

41、EJB 规范规定 EJB 中禁止的操作有哪些?

答: 1.不能操作线程和线程 API(线程 API 指非线程对象的方法如 notify,wait 等), 2.不能操作 awt, 3.不能实现服务器功能, 4.不能对静态属性存取, 5.不能使用 IO 操作直接存取文件系统, 6.不能加载本地库., 7.不能将 this 作为变量和返回, 8.不能循环调用。

42、remote 接口和 home 接口主要作用

答: remote 接口定义了业务方法, 用于 EJB 客户端调用业务方法。

home 接口是 EJB 工厂用于创建和移除查找 EJB 实例

43、bean 实例的生命周期

答: 对于 Stateless Session Bean、Entity Bean、Message Driven Bean 一般存在缓冲池管理, 而对于 Entity Bean 和 Statefull Session Bean 存在 Cache 管理, 通常包含创建实例, 设置上下文、创建 EJB Object (create)、业务方法调用、remove 等过程, 对于存在缓冲池管理的 Bean, 在 create 之后实例并不从内存清除, 而是采用缓冲池调度机制不断重用实例, 而对于存在 Cache 管理的 Bean 则通过激活和去激活机制保持 Bean 的状态并限制内存中实例数量。

44、EJB 的激活机制

答: 以 Stateful Session Bean 为例: 其 Cache 大小决定了内存中可以同时存在的 Bean 实例的数量, 根据 MRU 或 NRU 算法, 实例在激活和去激活状态之间迁移, 激活机制是当客户端调用某个 EJB 实例业务方法时, 如果对应 EJB Object 发现自己没有绑定对应的 Bean 实例则从其去激活 Bean 存储中 (通过序列化机制存储实例) 回复 (激活) 此实例。状态变迁前会调用对应的 ejbActive 和 ejbPassivate 方法。

45、EJB 的几种类型

答: 会话 (Session) Bean, 实体 (Entity) Bean 消息驱动的 (Message Driven) Bean

会话 Bean 又可分为有状态 (Stateful) 和无状态 (Stateless) 两种

实体 Bean 可分为 Bean 管理的持续性 (BMP) 和容器管理的持续性 (CMP) 两种

46、客户端调用 EJB 对象的几个基本步骤

答: 设置 JNDI 服务工厂以及 JNDI 服务地址系统属性, 查找 Home 接口, 从 Home 接口调用 Create 方法创建 Remote 接口, 通过 Remote 接口调用其业务方法。

应用服务器方面

47、如何给 weblogic 指定大小的内存?

答: 在启动 Weblogic 的脚本中 (位于所在 Domian 对应服务器目录下的 startServerName), 增加 set MEM_ARGS=-Xms32m -Xmx200m, 可以调整最小内存为 32M, 最大 200M

EJB 需直接实现它的业务接口或 Home 接口吗, 请简述理由。

远程接口和 Home 接口不需要直接实现, 他们的实现代码是由服务器产生的, 程序运行中对应实现类会作为对应接口类型的实例被使用。

48、应用服务器有那些?

答: BEA WebLogic Server, IBM WebSphere Application Server, Oracle9i Application Server, jBoss, Tomcat

49、如何设定的 weblogic 的热启动模式(开发模式)与产品发布模式?

答: 可以在管理控制台中修改对应服务器的启动模式为开发或产品模式之一。或者修改服务的启动文件或者 commenv 文件, 增加 set PRODUCTION_MODE=true。

50、如何启动时不需输入用户名与密码?

答: 修改服务启动文件, 增加 WLS_USER 和 WLS_PW 项。也可以在 boot.properties 文件中增加加密过的用户名和密码。

51、在 weblogic 管理制台中对一个应用域(或者说是一个网站,Domain)进行 jms 及 ejb 或连接池等相关信息进行配置后, 实际保存在什么文件中?

答: 保存在此 Domain 的 config.xml 文件中, 它是服务器的核心配置文件。

52、说说 weblogic 中一个 Domain 的缺省目录结构?比如要将一个简单的 helloWorld.jsp 放入何目录下,然的在浏览器上就可打入 aspestratio="t" v:ext="edit">http://主机:端口号//helloworld.jsp 就可以看到运行结果了? 又比如这其中用到了一个自己写的 javaBean 该如何办?

答: Domain 目录服务器目录 applications, 将应用目录放在此目录下将可以作为应用访问, 如果是 Web 应用, 应用目录需要满足 Web 应用目录要求, jsp 文件可以直接放在应用目录中, Javabean 需要放在应用目录的 WEB-INF 目录的 classes 目录中, 设置服务器的缺省应用将可以实现在浏览器上无需输入应用名。

53、在 weblogic 中发布 ejb 需涉及到哪些配置文件

答: 不同类型的 EJB 涉及的配置文件不同, 都涉及到的配置文件包括 ejb-jar.xml,weblogic-ejb-jar.xmlCMP 实体 Bean 一般还需要 weblogic-cmp-rdbms-jar.xml

54、如何在 weblogic 中进行 ssl 配置与客户端的认证配置或说说 j2ee(标准)进行 ssl 的配置

答: 缺省安装中使用 DemoIdentity.jks 和 DemoTrust.jks KeyStore 实现 SSL, 需要配置服务器使用 Enable SSL, 配置其

端口, 在产品模式下需要从 CA 获取私有密钥和数字证书, 创建 identity 和 trust keystore, 装载获得的密钥和数字证书。可以配置此 SSL 连接是单向还是双向的。

55、如何查看在 weblogic 中已经发布的 EJB?

答: 可以使用管理控制台, 在它的 Deployment 中可以查看所有已发布的 EJB

说说在 weblogic 中开发消息 Bean 时的 persistent 与 non-persisten 的差别

persistent 方式的 MDB 可以保证消息传递的可靠性, 也就是如果 EJB 容器出现问题而 JMS 服务器依然会将消息在此 MDB 可用的时候发送过来, 而 non-persistent 方式的消息将被丢弃。

J2EE,MVC 方面

56、MVC 的各个部分都有那些技术来实现?如何实现?

答: MVC 是 Model-View-Controller 的简写。"Model" 代表的是应用的业务逻辑(通过 JavaBean, f 组件实现), "View" 是应用的表示面(由 JSP 页面产生), "Controller" 是提供应用的处理过程控制(一般是一个 Servlet), 通过这种设计模型把应用逻辑, 处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

57、J2EE 是什么?

答: Je22 是 Sun 公司提出的多层(multi-tiered), 分布式(distributed), 基于组件(component-base)的企业级应用模型(enterprise application model). 在这样的一个应用系统中, 可按照功能划分为不同的组件, 这些组件又可在不同计算机上, 并且处于相应的层次(tier)中。所属层次包括客户层(client tier)组件, web 层和组件, Business 层和组件, 企业信息系统(EIS)层。

58、WEB SERVICE 名词解释。JSDDL 开发包的介绍。JAXP、JAXM 的解释。SOAP、UDDI,WSDL 解释。

答: Web Service 描述语言 WSDL

SOAP 即简单对象访问协议(Simple Object Access Protocol), 它是用于交换 XML 编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准; UDDI 是一套基于 Web 的、分布式的、为 Web Service 提供的、信息注册中心的实现标准规范, 同时也包含一组使企业能将自身提供的 Web Service 注册, 以使别的企业能够发现的访问协议的实现标

准。

59、J2EE 是技术还是平台还是框架?

答: J2EE 本身是一个标准, 一个为企业分布式应用的开发提供的标准平台。

J2EE 也是一个框架, 包括 JDBC、JNDI、RMI、JMS、EJB、JTA 等技术。

60、STRUTS 的应用(如 STRUTS 架构)

答: Struts 是采用 Java Servlet/JavaServer Pages 技术, 开发 Web 应用程序的开放源码的 framework。采用 Struts 能开发出基于 MVC(Model-View-Controller)设计模式的应用构架。Struts 有如下的主要功能:

一.包含一个 controller servlet, 能将用户的请求发送到相应的 Action 对象。

二.JSP 自由 tag 库, 并且在 controller servlet 中提供关联支持, 帮助开发人员创建交互式表单应用。

三.提供了一系列实用对象: XML 处理、通过 Java reflection APIs 自动处理 JavaBeans 属性、国际化的提示和消息。

61、WEB SERVICE 名词解释。JSDDL 开发包的介绍。JAXP、JAXM 的解释。SOAP、UDDI,WSDL 解释。

答: Web Service Web Service 是基于网络的、分布式的模块化组件, 它执行特定的任务, 遵守具体的技术规范, 这些规范使得 Web Service 能与其他兼容的组件进行互操作。

JAXP(Java API for XML Parsing) 定义了使用 DOM, SAX, XSLT 的通用的接口。这样在你的程序中你只要使用这些通用的接口, 当你需要改变具体的实现时候也不需要修改代码。

JAXM(Java API for XML Messaging) 是为 SOAP 通信提供访问方法和传输机制的 API。

WSDL 是一种 XML 格式, 用于将网络服务描述为一组端点, 这些端点对包含面向文档信息或面向过程信息的信息进行操作。这种格式首先对操作和消息进行抽象描述, 然后将其绑定到具体的网络协议和消息格式上以定义端点。相关的具体端点即组合成为抽象端点(服务)。

SOAP 即简单对象访问协议(Simple Object Access Protocol), 它是用于交换 XML 编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准; UDDI 是一套基于 Web 的、分布式的、为 Web Service 提供的、信息注册中心的实现标准规范, 同时也包含一组使企业能将自身提供的 Web Service 注册, 以使别的企业能够发现的访问协议的实现标准。

62、C/S 与 B/S 区别:

答: 有如下八个方面的不同:

1. 硬件环境不同:

C/S 一般建立在专用的网络上, 小范围里的网络环境, 局域网之间再通过专门服务器提供连接和数据交换服务。

B/S 建立在广域网之上的, 不必是专门的网络硬件环境, 例与电话上网, 租用设备. 信息自己管理. 有比 C/S 更强的适应范围, 一般只要有操作系统和浏览器就行

2. 对安全要求不同

C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强. 一般高度机密的信息系统采用 C/S 结构适宜. 可以通过 B/S 发布部分可公开信息.

B/S 建立在广域网之上, 对安全的控制能力相对弱, 可能面向不可知的用户。

3. 对程序架构不同

C/S 程序可以更加注重流程, 可以对权限多层次校验, 对系统运行速度可以较少考虑.

B/S 对安全以及访问速度的多重的考虑, 建立在需要更加优化的基础之上. 比 C/S 有更高的要求 B/S 结构的程序架构是发展的趋势, 从 MS 的 .Net 系列的 BizTalk 2000 Exchange 2000 等, 全面支持网络的构件搭建的系统. SUN 和 IBM 推的 JavaBean 构件技术等, 使 B/S 更加成熟.

4. 软件重用不同

C/S 程序可以不可避免的整体性考虑, 构件的重用性不如在 B/S 要求下的构件的重用性好.

B/S 对的多重结构, 要求构件相对独立的功能. 能够相对较好的重用. 就入买来的餐桌可以再利用, 而不是做在墙上的石头桌子

5. 系统维护不同

C/S 程序由于整体性, 必须整体考察, 处理出现的问题以及系统升级. 升级难. 可能是再做一个全新的系统

B/S 构件组成, 方面构件个别的更换, 实现系统的无缝升级. 系统维护开销减到最小. 用户从网上自己下载安装就可以实现升级.

6. 处理问题不同

C/S 程序可以处理用户面固定, 并且在相同区域, 安全要求高需求, 与操作系统相关. 应该都是相同的系统

B/S 建立在广域网上, 面向不同的用户群, 分散地域, 这是 C/S 无法作到的. 与操作系统平台关系最小.

7. 用户接口不同

C/S 多是建立的 Window 平台上, 表现方法有限, 对程序员普遍要求较高

B/S 建立在浏览器上, 有更加丰富和生动的表现方式与用户交流. 并且大部分难度减低, 减低开发成本.

8. 信息流不同

C/S 程序一般是典型的中央集权的机械式处理, 交互性相对低

B/S 信息流向可变化, B-B B-C B-G 等信息、流向的变化, 更像交易中心。

63、什么是 web 容器

答: 给处于其中的应用程序组件 (JSP, SERVLET) 提供一个环境, 使 JSP, SERVLET 直接更容器中的环境变量接**互, 不必关注其它系统问题. 主要有 WEB 服务器来实现. 例如: TOMCAT, WEBLOGIC, WEBSPHERE 等. 该容器提供的接口严格遵守 J2EE 规范中的 WEB APPLICATION 标准. 我们把遵守以上标准的 WEB 服务器就叫做 J2EE 中的 WEB 容器。

64、EJB 容器

答: Enterprise java bean 容器. 更具有行业领域特色. 他提供给运行在其中的组件 EJB 各种管理功能. 只要满足 J2EE 规范的 EJB 放入该容器, 马上就会被容器进行高效率的管理. 并且可以通过现成的接口来获得系统级别的服务. 例如邮件服务、事务管理

65、什么是 JNDI

答: (Java Naming & Directory Interface) JAVA 命名目录服务. 主要提供的功能是: 提供一个目录系统, 让其它各地的应用程序在其上面留下自己的索引, 从而满足快速查找和定位分布式应用程序的功能。

66、什么是 JMS

答: (Java Message Service) JAVA 消息服务. 主要实现各个应用程序之间的通讯. 包括点对点和广播。

67、什么是 JTA

答: (Java Transaction API) JAVA 事务服务. 提供各种分布式事务服务. 应用程序只需调用其提供的接口即可。

68、什么是 JAF

答: (Java Action FrameWork) JAVA 安全认证框架. 提供一些安全控制方面的框架. 让开发者通过各种部署和自定义实现自己的个性安全控制策略。

RMI/IIOP: (Remote Method Invocation /internet 对象请求中介协议) 他们主要用于通过远程调用服务. 例如, 远程有一台计算机上运行一个程序, 它提供股票分析服务, 我们可以在本地计算机上实现对其直接调用. 当然这是要通过一定的规范才能在异构的系统之间进行通信. RMI 是 JAVA 特有的。

69、MVC 的各个部分都有那些技术来实现? 如何实现?

答: MVC 是 Model—View—Controller 的简写. "Model" 代表的是应用的业务逻辑 (通过 JavaBean, EJB 组件实现), "View" 是应用的表示面 (由 JSP 页面产生), "Controller" 是提供应用的处理过程控制 (一般是一个 Servlet), 通过这种设计模型把应用逻辑, 处理过程和显示逻辑分成不同的组件实现. 这些组件可以进行交互和重用。

设计模式方面

70、开发中都用到那些设计模式?用在什么场合?

答: 每个模式都描述了一个在我们的环境中不断出现的问题, 然后描述了该问题的解决方案的核心。通过这种方式, 你可以无数次地使用那些已有的解决方案, 无需在重复相同的工作。主要用到了 MVC 的设计模式。用来开发 JSP/Servlet 或者 J2EE 的相关应用。简单工厂模式等。

71、说说你所熟悉或听说过的 j2ee 中的几种常用模式?及对设计模式的一些看法

答: Session Facade Pattern: 使用 SessionBean 访问 EntityBean

Message Facade Pattern: 实现异步调用

EJB Command Pattern: 使用 Command JavaBeans 取代 SessionBean, 实现轻量级访问

Data Transfer Object Factory: 通过 DTO Factory 简化 EntityBean 数据提供特性

Generic Attribute Access: 通过 AttributeAccess 接口简化 EntityBean 数据提供特性

Business Interface: 通过远程 (本地) 接口和 Bean 类实现相同接口规范业务逻辑一致性

E J B 架构的设计好坏将直接影响系统的性能、可扩展性、可维护性、组件可重用性及开发效率。项目越复杂, 项目队伍越庞大则越能体现良好设计的重要性。

72、j2ee 常用的设计模式? 说明工厂模式。

答: Java 中的 23 种设计模式:

Factory (工厂模式), Builder (建造模式), Factory Method (工厂方法模式),

Prototype (原始模型模式), Singleton (单例模式), Facade (门面模式),

Adapter (适配器模式), Bridge (桥梁模式), Composite (合成模式),

Decorator (装饰模式), Flyweight (享元模式), Proxy (代理模式),

Command (命令模式), Interpreter (解释器模式), Visitor (访问者模式),

Iterator (迭代子模式), Mediator (调停者模式), Memento (备忘录模式),

Observer (观察者模式), State (状态模式), Strategy (策略模式),

Template Method (模板方法模式), Chain Of Responsibility (责任链模式)

工厂模式: 工厂模式是一种经常被使用到的模式, 根据工厂模式实现的类可以根据提供的数据生成一组类中某一个类的实例, 通常这一组类有一个公共的抽象父类并且实现了相同的方法, 但是这些方法针对不同的数据进行了不同的操作。首先需要定义一个基类, 该类的子类通过不同的方法实现了基类中的方法。然后需要定义一个工厂类, 工厂类可以根据条件生成不同的子类实例。当得到子类的实例后, 开发人员可以调用基类中的方法而不必考虑到底返回的是哪一个子类的实例。

73、CORBA 是什么?用途是什么?

答: CORBA 标准是公共对象请求代理结构(Common Object Request Broker Architecture), 由对象管理组织 (Object Management Group, 缩写为 OMG) 标准化。它的组成是接口定义语言(IDL), 语言绑定(binding:也译为联编)和允许应用程序间互操作的协议。其目的为:

用不同的程序设计语言书写

在不同的进程中运行

为不同的操作系统开发

74、Linux 下线程, GDI 类的解释。

答: Linux 实现的就是基于核心轻量级进程的"一对一"线程模型, 一个线程实体对应一个核心轻量级进程, 而线程之间的管理在核外函数库中实现。

GDI 类为图像设备编程接口类库。