

java认证:采用EJB开发的三个优势Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/583/2021\\_2022\\_java\\_E8\\_AE\\_A4\\_E8\\_AF\\_81\\_c104\\_583999.htm](https://www.100test.com/kao_ti2020/583/2021_2022_java_E8_AE_A4_E8_AF_81_c104_583999.htm) JAVA语言已经慢慢的在成为主流的开发语言之一，或者说现在已经成为了主流的开发语言。在JAVA语言平台上，也出现了多种开发模型。对于刚入门的JAVA程序员来说，也许面对这么多的开发模型，会眼花缭乱，不知道该如何选择。笔者刚开始接触JAVA语言的时候没有多少的开发模型可以选择。而前几年笔者也遇到了这个问题。可选的开发模型比较多，笔者必须选择一个开发模型作为未来自己的主攻方向。因为人的精力是有限的，特别是我们做程序开发的。我们要把有限的精力花在刀口上。笔者在这里向大家推荐EJB开发模型。这个EJB本质上就是一个被管理的组件，存在于J2EE容器中，由J2EE容器进行创建、控制和销毁。J2EE容器复杂控制当前存在的EJB数目和EJB所使用的资源。在重负载的情况下，即使是客户端正在使用的EJB，也将被返回到实例池，如此的话，这个EJB实例还可以供其他客户端使用，从而提高EJB实例的利用率。虽然J2EE官方也是推荐使用EJB，但是这并不是一个强制性的措施。程序开发人员除了利用EJB之外，还可以利用JSP或者单机版的JAVA应用程序等等。但是如果应用程序需要不断的升级、性能要求比较高等等，那么笔者就向大家推荐使用EJB，主要有如下三个方面的原因。一、可以隐藏管道代码。现在音乐喷泉在各地迅速的被采用，成为高科技景观的一个代表之作。程序员在开发这个应用程序的时候，程序人员需要用到这些管道，但是并不需要知道这些水管的具体走向。这不是程序开发人

员所需要关注的内容。程序开发人员之需要直接使用这些现成的管道即可。我们把这些管道就叫做“管道代码”。其实程序开发人员有时候就好像一个工业设计师。工业设计师在设计洗澡用的花撒水笼头的时候，其根本不用关心自来水管。为什么呢？因为自来水管都是采用同一的标准，水压的话也是国家有一个强制性的标准。为此在需要使用管道的时候，设计者之需要直接引用这些标准化的参数即可。在早期的一些开发模型中，如最原始的CORBA开发模型，程序开发人员不得不便写大量的代码来完成同Corba环境的交互、连接、注册过程。其实这些代码就是通常所说的管道代码。而如果采用EJB模型的话则可以最大限度的减少这些管道代码的编写工作。如程序开发人员通过声明属性就可以无需要编写代码来控制这些功能即可指定组件的事务性为.不用通过编写管道代码来定义EJB组件之间的关系以及所需要用到的资源，因为可部署的J2EE应用程序在部署描述信息中定义了多个EJB组件之间的关系同时定义了EJB组件所需要用到的资源.如每个Bean都遵循一个定义的声明周期和一套规则，为此程序开发人员不需要知道“管道”的设计，而只需要知道管道接口的参数即可，如此的话系统代码与应用程序代码之间就是两个互相独立的内容。显然，通过J2EE提供的EJB组件，可以让程序开发人员将精力集中在业务代码的编写上，而尽量减少编写管道代码。这不仅可以提高应用程序的开发效率，而且把管道代码与应用程序代码独立开来，也利于后续的调试与维护。这就是笔者推荐使用EJB模型来开发JAVA应用程序的第一个原因。

二、EJB预定义了一些复杂的处理机制。在应用程序开发的过程中，或多或少有一些共性的内容。如需要

进行应用程序的生命周期管理，需要进行命名和注册，需要进行事务管理等等。如果每次在开发应用程序的时候，都需要从零开始来开发这些功能，那么工作量就会很大，而且代码的重复利用性也会比较差。为了解决这些问题，EJB提供了一些预定义的服务，把一些应用程序开发中要用到的服务集成到J2EE开发环境中。需要用到这些服务的时候，程序开发人员之需要声明一下或者通过少量的代码就可以调用这些服务，实现一些复杂的控制管理机制。如在应用程序开发中，为了保持数据的一致性事务管理机制是必须要实现的一个机制。如果在应用程序层面没有实现事务管理机制的话，则当同一个业务涉及到多条记录的时候，很容易破坏数据的一致性。而如果从零开始来编写事务处理机制代码的话，那么工作量会很大。在EJB的容器服务中就预先提供了事务管理的解决方式，程序开发人员可以凭借这个预定义地解决方案轻松的创建事务、处理与控制事务等等。如在应用程序开发中命名与注册也是很麻烦的一件工作。而EJB也提供另一个命名与注册的容器，EJB容器和服务器为EJB提供了对命名服务的访问。远程和本地客户端使用这些服务来寻找EJB.EJB组件本身也使用这些服务来查询自身所需要的资源。也就好说，程序开发人员在应用程序开发中不用通过代码来实现命名与注册服务，而直接调用EJB组件中的命名与注册容器即可。这个容器会自动生成相关的代码来完成所需要实现的功能。另外，EJB组件还提供了生命周期管理容器、安全性和访问控制容器、持久性容器等等，通过这些容器可以让程序开发人员少写大量的代码，不仅可以提高程序的开发效率，而且同意了这些基础性内容解决方案。这也有利于后来的人员了解源代

码，有利于应用管理软件的后续升级。三、用户接口与底层业务功隔离。在企业管理中共性与个性是并存的，这也体现在了企业的管理软件上。如同一家企业，如果管理者的文化背景不同，其或许多同一个业务具有不同的管理方式。这个用我们程序开发人员专业的术语来讲就是用户接口不同。但是其背后的管理模型是相同的，也就是说其业务功能是相同的。如利用JAVA语言开发的一个订单管理系统，其订单的处理机制是相同的，都在数据库中建立相关的纪录并在保存记录之前进行数据有效性的审核。但是不同的订单类型其处理方式可能稍有不同。如对于预付订单，必须要先收到客户的款项才能够下订单给生产部门安排生产或者仓库部门准备出货。如对于仓库订单，则在流程处理上不需要经过生产而直接转到仓库出货等等。也就说是，10种不同类型的订单，其80%的功能是相同的，而又20%的内容由于管理方式或者其他的原因而有所不同。在这种情况下难道要写十个不同的代码来实现这十种不同的需求吗？在EJB开发模型中不用这么复杂，因为EJB允许独立于表达层开发和部署业务功能。如上面这个订单管理需求，程序开发人员可以利用EJB模型来实现底层的功能(80%的共性内容)，然后再无需重新设计或者开发整个应用程序或者销售订单管理模块的情况下，可以利用不同的用户接口来实现用户的不同需求。这就好像父母与子女的关系。现把父母的特性定义好，然后再根据不同的需要生养不同的子女即可(用户接口)。由于子女继承了父母的全部特性。那么只需要把用户需要实现的一些个性特点嫁接到子女身上即可。所以这种业务需求与业务功能相分离，各自独立的特征，是EJB开发模型的最大优势。程序开发人员可以利用EJB

实现分布式应用程序，将用户接口与底层业务功能隔离开来。更多优质资料尽在百考试题论坛 百考试题在线题库 java认证更多详细资料 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)