

Technical Papers Java Questions: Satyam JAVA Test on Apr 2005

Examrace Placement Series prepares you for the toughest placement exams to top companies.

Interview Question at Satyam for JAVA platform holding 3 yrs of Exp

1. What is difference between Stateful and Stateless Session Bean? A Stateful Session Bean is a bean that is designed to service business processes that span multiple method requests or transactions. Stateful Session beans retain state on behalf of an individual client. Stateless Session Beans do not maintain state. EJB containers pools stateless session beans and reuses them to service many clients. Stateful session beans can be passivated and reused for other clients. But this involves I/O bottlenecks. Because a stateful session bean caches client conversation in memory, a bean failure may result in losing the entire client conversation. Therefore, while writing a stateful session bean the bean developer has to keep the bean failure and client conversation loss in mind. In case of stateless session beans, client specific data has to be pushed to the bean for each method invocation which will result in increase in the network traffic. This can be avoided in a number of ways like persisting the client specific data in database or in JNDI. But this also results in I/O performance bottlenecks. If the business process spans multiple invocations thereby requiring a conversation then stateful session bean will be the ideal choice. On the other hand, if business process lasts only for a single method call, stateless session bean model suits. Stateful session beans remembers the previous request and responses. But stateless beans do not. Stateful does not have pooling concept, whereas the stateless bean instances are pooled
2. What is difference between BeanManagedPersistence and ContainerManagedPersistence?
CMP: Tx behaviour in beans are defined in transaction attributes of the methods
BMP: Programmers has to write a code that implements Tx behaviour to the bean class. Tuned CMP entity beans offer better performance than BMP entity beans. Moving towards the CMP based approach provides database independence since it does not contain any database storage APIs within it. Since the container performs database operations on behalf of the CMP entity bean, they are harder to debug. BMP beans offers more control and flexibility than CMP beans.
Diff 1: In BMP you will take care of all the connection and you write the SQL code inside the bean whereas in CMP the container will take care of it
Diff 2: The BMP is not portable across all DB's. Whereas the CMP is
3. Draw and explain MVC architecture? MVC Architecture is Module-View-Controller Architecture. Controller is the one which controls the flow of application/services the

requests from the View. Module is the other layer which performs the exact operations. Each layer should be loosely coupled as much as possible.

4. Difference between forward (request, response) and SendRedirect (url) in Servlet? With Forward, request & response would be passed to the destination URL which should be relative (means that the destination URL should be within a servlet context). Also, after executing forward method, the control will return back to the same method from where the forward method was called. All the opposite to the above points apply to sendRedirect (OR). The forward will redirect in the application server itself. It does not come to the client. Whereas Response. Sendredirect () will come to the client and go back ... Ie. URL appending will happen.
5. What is Synchronise? Synchronize is a technique by which a particular block is made accessible only by a single instance at any time (OR). When two or more objects try to access a resource, the method of letting in one object to access a resource is called sync
6. How to prevent Dead Lock? Using synchronization mechanism. For Deadlock avoidance use Simplest algorithm where each process tells max number of resources it will ever need. As process runs, it requests resources but never exceeds max number of resources. System schedules processes and allocates resources in a way that ensures that no deadlock results.
7. Explain different way of using thread? The thread could be implemented by using Runnable interface or by inheriting from the Thread class. The former is more advantageous, because when you are going for multiple inheritance. The only interface can help
8. what are pass by reference and passby value? Pass By Reference means the passing the address itself rather than passing the value. Passby Value means passing a copy of the value to be passed.
9. How Servlet Maintain Session and EJB Maintain Session? Servlets maintain session in ServletContext and EJB's in EJBContext.
10. Explain DOM and SAX Parser? DOM parser is one which makes the entire XML passed as a tree Structure and will have it in memory. Any modification can be done to the XML. SAX parser is one which triggers predefined events when the parser encounters the tags in XML. Event-driven parser. Entire XML will not be stored in memory. Bit faster than DOM. NO modifications can be done to the XML.
11. What is HashMap and Map? Map is Interface and Hashmap is class that implements that and its not serialized HashMap is non serialized and Hashtable is serialized
12. Difference between HashMap and HashTable? The HashMap class is roughly equivalent to Hashtable, except that it is unsynchronized and permits nulls (HashMap allows null values as key and value whereas Hashtable doesnt allow). HashMap does not guarantee that the order of the map will remain constant over time. Difference between Vector and ArrayList? Vector is serialized whereas arraylist is not

13. Difference between Swing and Awt? AWT are heavy-weight componenets. Swings are light-weight components. Hence swing works faster than AWT.
14. Explain types of Enterprise Beans? Session beans → Associated with a client and keeps states for a client Entity Beans → Represents some entity in persistent storage such as a database
15. What is enterprise bean? Server side reusable java component Offers services that are hard to implement by the programmer Sun: Enterprise Bean architecture is a component architecture for the deployment and development of component-based distributed business applications. Applications written using enterprise java beans are scalable, transactional and multi-user secure. These applications may be written once and then deployed on any server plattform that supports enterprise java beans specification. Enterprise beans are executed by the J2EE server. First version 1.0 contained session beans, entity beans were not included. Entity beans were added to version 1.1 which came out during year 1999. Current release is EJB version 1.2
16. Services of EJB? Database management: Database connection pooling DataSource, offered by the J2EE server. Needed to access connection pool of the server. Database access is configured to the J2EE server → easy to change database/database driver Transaction management: Distributed transactions J2EE server offers transaction monitor which can be accessed by the client. Security management: Authetication Authorization encryption Enterprise java beans can be distributed/replicated into separate machines Distribution/replication offers Load balancing, load can be divided into separate servers. Failover, if one server fails, others can keep on processing normally. Performance, one server is not so heavy loaded. Also, for example Weblogic has thread pools for improving performance in one server.
17. When to choose EJB? Server will be heavy loaded: Distribution of servers helps to achieve better performance. Server should have replica for the case of failure of one server: Replication is invisible to the programmer Distributed transactions are needed J2EE server offers transaction monitor that takes care of transaction management. Distributed transactions are invisible to the programmer Other services vs. Money: Weblogic J2EE server ~ 80 000 mk and Jbuilder X Professional Edition ~ 5 000mk
18. Why not to use free J2EE servers? no tecnical support harder to use (no graphical user interface ...) no integration to development tools (for example, Jbuilder) Bugs? Other problems during project?
19. Alternative: Tuxedo Tuxedo is a middleware that offers scalability services and transaction monitors. C or C + + based. Can be used with Java client by classes in JOLT package offered by BEA. Faster that J2EE server? Harder to program? Harder to debug? Implementation is platform dependent.

20. J2EE server offers DataSource: Object that can be used to achieve database connection from the connection pool. Can be accessed by the interface DataSource Transaction monitor: Can be accessed by the interface UserTransaction. Java Naming and the Directory Service:
21. Java Naming and the Directory Service Naming service is needed to locate beans home interfaces or other objects (DataSource, UserTransaction): For example, jndi name of the DataSource Directory service is needed to store and retrieve properties by their name: Jndi name: Java: Comp/env/propertyName
22. XML deployment descriptor ejb-jar. Xml + server-specific xml-file Which is then Packed in a jar file together with bean classes. Beans are packaged into EJB JAR file, Manifest file is used to list EJB's and jar file holding Deployment descriptor.
23. Session Bean Developer programs three classes: Home interface, contains methods for creating (and locating for entity beans) bean instances. Remote interface, contains business methods the bean offers. Bean class, contains the business logic of the enterprise bean.
24. Entity Beans Represents one row in the database: Easy way to access database business logic concept to manipulate data. Container managed persistence vs. Bean managed persistence: Programmer creates three or four classes: Home interface for locating beans Remote interface that contains business methods for clients. Bean class that implements bean's behaviour. Primary key class that represents primary key in the database. Used to locate beans. Primary key class is not needed if primary key is a single field that could
25. When to use which bean? Entity beans are effective when application wants to access one row at a time. If many rows needs to be fetched, using session beans can be better alternative ava class (for example, Integer). Entity beans are efficient when working with one row at a time Cause a lot of network traffic. Session Beans are efficient when client wants to access database directry. Fetching/updating multiple rows from the database
26. Explain J2EE Arch?

Normally, thin-client multitiered applications are hard to write because they involve many lines of intricate code to handle transaction and state management, multithreading, resource pooling, and other complex low-level details. The component-based and platform-independent J2EE architecture makes J2EE applications easy to write because business logic is organized into reusable components and the J2EE server provides underlying services in the form of a container for every component type. Because you do not have to develop these services yourself, you are free to concentrate on solving the business problem in hand.

Containers and Services: Component are installed in their containers during deployment and are the interface between a component and the low-level platform-specific functionality that supports the component. Before a web, enterprise bean, or application client component can be executed, it must be assembled into a J2EE application and deployed into its container.

Visit examrace.com for free study material, doorsteptutor.com for questions with detailed explanations, and "Examrace" YouTube channel for free videos lectures

The assembly process involves specifying container settings for each component in the J2EE application and for the J2EE application itself. Container settings customize the underlying support provided by the J2EE Server, which include services such as security, transaction management, Java Naming and Directory Interface™ (JNDI) lookups, and remote connectivity.

Container Types: The deployment process installs J2EE application components in the following types of J2EE containers. The J2EE components and container addressed in this tutorial are shown in Figure 5.

An Enterprise JavaBeans (EJB) container manages the execution of all enterprise beans for one J2EE application. Enterprise beans and their container run on the J2EE server.

A web container manages the execution of all JSP page and servlet components for one J2EE application. Web components and their container run on the J2EE server.

An application client container manages the execution of all application client components for one J2EE application. Application clients and their container run on the client machine.

An applet container is the web browser and Java Plug-in combination running on the client machine.