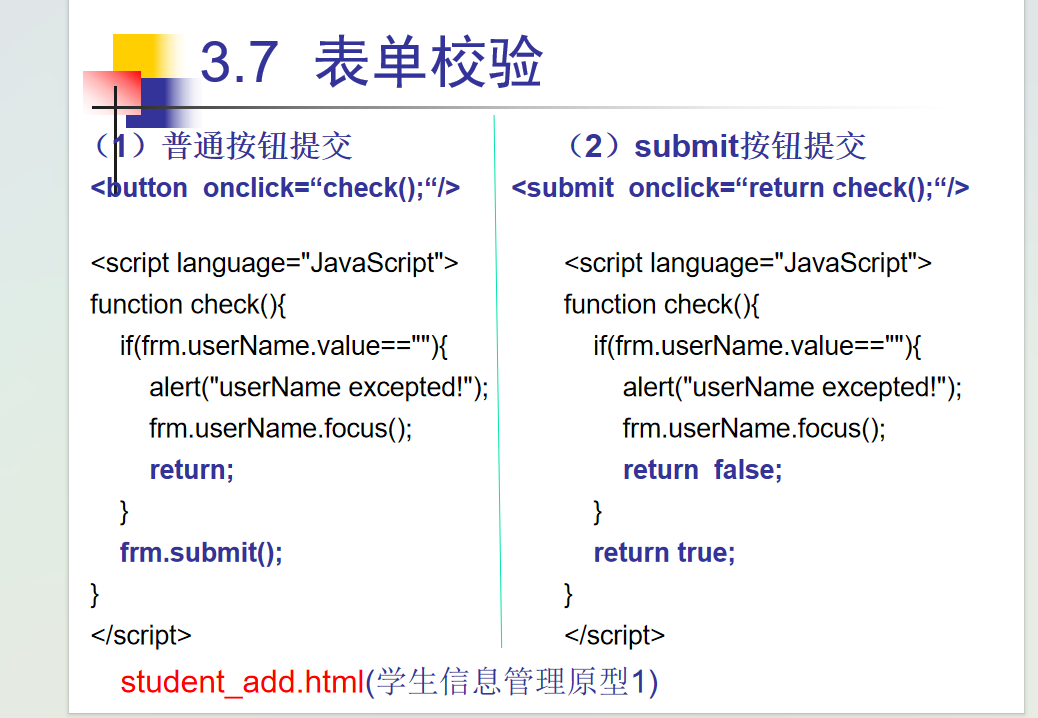
16、JavaScript变量名是区分大小写的（T）

17、JavaScript主要用于接收服务器端的响应。（F）用户交互

18、JavaScript变量的类型根据变量赋值来确定。（T）

19、JavaScript变量使用var 声明时必须赋初值。（F）

20、JavaScript实现表单提交参数验证



21、正则表达式



22、JSON是一种轻量级的数据交换格式。（T）

23、JSON和XML比较JSON体积更小。（T）

24、JSON数据前端显示（）

对象{“name”:”jack”,”age”:”18”,”属性名”:”属性值”}

数组[{“name”:”tom”,”address”:”江苏”,”email”:””},

{“name”:”tina”,”address”:”四川”,”email”:””}]

25、JSP开发中最常用的两种设计模式（jsp+Javabean；jsp+javabean+servlet）

26、JSP技术的MVC模式描述

M；①模型：一个或多个JavaBean对象，用于存储数据就。JavaBean主要提供简单的setXxx方法和getXxx方法，在这些方法中不涉及对数据的具体处理细节，以便增强模型的通用性。

V：②视图：一个或多个JSP页面，其作用是向控制器提交必要的数据和显示数据。JSP页面可以使用HTML标记、JavaBean标记以及Java程序片或Java表达式来显示数据。视图的主要工作就是显示数据，对数据的逻辑操作由控制器负责。 C：③控制器：一个或多个servlet对象，根据视图提交的要求进行数据处理操作，并将有关的结果存储到JavaBean中，然后servlet使用转发或重定向的方式请求视图中的某个JSP页面显示数据

27、JSP页面中的Java脚本标记包括哪三种元素：

（jsp表达式Expression；<%= 变量或可以返回值的方法或Java表达式 %>

声明标识Declaration；<%! 声明变量或方法的代码 %>

脚本标识Scriptlet：<% Java程序片段 %>

）

28、JSP页面中<%! %>标记的用途（声明变量或方法的代码）

<% %>标记 java程序片段

29、JSP内置对象（

Request：用来得到客户端的信息

response：处理服务器端对客户端的一些响应,

Out：把信息回送到客户端的浏览器,

Appliaction：用来保存网站的一些全局变量，存放所有的请求中数据

Session：用来保存单个用户访问时的信息，存放会话中的所有数据）

30、JSP过第一次访问过后，将生成一个exe文件（F）class文件

31、转发和重定向的区别（转发：request.sendRedirectDispatcher().forward()

重定向：response.sendRedirect().）

转发只能在当前应用跳转，重定向可以跳转到任何一个URL

转发地址栏不发生变化，重定向地址栏发生变化

转发会对request对象中的数据再次封装，重定向则是一个全新的请求

32、JavaBean的特点，一次编写（任何地方执行，任何地方重用）

33、根据用途JavaBean可以分类（通用工具Javabean类-数据转换，值Javabean类-封装，业务处理Javabean类-处理数据）

34、不可视化JavaBean的属性主要有哪些

简单，索引，绑定，限定

35、Javabean编写规范

（（1）实现java.io.Serializable接口；

（2）是一个公共类；

（3）类中必须存在一个无参数的构造函数；

（4）提供对应的setXxx()和getXxx()方法来存取类中的属性，方法中的“Xxx”为属性名称，属性的第一个字母应大写。若属性为布尔类型，则可使用isXxx()方法代替getXxx()方法。

）

36、JDBC访问数据库的过程

JDBC数据库管理操作的基本步骤

1、加载驱动程序

2、建立数据库连接（connection）

3、执行SQL语句（statement）

4、处理SQL语句执行结果（resultset）

5、关闭数据库连接

37、JDBC常用接口及其描述

DriverManager（驱动管理器获得数据库链接）

Connection（数据库链接接口）

Statement（语句接口，用来静态操作SQL语句）

PreparedStatement（预定义语句，用来动态操作SQL语句）

CallableStrtement（可以调用存储过程的预定义语句）

ResultSet（结果集，保存数据记录的结果集合）

ResultSetMetaData（结果集元素数据如：列名、列类型）

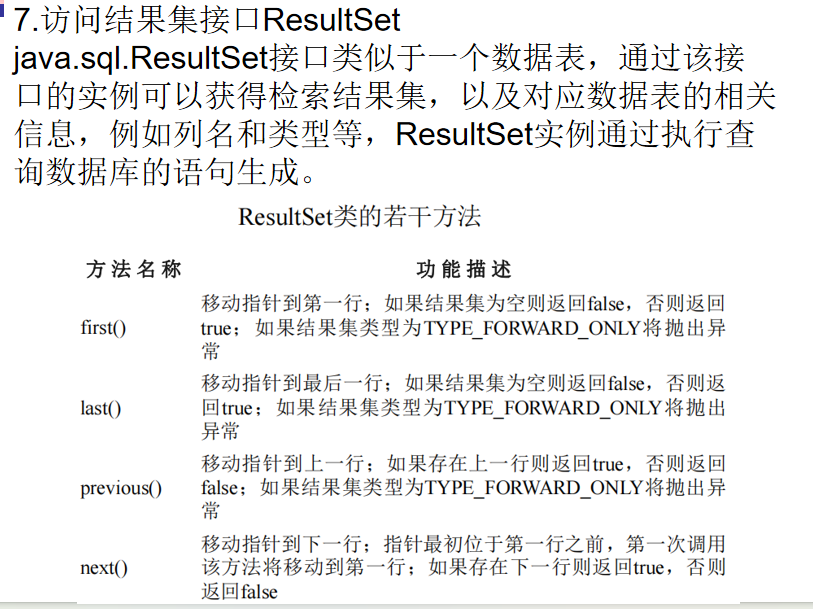
DatabaseMetaData（数据元数据：数据库名称、版本）

38、PreparedStatement接口的特点

java.sql.PreparedStatement接口继承自Statement， 是Statement接口的扩展，用来执行动态的SQL语句，即包含参数的SQL语句。

该接口扩展了带有参数SQL语句的执行操作，应用该接口中的SQL语句可以使用占位符“?”来代替其参数，然后通过setXxx()方法为SQL语句的参数赋值。

39、Resulset接口的



40、DBManage类的获得数据库连接、执行sql查询操作流程

41、业务逻辑Bean（获取数据库数据封装到值型JavaBean）流程

42、Servlet的生命周期中 init、service、doFilter、destroy

Servlet有良好的生存期的定义，包括加载和实例化、初始化、处理请求以及服务结束。这个生存期由Servlet接口的init,service和destroy方法表达。 Servlet被服务器实例化后，容器运行其init方法，请求到达时运行其service方法，service方法自动派遣运行与请求对应的doXXX方法（doGet，doPost）等，当服务器决定将实例销毁的时候调用其destroy方法

doFilter：过滤器Filter必须执行的方法

43、web.xml中servlet配置的XML标记

44、Ajax代码异步请求方式

45、服务器端servlet相应处理请求流程（获取表单参数，校验或提交，生成响应）

46、Ajax请求取列表形式的JSON数据

47、Spring MVC 是对 Servlet 的再封装（T）

48、Spring MVC的工作原理

（1）客户端（浏览器）发送请求，直接请求到 DispatcherServlet。

（2）DispatcherServlet 根据请求信息调用 HandlerMapping，解析请求对应的 Handler。

（3）解析到对应的 Handler后，开始由 HandlerAdapter 适配器处理。

（4）HandlerAdapter 会根据 Handler 来调用真正的处理器开处理请求，并处理相应的业务逻辑。

（5）处理器处理完业务后，会返回一个 ModelAndView 对象，Model 是返回的数据对象，View 是个逻辑上的 View。

（6）ViewResolver 会根据逻辑 View 查找实际的 View。

（7）DispaterServlet 把返回的 Model 传给 View（视图渲染）。

（8）把 View 返回给请求者（浏览器）

49、Spring 容器管理的 bean 默认是单实例的（T）

50、Spring 配置文件名是固定的（T）

51、Spring用<bean>标签定义对象时，<bean>标签中必须申明id（T）

52、Spring获取容器创建的对象的方法

53、Spring AOP的一个切入点可能横切多个业务组件。（T）

54、Spring容器实现了IOC和AOP机制，这些机制可以简化Bean对象创建和Bean对象之间的解耦。（T）

55、@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（F）类型装配，@resource是默认按照名称

56、Hibernate和MyBatis均是全自动映射的框架（F）

Hibernate是全自动的框架，是当前最流行的ORM框架之一，对于JDBC提供了比较完整的Hibernate的O/R Mapping实现了POJO和数据库表之间的映射，以及SQL的自动生成和执行。

Mybatis：半自动框架，也是现在非常流行的ORM框架之一，主要着力点在于POJO于SQL之间的映射关系。然后通过映射的配置文件，将sql所需要的参数以及返回的字段映射到指定的POJO。相对于hibernate的O/R而言，mybatis是一种Sql Mapping的ORM实现。

57、Hibernate和MyBatis都可以自动生成SQL（F） mybatis自己编写sql语句

58、应用在Web应用中的数据库系统有哪些

59、使用MyBatis框架，一般需要先建立与数据库表对应的实体类。（T）

60、MyBatis 和Hibernate是目前流行的ORM框架，前者因轻量而更加流行。（F）

Hibernate更流行

61、MyBatis 的程序里最终使用的对象是sql Session

62、MyBatis 工作流程或主要步骤。

63、MyBatis映射文件中的常用元素及其作用

64、MyBatis中动态SQL的常用元素及其作用

65、Spring中的通知按照在目标类方法的连接点位置，分为5种通知

66、Hibernate和MyBatis和JDBC各自特点和优势

67、JSP、Servlet、SpringMVC各自优势

68、Spring框架的优势和特点

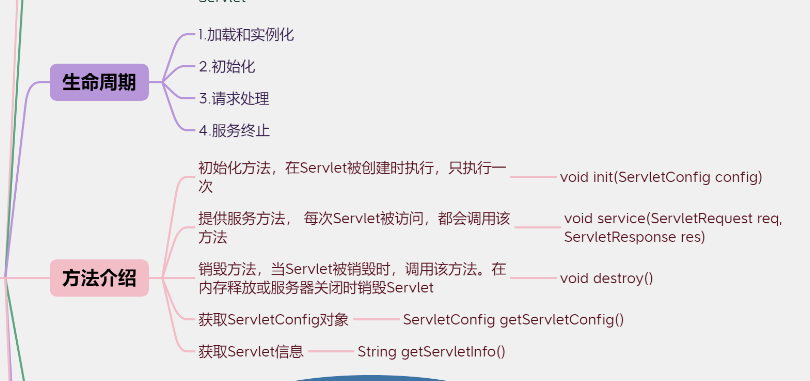


简答题

4.Spring中的通知按照在目标类方法的连接点位置，分为5种通知

环绕通知，前置通知，后置返回通知，后置（最终）通知，异常通知，引入通知

5.Servlet的生命周期中 init、service、doFilter、destroy

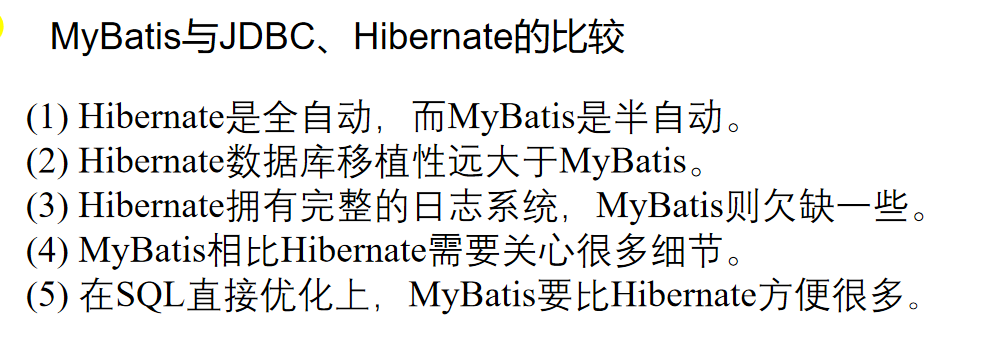


7.Hibernate和MyBatis和JDBC各自特点和优势

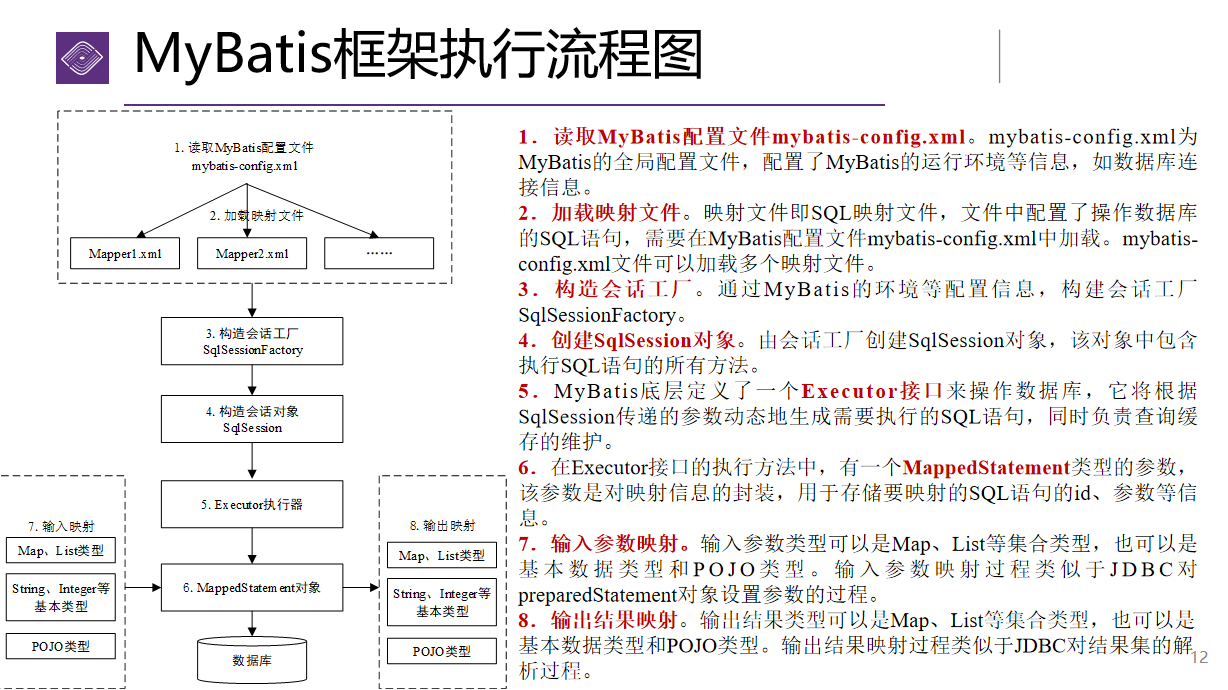








62、MyBatis 工作流程或主要步骤。



63、MyBatis映射文件中的常用元素及其作用



<properties>是一个配置属性的元素，该元素通常用来将内部的配置外在化，即通过外部的配置来动态的替换内部定义的属性。

<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。

<typeAliases>元素用于为配置文件中的Java类型设置一个简短的名字，即设置别名。别名的设置与XML配置相关，其使用的意义在于减少全限定类名的冗余。

typeHandler的作用就是将预处理语句中传入的参数从javaType（Java类型）转换为jdbcType（JDBC类型），或者从数据库取出结果时将jdbcType转换为javaType。

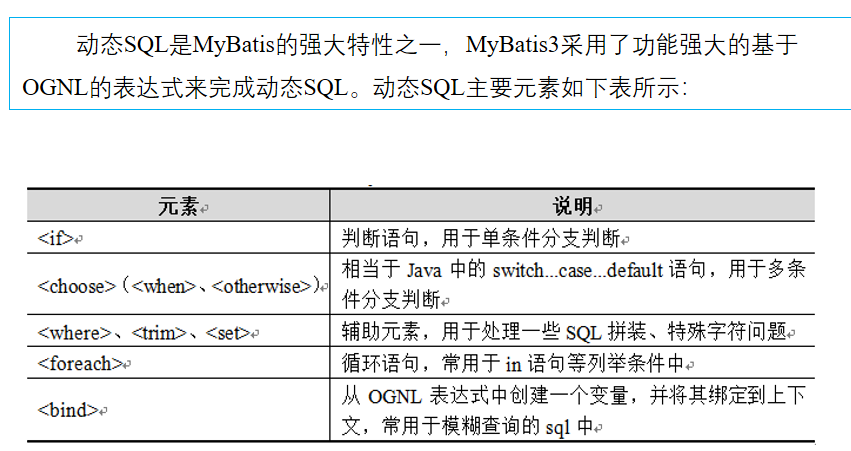
<typeHandler>元素可以在配置文件中注册自定义的类型处理器，它的使用方式有两种。

ObjectFactory的作用是实例化目标类，它既可以通过默认构造方法实例化，也可以在参数映射存在的时候通过参数构造方法来实例化。通常使用默认的ObjectFactory即可。

<plugins>元素的作用就是配置用户所开发的插件。

<environments>元素用于对环境进行配置。MyBatis的环境配置实际上就是数据源的配置，

MyBatis中动态SQL的常用元素及其作用



JSP、Servlet、SpringMVC各自优势

JSP:

与纯 Servlets相比：JSP可以很方便的编写或者修改HTML网页而不用去面对大量的println语句。

与JavaScript相比：虽然JavaScript可以在客户端动态生成HTML，但是很难与服务器交互，因此不能提供复杂的服务，比如访问数据库和图像处理等等。

与静态HTML相比：静态HTML不包含动态信息

Sevlet：

方便：Servlet提供了大量的实用工具例程，如处理很难完成的HTML表单数据、读取和设置HTTP头，以及处理Cookie和跟踪会话等。

跨平台：Servlet用Java类编写，可以在不同操作系统平台和不同应用服务器平台下运行。

灵活性和可扩展性：采用Servlet开发的Web应用程序，由于Java类的继承性及构造函数等特点，使得应用灵活，可随意扩展。

除了上述几点外，Servlet还具有功能强大、能够在各个程序之间共享数据、安全性强等特点。

SpringMVC：

（1）MVC模式：SpringMVC为MVC模型提供极佳的支持。MVC模式可以提高代码的复用性、灵活性和可扩展性，同时也有利于维护Web应用程序。

（2）灵活的配置：Spring MVC采用了灵活的配置方法，可以通过XML配置或注解的方式实现。

（3）易于测试：SpringMVC中的Controller类是POJO（纯Java对象），它们的测试很容易，可以用JUnit等测试框架进行测试。

（4）兼容OpenAPI：SpringMVC可以与其他框架集成，例如OpenAPI（Swagger），可以为API文档和开发者交互提供支持。