中间件技术名词解释及简答题

目录

[名词解释题 3](#_Toc8495604)

[什么是DCE（分布计算环境）？ 3](#_Toc8495605)

[什么是RPC（远程过程调用）？ 3](#_Toc8495606)

[什么是RMI（远程方法激活）？ 3](#_Toc8495607)

[什么是ORB（对象请求代理）？ 3](#_Toc8495608)

[什么是MOM（面向消息中间件）？ 3](#_Toc8495609)

[什么是数据库访问中间件？ 3](#_Toc8495610)

[什么是事务？ 3](#_Toc8495611)

[什么是TPM（交易中间件）？ 3](#_Toc8495612)

[什么是COM？ 3](#_Toc8495613)

[什么是组件？ 4](#_Toc8495614)

[什么是DCOM？ 4](#_Toc8495615)

[什么是MTS？ 4](#_Toc8495616)

[~~什么是.NET？~~ 4](#_Toc8495617)

[什么是EJB？ 4](#_Toc8495618)

[什么是EJB容器？ 4](#_Toc8495619)

[什么是JNDI？ 4](#_Toc8495620)

[什么是JMS？ 4](#_Toc8495621)

[什么是CORBA？ 4](#_Toc8495622)

[什么是GIOP（通用ORB互通协议）？ 4](#_Toc8495623)

[什么是IIOP（互联网ORB互通协议）？ 4](#_Toc8495624)

[什么是X/Open DTP？ 4](#_Toc8495625)

[什么是SOAP（简单对象访问协议）？ 4](#_Toc8495626)

[什么是WSDL（网络服务描述语言）？ 5](#_Toc8495627)

[什么是UDDI？ 5](#_Toc8495628)

[什么是SOA（面向服务架构）？ 5](#_Toc8495629)

[什么是EAI（企业应用集成）？ 5](#_Toc8495630)

[什么是IDL（界面定义语言）？ 5](#_Toc8495631)

[什么是反射中间件？ 5](#_Toc8495632)

[简答题 5](#_Toc8495633)

[开放系统的特征 5](#_Toc8495634)

[中间件定义 5](#_Toc8495635)

[中间件分类 5](#_Toc8495636)

[中间件功能 5](#_Toc8495637)

[分布计算环境提供哪些服务？ 6](#_Toc8495638)

[DCE的优缺点 6](#_Toc8495639)

[远程过程调用中间件工作流程 6](#_Toc8495640)

[远程过程调用的特点 6](#_Toc8495641)

[在RMI中Stub和Skeleton的作用分别是什么? 6](#_Toc8495642)

[消息中间件的功能 6](#_Toc8495643)

[消息中间件的两种模式 7](#_Toc8495644)

[几种常见的数据库访问中间件 7](#_Toc8495645)

[数据库访问中间件优点 7](#_Toc8495646)

[交易中间件作用 7](#_Toc8495647)

[交易中间件功能 7](#_Toc8495648)

[COM技术的体系结构 7](#_Toc8495649)

[DCOM与COM之间的关系 8](#_Toc8495650)

[~~.NET与COM之间的关系~~ 8](#_Toc8495651)

[EJB的分类 8](#_Toc8495652)

[显式中间件与隐式中间件有什么区别？EJB容器如何实现隐式中间件？ 8](#_Toc8495653)

[CORBA体系结构 8](#_Toc8495654)

[ORB系统运行过程概述 8](#_Toc8495655)

[CORBA核心服务 8](#_Toc8495656)

[CORBA中OA（对象适配器）的作用 9](#_Toc8495657)

[ORB中界面仓库和实现仓库 9](#_Toc8495658)

[GIOP的组成部分 9](#_Toc8495659)

[事务的ACID特性 9](#_Toc8495660)

[分布式事务处理模型X/Open DTP 的组成 9](#_Toc8495661)

[X/Open DTP模型各组成部分之间的作用关系： 9](#_Toc8495662)

[两阶段提交协议（2PC）的工作流程 9](#_Toc8495663)

[Web Service的优缺点 10](#_Toc8495664)

[COM类工程机制 10](#_Toc8495665)

[EJB的生命周期 11](#_Toc8495666)

[EJB三种Bean相关知识 12](#_Toc8495667)

[客户端使用EJB服务的工作流程 12](#_Toc8495668)

[CORBA体系结构 13](#_Toc8495669)

[X/Open DTP 事务处理过程 13](#_Toc8495670)

[图 1 客户程序调用COM库创建组件对象顺序图 10](#_Toc8495671)

[图 2 无状态会话Bean的生命周期 11](#_Toc8495672)

[图 3 有状态会话Bean的生命周期 11](#_Toc8495673)

[图 4 EJB的三种Bean知识体系 12](#_Toc8495674)

[图 5 客户端使用EJB服务的工作流程 12](#_Toc8495675)

[图 6 CORBA体系结构 13](#_Toc8495676)

[图 7 X/Open DTP 模型 13](#_Toc8495677)

# 名词解释题

## 什么是DCE（分布计算环境）？

分布计算环境（DCE：Distributed Computing Environment）是开放软件基金会（OSF：Open Software Foundation）制定的以远程过程调用（RPC）为核心的一系列规范；

## 什么是RPC（远程过程调用）？

RPC是从一台机器或一个进程调用另一台机器或另一个进程的服务或方法。

## 什么是RMI（远程方法激活）？

远程方法激活，使软件开发人员能够开发这样的分布式程序：在这个分布式程序中，其远程对象的方法能够被运行在不同主机上的其他Java虚拟机（JVM）的方法所调用。

## 什么是ORB（对象请求代理）？

ORB（Object Request Broker）提供一种通信机制透明地在异构的通信环境中传递对象请求，各对象可以位于本地或远程机器，且对象之间的客户机/服务器的角色是可以互换的。

## 什么是MOM（面向消息中间件）？

面向消息中间件使分布式应用程序可以通过发送和接收消息来进行通信和交换数据。

## 什么是数据库访问中间件？

数据库访问中间件专门负责和完成对数据库访问操作的优化、用户连接数的管理、保证访问的安全性和可靠性，以及实现应用对来自不同厂商的数据库的访问等，进而提高对数据库访问的效率。

## 什么是事务？

事务是指一个程序或程序段，在一个或多个资源上为完成某些功能的执行过程。

## 什么是TPM（交易中间件）？

在开放系统环境下提供保证交易完整性和数据一致性的一种环境平台。

## 什么是COM？

COM（Component Object Model，即组件对象模型）是一种以组件为发布单元的对象模型，这种模型使各软件组件可以用一种统一的方式进行交互。

COM既提供了组件之间交互的规范（组件模型），也提供了实现交互的环境（COM库）。

## 什么是组件？

组件是具有特定功能的，能够跨越进程的边界，实现跨网络、语言、应用程序、开发工具和操作系统的“即拆即用”的独立对象。

## 什么是DCOM？

DCOM（Distributed Component Object Model)是分布式应用环境中的COM技术。

支持不同计算机上组件对象与客户程序之间或者组件对象之间的相互通信。

## 什么是MTS？

MTS（Microsoft Transaction Server),COM组件的一致管理系统，为COM技术的企业应用提供基本的组件管理与配置环境。

## ~~什么是.NET？~~

.NET是Microsoft XML Web services平台，允许应用程序通过Internet进行通信和共享数据。

.NET是COM技术的进一步发展，它扩展了通过任何设备随时随地操作数据和进行通信的能力。

## 什么是EJB？

EJB是一个由Java语言开发的，可在多层的分布式环境中部署的服务器端软件组件,是J2EE技术的核心。

EJB的核心思想是将商业逻辑与底层的系统逻辑分开，使开发者只需关心商业逻辑，而由EJB容器实现目录服务、事务处理、持久性、安全性等底层系统逻辑，并且可创建具有动态扩展性的服务器应用。

## 什么是EJB容器？

EJB容器是EJB组件的运行环境。EJB容器装载EJB，并且使得客户端程序能进行远程调用。

## 什么是JNDI？

Java名称和目录接口，用于网络中定位组件和其他资源。

## 什么是JMS？

JMS（java 消息服务）用于分布式对象的异步通信，Java消息服务是一个与具体平台无关的API。

## 什么是CORBA？

CORBA（Common Object Request Broker Architecture ）是由OMG提出的应用软件体系结构和对象技术规范，其核心是一套标准的语言、接口和协议，以支持异构分布应用程序间的互操作性及独立于平台和编程语言的对象重用。

## 什么是GIOP（通用ORB互通协议）？

GIOP是一种通信协议，它规定了两个实体：客户和服务器ORB间的通信机制。

## 什么是IIOP（互联网ORB互通协议）？

是GIOP的一种映射，它定义了用于Internet之上的一种标准互操作协议，它利用的运输层协议就是Internet所采用的TCP/IP协议。

## 什么是X/Open DTP？

X/Open DTP是一个用于规范在分布、异构的环境下应用程序与不同资源之间相互协调进行事务处理的标准。

## 什么是SOAP（简单对象访问协议）？

SOAP 是基于 XML 的简易协议，可使应用程序在 HTTP 之上进行信息交换。

或者更简单地说：SOAP 是用于访问网络服务的协议。

## 什么是WSDL（网络服务描述语言）？

WSDL（网络服务描述语言，Web Services Description Language）是一门基于 XML 的语言，用于描述 Web Services 以及如何对它们进行访问。

## 什么是UDDI？

UDDI是一种用于描述、发现、集成Web Service的技术，它是Web Service协议栈的一个重要部分。

## 什么是SOA（面向服务架构）？

面向服务的架构（SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）进行拆分，并通过这些服务之间定义良好的接口和契约联系起来。

## 什么是EAI（企业应用集成）？

EAI（**E**nterprise **A**pplication **I**ntegration，企业应用集成）是将基于各种不同平台、用不同方案建立的异构应用集成的一种方法和技术。

## 什么是IDL（界面定义语言）？

IDL是一种描述性语言，属于远程过程调用RPC技术方面，IDL对接口和组件进行描述，指定接口或组件的属性信息用来生成所需要的代理/存根代码、调度代码或者类型库。IDL描述的接口和组件等数据类型是各种流行语言都能识别和支持的。

## 什么是反射中间件？

一个程序可以访问、推理和改变其自身的解释。

# 简答题

## 开放系统的特征

**分布** 任务已不只是在单机上运行，而是由网络中多台计算机上的相关应用共同协作完成；

**异构**  计算机硬件、操作系统、网络协议、数据库系统以及开发工具种类不同；

**动态协作(互操作)** 参与协作的应用允许位置透明性、迁移透明性、负载平衡性等需求。

## 中间件定义

中间件(Middleware)是一种软件，处于系统软件（操作系统和网络软件）与应用软件之间，它能使应用软件之间进行跨网络的协同工作（也就是互操作），这时允许各应用软件之下所涉及的“系统结构、操作系统、通信协议、数据库和其它应用服务”各不相同。

## 中间件分类

1. 终端仿真/屏幕转换中间件
2. 数据访问中间件
3. 远程过程调用中间件
4. 面向消息中间件
5. 事务（交易）中间件
6. 分布式对象中间件

## 中间件功能

1. 跨平台，包括不同的网络、主机、数据库
2. 支持复杂的应用模式
3. 不同应用系统的集成
4. 保证系统的处理效率
5. 保证系统的高可用性
6. 保证系统的动态扩展能力
7. 提供调试、维护、管理工具
8. 保障系统安全
9. 加快开发周期、保证质量

## 分布计算环境提供哪些服务？

提供线程服务、RPC服务、目录服务、安全服务、分布式文件服务以及时间服务；

## DCE的优缺点

优点：

1. 完善的RPC机制、安全服务、命名服务
2. 在微软平台上提供了许多DCE/RPC的功能

缺点：

1. 缺乏概念模型，给开发带来困难
2. 后来引进了面向对象DCE(即OODCE)，但这一包层，又给DCE带来更多层次，使系统交互过多，也增加了维护困难，
3. 整个系统必须完整安装（安装困难、许多功能由于用不到而闲置）

## 远程过程调用中间件工作流程

1. 用中间件中的IDL对调用顺序加以描述；
2. 利用IDL编译器对这种定义和描述进行编译，从而生成支持客户机和服务器进行通信的管道；
3. 在服务器端，被调用的远端过程需单独编写，并被置入服务器端框架，同时要在远程过程调用中间件中进行注册，以备调用。
4. 启动本地客户端存根-->存根捆绑远端过程名和参数-->利用网络传输到服务器端框架，从而实现数据格式的转换和参数的传递
5. 采用调用/返回模式具体地完成远端过程调用。

## 远程过程调用的特点

1. 客户/服务器模式

客户端的应用调用一个位于远端服务器平台的进程或服务，RPC实际上是通过C/S方式实现的。

1. 数据封装性

远端过程调用负责捆绑参数，实现不同平台之间参数的传递，发送参数到远端过程。

1. 同步性

当客户机发出请求时，服务器进程处于运行状态。

1. 复杂性

远程调用的实现环境可能要用系统级的任务来创建，但一旦完成，对程序员就是透明的了。

## 在RMI中Stub和Skeleton的作用分别是什么?

1. 桩是客户内部完成打包数据和管理网络连接工作的本地对象，在客户调用远程对象上作为方法时，它实际调用的本地存根对象上的方法。
2. 框架负责接收并解包远程方法调用为本地方法调用。

## 消息中间件的功能

1. 无连接消息传递
2. 消息优先化
3. 有保障的消息传递
4. 事务处理消息
5. 动态队列创建
6. 消息路由
7. 不同硬件系统的集成
8. 跨软件平台的支持

## 消息中间件的两种模式

1. 消息队列：

消息被发送到一个队列里，收件人可以在任何时候查看该队列，消息队列类似于运行的很好的电子邮件，传输质量的到保证，但并不知道收件人是否看到该消息。

1. 消息传递（发布-预定模式）：

发布-预定（消息传递）是把消息广播到多个收件人，发送方将消息发送到一个特定队列，客户机可以对该队列作预定，并从中获得消息。消息传递机制要保证将发送者的消息传送到目的地。在消息传递中，应用程序之间不必须建立直接的联系，从而实现消息传送的异步性。

## 几种常见的数据库访问中间件

1. ODBC 2. OLE DB 3. JDBC 4. 数据库引擎 5. 数据库网关

## 数据库访问中间件优点

* 1. 移植性好

中间件封装了各种与平台有关的细节，使更换操作系统和通信协议等底层的配置无需改变应用程序代码。

* 1. 集成方便

能非常容易地集成到应用开发环境中，无需大的代码改动。

* 1. 易于扩充

局部改进和整体升级只要保持对外接口不变就不会影响到系统的其他部分，在功能上对应用程序实现了透明性。

* 1. 使用简单

对各种数据源使用统一的访问方式，降低了用户参与程度。

## 交易中间件作用

①提供高效的处理能力 ②简化应用系统开发

③可以有效完成多服务进程和负载均衡 ④能保证分布式交易的完整性

⑤提高系统的容错能力 ⑥减少系统的运行、维护工作量

## 交易中间件功能

①进程管理 ②交易优先级管理

③数据传输 ④均衡负载

⑤权限控制、安全管理

## COM技术的体系结构

1. COM组件：具有一定逻辑功能，可独立发布的二进制文件，是组成应用程序的部件，在Windows平台上发布为DLL或者EXE。
2. COM接口：客户与对象之间的协议，客户使用COM接口调用COM对象的服务。
3. COM对象：实现COM接口，通过COM接口提供服务，支持一个或多个接口。
4. COM组件客户：组件的调用者，是应用程序中直接与用户交互的界面和调用组件的程序框架。
5. 组件服务器：通过若干个COM对象来实现应用程序所需的功能。

## DCOM与COM之间的关系

1. DCOM支持不同计算机上组件对象与客户程序之间或者组件对象之间的相互通信。
2. DCOM把COM技术的进程透明性拓展为位置透明性。
3. DCOM处理了底层网络协议的所有细节，编程人员只需把重点放在应用的业务逻辑上。
4. DCOM是COM的无缝扩展。

## ~~.NET与COM之间的关系~~

1. .NET框架简化了COM开发。
2. .NET框架组件依然是一种COM组件。
3. 在大型分布式应用开发方面，由于COM+是.NET框架的一部分，.NET框架类可直接访问COM+所提供的服务。

## EJB的分类

1. 会话Bean，典型的EJB类包含与商务过程相关的逻辑。必须实现javax.ejb.SessionBean接口。
2. 实体Bean，典型的EJB类包含与商务数据相关的逻辑。必须实现javax.ejb.EntityBean接口。
3. 消息Bean，典型的EJB类包含面向消息的逻辑。必须实现javax.ejb.MessageDrivenBean接口。

## 显式中间件与隐式中间件有什么区别？EJB容器如何实现隐式中间件？

显式使用中间件服务，是指手工调用中间件服务的API。隐式使用中间件服务，是指不用在代码中使用这类API。

通过请求拦截器实现隐式中间件。请求拦截器通过EJB在部署描述文件中描述的需求信息，知道怎样去使用API。

## CORBA体系结构

1. ORB 核心——软总线（跨平台）
2. OMG界面定义语言——IDL（跨语言）
3. 语言映射——C++ 或Java
4. 界面仓库和实现仓库——动态调用
5. 静态存根和框架——静态调用
6. 动态存根和框架——动态调用
7. 对象适配器——适配层
8. ORB之间的互操作——GIOP/IIOP

## ORB系统运行过程概述

1. 客户通过某种方式找到特定对象实现的对象引用；
2. 如果该对象实现有相应的静态存根，则客户可以通过该静态存根向对象实现发送请求，否则，在界面仓库的协助下，客户可以使用动态调用界面DII来向对象实现发出请求；
3. 当对象调用请求通过静态存根或动态调用界面DII到达ORB核心以后，ORB核心负责请求的传送，将其送给相应的对象适配器，其具体的请求传递方式由ORB的具体实现决定；
4. 对象适配器接到请求后，判断一下所请求的对象实现是否有静态框架存在，如果有，则对象适配器通过静态框架调用执行对象实现中的操作，否则，对象适配器将通过动态框架界面DSI中的动态实现例程来调用对象实现中的操作；
5. 对象实现的特定操作方法执行完成后，结果将按照对象请求传递和执行路径逆向返回给客户对象；

## CORBA核心服务

1. 对象定位：①名称服务 ②交易对象服务 ③对象引用字符串 ④对象工厂
2. 消息传输：①客户与对象间的通信机制 ②事件服务
3. 安全服务：身份认证、授权和存取控制、安全审计、加密和防抵赖等方面

## CORBA中OA（对象适配器）的作用

1. 对象引用的生成与解释；
2. 对象实现的注册；
3. 根据对象引用找到它对应的对象实现（定位）；
4. 服务器进程的激活；
5. 对象的激活与撤销；
6. 对象向上调用。

## ORB中界面仓库和实现仓库

1. 界面仓库存储各个界面信息的模块，如用IDL编写的界面定义、常量、类型等。它本身作为一个对象而存在。
2. 实现仓库存储对象实现的信息，并且它包含服务器的详细信息（即，哪一个执行程序需要被放置到哪一个服务器上）。

## GIOP的组成部分

1. 公共数据表示：OMG IDL数据类型在网上传输时的编码方案。它对所有IDL数据类型的映射都作了规定。
2. GIOP消息格式：规定了Client和Server两个角色之间要传输的消息格式。
3. GIOP消息传递：主要规定在任何面向连接的网络传输层上的一些操作规则。

## 事务的ACID特性

1. 原子性（Atomicity)——指事务中的所有操作是一个不可分割的整体，正如人们以前认为原子是不可分割的一样。
2. 一致性（Consistency)——保持在事务前后，事务所涉及的资源的数据一致性。
3. 孤立性（Isolation)——一个事务只有在结束后才对其他事务产生影响，其中间结果对其他事务是不可见的。
4. 持久性（Durability)——事务一旦成功完成，其影响是持久的，即使系统瘫痪，也应该能恢复到事务最近完成的状态。

## 分布式事务处理模型X/Open DTP 的组成

1. 应用程序（AP）：事务的使用者,它发出开始、提交或回滚事务的请求。它规定了一个事务的界限，并给出事务所包含的操作。
2. 资源管理器（RM）：提供对共享资源的访问。
3. 事务管理器（TM）：AP与RM之间的协调员。它给每个事务分配标识符，监视它们的进展，保证事务处理的顺利进行，并负责事务在失败情况下的恢复。

## X/Open DTP模型各组成部分之间的作用关系：

1. AP-RM之间采用RM的特有API接口，X/Open对这部分的规范较少。
2. AP-TM之间采用TX接口。AP利用这个接口向TM发出开始和结束一个事务的请求，进行全局事务的管理。
3. TM-RM之间采用XA双向接口。TM利用这个接口使各个RM合作完成一个全局事务。

## 两阶段提交协议（2PC）的工作流程

阶段一：准备阶段

TM询问所有的RM是否已经准备好提交。

如果一个RM给出否定的回答，该RM可以自动进行回滚，并“忘记”有关该事务的信息。

阶段二：提交阶段

TM检查所有RM的回答，只要有一个RM给出否定的回答，TM就指示所有的RM进行回滚。

否则，TM将指示所有的RM提交。

## Web Service的优缺点

优点：

1. 可操作的的分布式应用程序。
2. 普遍性、使用HTTP和XML进行通信。
3. Web Service 甚至可以穿越防火墙，真正的自由通信。
4. 通过 SOAP 协议实现异地调用。

缺点：

1. 使用单机应用程序时资源消耗很大。
2. 在同一局域网的同构应用程序，使用DCOM技术比Web Service技术的效率更高。

## COM类工程机制

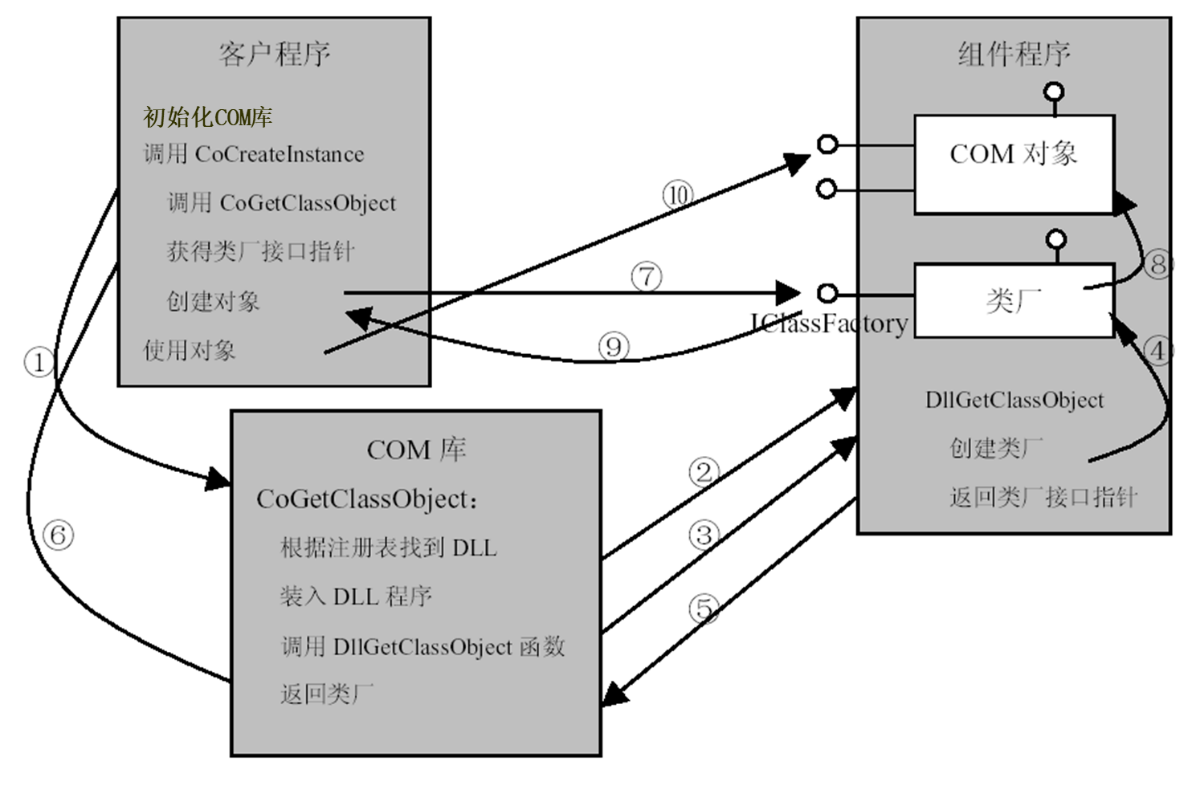


图 1 客户程序调用COM库创建组件对象顺序图

## EJB的生命周期

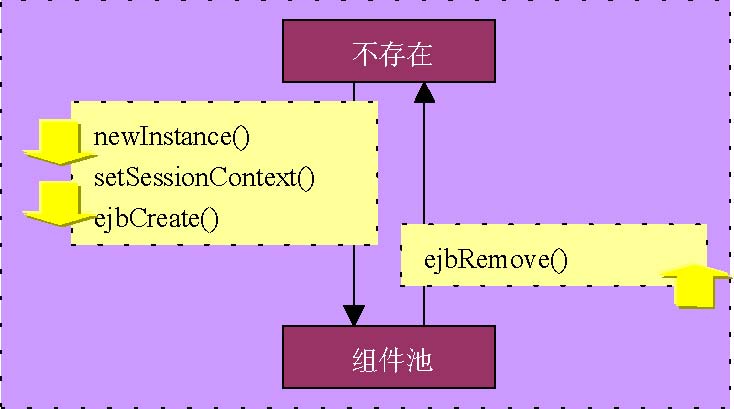


图 2 无状态会话Bean的生命周期

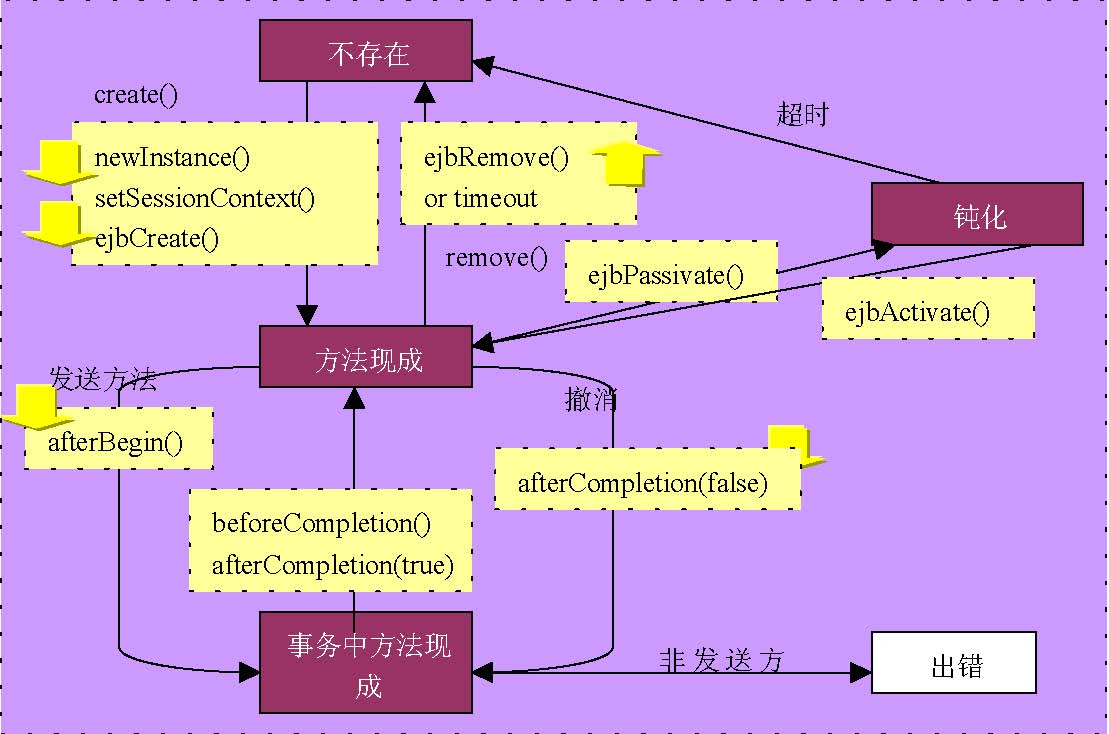


图 3 有状态会话Bean的生命周期

## EJB三种Bean相关知识

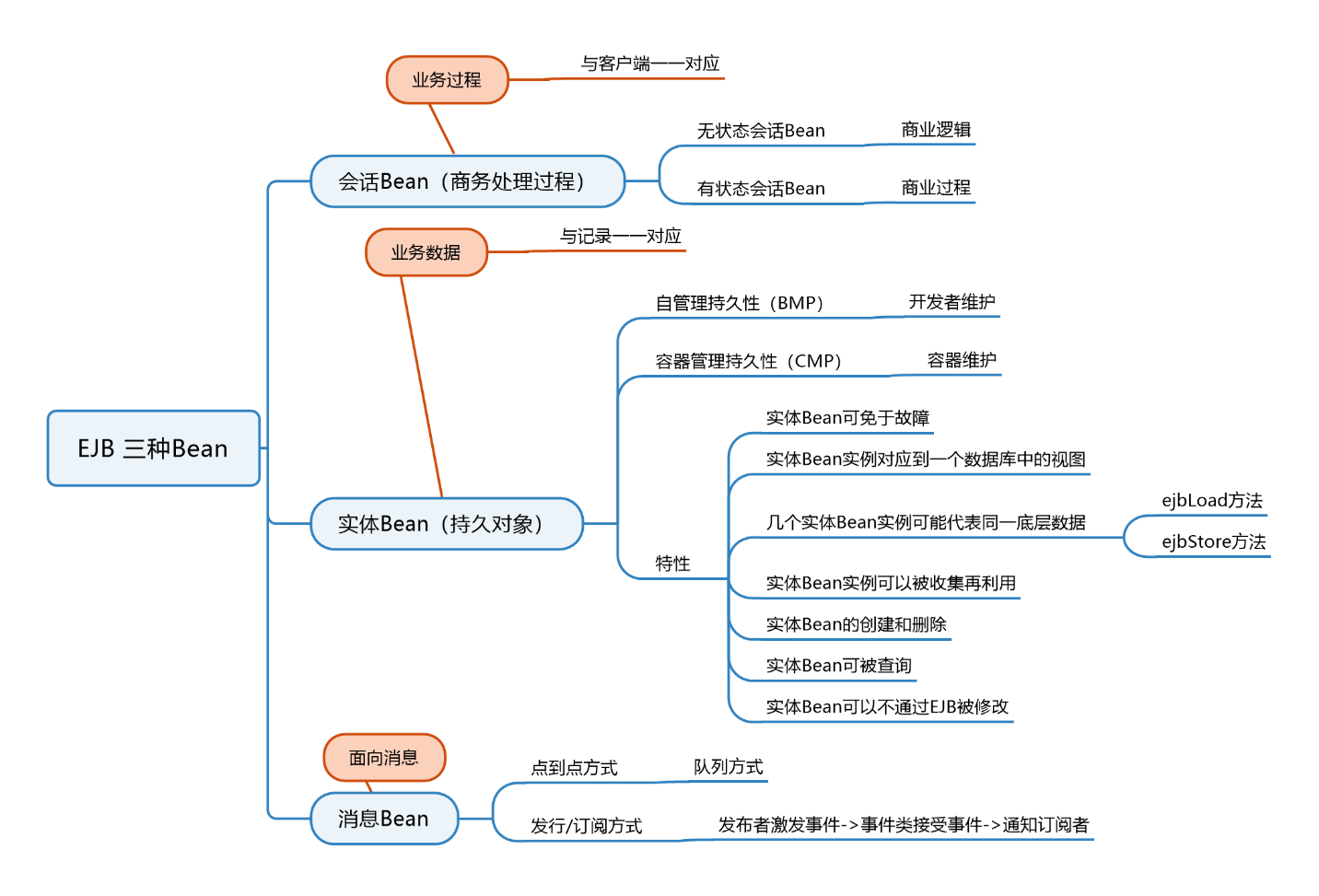


图 4 EJB的三种Bean知识体系

## 客户端使用EJB服务的工作流程

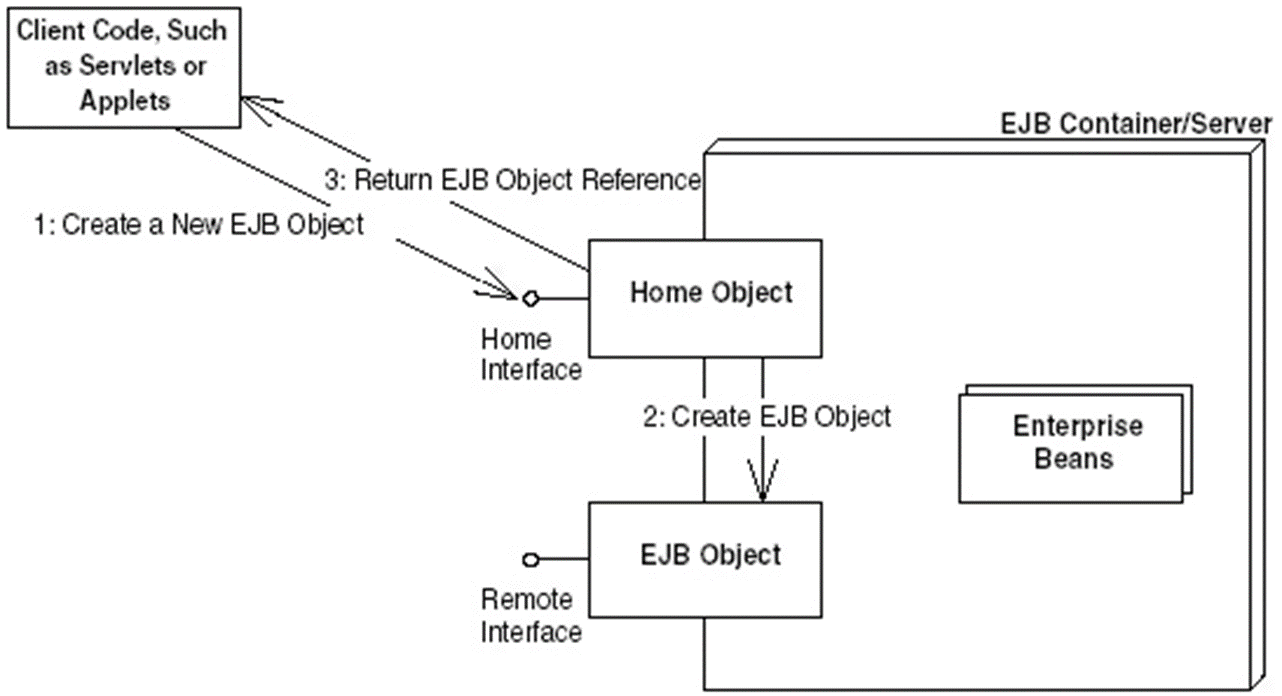


图 5 客户端使用EJB服务的工作流程

## CORBA体系结构



图 6 CORBA体系结构

## X/Open DTP 事务处理过程

①AP首先通过TX接口告诉TM要开始一个新事务；

②TM分配全局事务ID，调用XA接口通知各RM新事务开始；

③AP访问RM；

④操作完成后，AP要求TM提交事务；

⑤TM调用XA接口协调各RM进行事务提交。

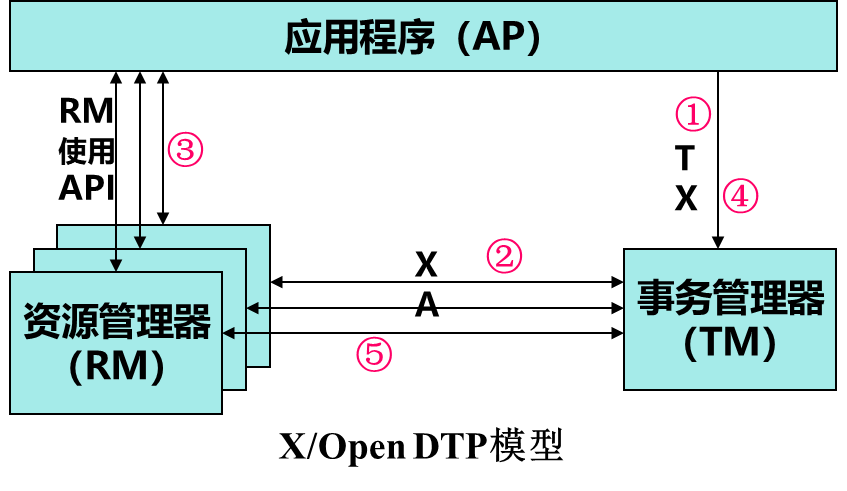


图 7 X/Open DTP 模型