**51. 解释AOP**

**52. Aspect 切面**

**52. 在Spring AOP 中，关注点和横切关注的区别是什么？**

**54. 连接点**

**55. 通知**

**56. 切点**

**57. 什么是引入?**

**58. 什么是目标对象?**

**59. 什么是代理?**

**60. 有几种不同类型的自动代理？**

**61. 什么是织入。什么是织入应用的不同点？**

**62. 解释基于XML Schema方式的切面实现。**

**63. 解释基于注解的切面实现**

**64. 什么是Spring的MVC框架？**

**65. DispatcherServlet**

**66. WebApplicationContext**

**67. 什么是Spring MVC框架的控制器？**

**68. @Controller 注解**

**69. @RequestMapping 注解**

|  |  |
| --- | --- |
|  |  |

**答案解析**ID:javaxxf

**51. 解释AOP**

面向切面的编程，或AOP，是一种编程技术，允许程序模块化横向切割关注点，或横切典型的责任划分，如日志和事务管理。

**52. Aspect 切面**

AOP核心就是切面，它将多个类的通用行为封装成可重用的模块，该模块含有一组API提供横切功能。比如，一个日志模块可以被称作日志的AOP切面。根据需求的不同，一个应用程序可以有若干切面。在Spring AOP中，切面通过带有@Aspect注解的类实现。

**52. 在Spring AOP 中，关注点和横切关注的区别是什么？**

关注点是应用中一个模块的行为，一个关注点可能会被定义成一个我们想实现的一个功能。

横切关注点是一个关注点，此关注点是整个应用都会使用的功能，并影响整个应用，比如日志，安全和数据传输，几乎应用的每个模块都需要的功能。因此这些都属于横切关注点。

**54. 连接点**

连接点代表一个应用程序的某个位置，在这个位置我们可以插入一个AOP切面，它实际上是个应用程序执行Spring AOP的位置。

**55. 通知**

通知是个在方法执行前或执行后要做的动作，实际上是程序执行时要通过SpringAOP框架触发的代码段。

Spring切面可以应用五种类型的通知：

* **before**：前置通知，在一个方法执行前被调用。
* **after**: 在方法执行之后调用的通知，无论方法执行是否成功。
* **after-returning**:仅当方法成功完成后执行的通知。
* **after-throwing**: 在方法抛出异常退出时执行的通知。
* **around**: 在方法执行之前和之后调用的通知。

**56. 切点**

切入点是一个或一组连接点，通知将在这些位置执行。可以通过表达式或匹配的方式指明切入点。

**57. 什么是引入?**

引入允许我们在已存在的类中增加新的方法和属性。

**58. 什么是目标对象?**

被一个或者多个切面所通知的对象。它通常是一个代理对象。也指被通知（advised）对象。

**59. 什么是代理?**

代理是通知目标对象后创建的对象。从客户端的角度看，代理对象和目标对象是一样的。

**60. 有几种不同类型的自动代理？**

1. BeanNameAutoProxyCreator
2. DefaultAdvisorAutoProxyCreator
3. Metadataautoproxying

**61. 什么是织入。什么是织入应用的不同点？**

织入是将切面和到其他应用类型或对象连接或创建一个被通知对象的过程。

织入可以在编译时，加载时，或运行时完成。

**62. 解释基于XML Schema方式的切面实现。**

在这种情况下，切面由常规类以及基于XML的配置实现。

**63. 解释基于注解的切面实现**

在这种情况下(基于@AspectJ的实现)，涉及到的切面声明的风格与带有java5标注的普通java类一致。

**64. 什么是Spring的MVC框架？**

Spring 配备构建Web 应用的全功能MVC框架。Spring可以很便捷地和其他MVC框架集成，如Struts，Spring 的MVC框架用控制反转把业务对象和控制逻辑清晰地隔离。它也允许以声明的方式把请求参数和业务对象绑定。

**65. DispatcherServlet**

Spring的MVC框架是围绕DispatcherServlet来设计的，它用来处理所有的HTTP请求和响应。

**66. WebApplicationContext**

WebApplicationContext继承了ApplicationContext 并增加了一些WEB应用必备的特有功能，它不同于一般的ApplicationContext，因为它能处理主题，并找到被关联的servlet。

**67. 什么是Spring MVC框架的控制器？**

控制器提供一个访问应用程序的行为，此行为通常通过服务接口实现。控制器解析用户输入并将其转换为一个由视图呈现给用户的模型。Spring用一个非常抽象的方式实现了一个控制层，允许用户创建多种用途的控制器。

**68. @Controller 注解**

该注解表明该类扮演控制器的角色，Spring不需要你继承任何其他控制器基类或引用Servlet API。

**69. @RequestMapping 注解**

该注解是用来映射一个URL到一个类或一个特定的方处理法上。