

Java / J2EE Technical Lead - Interview Questions and Answers

 buggybread.com/2015/02/java-j2ee-technical-lead-interview.html

Q1. What is meant by an "asynchronous event." Give some examples ?

Ans. An asynchronous event is one that occurs at an unpredictable time outside the control of the program that the CPU is running. It is not "synchronized" with the program.

Q2. What are the benefits of using Spring Framework ?

Ans. Spring enables developers to develop enterprise-class applications using POJOs. The benefit of using only POJOs is that you do not need an EJB container product.

Spring is organized in a modular fashion. Even though the number of packages and classes are substantial, you have to worry only about ones you need and ignore the rest.

Spring does not reinvent the wheel instead, it truly makes use of some of the existing technologies like several ORM frameworks, logging frameworks, JEE, Quartz and JDK timers, other view technologies.

Testing an application written with Spring is simple because environment-dependent code is moved into this framework. Furthermore, by using JavaBean-style POJOs, it becomes easier to use dependency injection for injecting test data.

Spring's web framework is a well-designed web MVC framework, which provides a great alternative to web frameworks such as Struts or other over engineered or less popular web frameworks.

Spring provides a convenient API to translate technology-specific exceptions (thrown by JDBC, Hibernate, or JDO, for example) into consistent, unchecked exceptions.

Lightweight IoC containers tend to be lightweight, especially when compared to EJB containers, for example. This is beneficial for developing and deploying applications on computers with limited memory and CPU resources.

Spring provides a consistent transaction management interface that can scale down to a local transaction

Q3. Why Struts 1 Classes are not Thread Safe whereas Struts 2 classes are thread safe ?

Ans. Struts 1 actions are singleton. So all threads operates on the single action object and hence makes it thread unsafe.

Struts 2 actions are not singleton and a new action object copy is created each time a new action request is made and hence its thread safe.

Q4. Name few tools for probing Java Memory Leaks ?

Ans. JProbe, Optimizelt

Q5. Difference between SAX and DOM Parser ?

Ans. A DOM (Document Object Model) parser creates a tree structure in memory from an input document whereas A

SAX (Simple API for XML) parser does not create any internal structure.

A SAX parser serves the client application always only with pieces of the document at any given time whereas A DOM parser always serves the client application with the entire document no matter how much is actually needed by the client.

A SAX parser, however, is much more space efficient in case of a big input document whereas DOM parser is rich in functionality.

Use a DOM Parser if you need to refer to different document areas before giving back the information. Use SAX is you just need unrelated nuclear information from different areas.

Xerces, Crimson are SAX Parsers whereas XercesDOM, SunDOM, OracleDOM are DOM parsers.

Q6. What are LDAP servers used for ?

Ans. LDAP servers are typically used in J2EE applications to authenticate and authorise users. LDAP servers are hierarchical and are optimized for read access, so likely to be faster than database in providing read access.

Q7. Explain Flow of Spring MVC ?

Ans. The DispatcherServlet configured in web.xml file receives the request.

The DispatcherServlet finds the appropriate Controller with the help of HandlerMapping and then invokes associated Controller.

Then the Controller executes the logic business logic and then returns ModelAndView object to the DispatcherServlet.

The DispatcherServlet determines the view from the ModelAndView object.

Then the DispatcherServlet passes the model object to the View.

The View is rendered and the Dispatcher Servlet sends the output to the Servlet container. Finally Servlet Container sends the result back to the user.

Q8. What is session tracking and how do you track a user session in servlets?

Ans. Session tracking is a mechanism that servlets use to maintain state about a series requests from the same user across some period of time. The methods used for session tracking are:

User Authentication - occurs when a web server restricts access to some of its resources to only those clients that log in using a recognized username and password

Hidden form fields - fields are added to an HTML form that are not displayed in the client's browser. When the form containing the fields is submitted, the fields are sent back to the server

URL rewriting - every URL that the user clicks on is dynamically modified or rewritten to include extra information. The extra information can be in the form of extra path information, added parameters or some custom, server-specific URL change.

Cookies - a bit of information that is sent by a web server to a browser and which can later be read back from that browser.

HttpSession- places a limit on the number of sessions that can exist in memory.

Q9. What is connection pooling?

Ans. It's a technique to allow multiple clients to make use of a cached set of shared and reusable connection objects providing access to a database or other resource.

Q10. How Java provide high Performance ?

Ans. Java uses Just-In-Time compiler to enable high performance. Just-In-Time compiler is a program that turns Java bytecode into instructions that can be sent directly to the processor.

Q11. What things should be kept in mind while creating your own exceptions in Java?

Ans. All exceptions must be a child of Throwable.

If you want to write a checked exception that is automatically enforced by the Handle or Declare Rule, you need to extend the Exception class.

You want to write a runtime exception, you need to extend the RuntimeException class.

Q12. What are RESTful Web Services ?

Ans. REST or Representational State Transfer is a flexible architecture style for creating web services that recommends the following guidelines -

1. http for client server communication,
2. XML / JSON as formatiing language ,
3. Simple URI as address for the services and,
4. stateless communication.

Q13. Which markup languages can be used in restful web services ?

Ans. XML and JSON (Javascript Object Notation).

Q14. What is database deadlock ? How can we avoid them?

Ans. When multiple external resources are trying to access the DB locks and runs into cyclic wait, it may makes the DB unresponsive.

Deadlock can be avoided using variety of measures, Few listed below -

Can make a queue wherein we can verify and order the request to DB.

Less use of cursors as they lock the tables for long time.

Keeping the transaction smaller.

Q15. Difference between first level and second level cache in hibernate ?

Ans. 1. First level cache is enabled by default whereas Second level cache needs to be enabled explicitly.

2. First level Cache came with Hibernate 1.0 whereas Second level cache came with Hibernate 3.0.

3. First level Cache is Session specific whereas Second level cache is shared by sessions that is why First level cache is considered local and second level cache is considered global.

Q16. What are the advantages of Hibernate ?

Ans. 1. No need to know SQL, RDBMS, and DB Schema.

2. Underlying Database can be changed without much effort by changing SQL dialect and DB connection.

3.Improved Performance by means of Caching.

Q17. Is It Good to use Reflection in an application ? Why ?

Ans. no, It's like challenging the design of application.

Q18. What is Lazy Initialization in Hibernate ?

Ans. It's a feature to lazily initialize dependencies , relationship and associations from the Database. Any related references marked as @OneToMany or @ManyToMany are loaded lazily i.e when they are accessed and not when the parent is loaded.

Q19. What are new features introduced with Java 8 ?

Ans. Lambda Expressions , Interface Default and Static Methods , Method Reference , Parameters Name , Optional , Streams, Concurrency.

Q20. What things you would care about to improve the performance of Application if its identified that its DB communication that needs to be improved ?

Ans. 1. Query Optimization (Query Rewriting , Prepared Statements)

2. Restructuring Indexes.

3. DB Caching Tuning (if using ORM)

4. Identifying the problems (if any) with the ORM Strategy (If using ORM)

Q21. Name few Java Mocking frameworks ?

Ans. Mockito, PowerMock, EasyMock, JMock, JMockit

Q22. Should we create system software (e.g Operating system) in Java ?

Ans. No, Java runs on a virtual machine called JVM and hence doesn't embed well with the underlying hardware. Though we can create a platform independent system software but that would be really slow and that's what we would never need.

Q23. What is the difference between JPA and Hibernate ?

Ans. JPA or Java Persistence API is a standard specification for ORM implementations whereas Hibernate is the actual ORM implementation or framework.

Q24. Which UML diagrams you usually use for design ?

Ans. Use Case Diagram, Component Diagram for High level Design and Class Diagram , Sequence Diagram for low level design.

Q25. How do you coordinate and communicate with the team developers ?

Ans. We as a team of developers , testers , analyst , lead and architect sit close to each other. Most of the time I would just jump to their seat and talk to them (if required). We have daily stand up where we discuss things that needs team attention.

Q26. What kind of software architecture your organization follow ?

Ans. We have multi tier architecture with multiple layers , We have series of web servers and applications in application tier, infrastructure libraries at middle tier and Database servers at the lower tier. We are using Oracle as Database, ESB (Enterprise service Bus) for asynchronous communication and Rest Web Services.

Q27. What Design pattern Wrapper Classes implement ?

Ans. Adapter.

Q28. Difference between Factory and Abstract Factory Design Pattern ?

Ans. Factory Pattern deals with creation of objects delegated to a separate factory class whereas Abstract Factory patterns works around a super-factory which creates other factories.

Q29. Difference between Factory and Builder Design Pattern ?

Ans. Builder pattern is the extension of Factory pattern wherein the Builder class builds a complex object in multiple steps.

Q30. Difference between Proxy and Adapter ?

Ans. Adapter object has a different input than the real subject whereas Proxy object has the same input as the real subject. Proxy object is such that it should be placed as it is in place of the real subject.

Q31. Difference between Adapter and Facade ?

Ans. The Difference between these patterns in only the intent. Adapter is used because the objects in current form cannot communicate where as in Facade , though the objects can communicate , A Facade object is placed between the client and subject to simplify the interface.

Q32. Difference between Builder and Composite ?

Ans. Builder is a creational Design Pattern whereas Composite is a structural design pattern. Composite creates

Parent - Child relations between your objects while Builder is used to create group of objects of predefined types.

Q33. Example of Chain of Responsibility Design Pattern ?

Ans. Exception Handling Throw mechanism.

Q34. Example of Observer Design Pattern ?

Ans. Listeners.

Q35. Difference between Factory and Strategy Design Pattern ?

Ans. Factory is a creational design pattern whereas Strategy is behavioral design pattern. Factory revolves around the creation of object at runtime whereas Strategy or Policy revolves around the decision at runtime.

Q36. Shall we use abstract classes or Interfaces in Policy / Strategy Design Pattern ?

Ans. Strategy deals only with decision making at runtime so Interfaces should be used.

Q37. What are use cases?

Ans. It is part of the analysis of a program and describes a situation that a program might encounter and what behavior the program should exhibit in that circumstance.

Q38. Explain Singleton Design Pattern ?

Ans. <http://www.buggybread.com/2014/03/java-design-pattern-singleton-interview.html>

Q39. How to implement an immutable class ?

Ans. We can make a class immutable by

1. Making all methods and variables as private.
2. Setting variables within constructor.

```
Public Class ImmutableClass{  
    private int member;  
    ImmutableClass(int var){  
        member=var;  
    }  
}
```

and then we can initialize the object of the class as

```
ImmutableClass immutableObject = new ImmutableClass(5);
```

Now all members being private , you can't change the state of the object.

Q40. What is the best practice configuration usage for files in Maven - pom.xml or settings.xml ?

Ans. The best practice guideline between settings.xml and pom.xml is that configurations in settings.xml must be specific to the current user and that pom.xml configurations are specific to the project.

Q41. How substring() method of String class create memory leaks?

Ans. substring method would build a new String object keeping a reference to the whole char array, to avoid copying it. Hence you can inadvertently keep a reference to a very big character array with just a one character string.

Q42. Difference between first level and second level cache in hibernate ?

Ans. 1. First level cache is enabled by default whereas Second level cache needs to be enabled explicitly.

2. First level Cache came with Hibernate 1.0 whereas Second level cache came with Hibernate 3.0.

3. First level Cache is Session specific whereas Second level cache is shared by sessions that is why First level cache is considered local and second level cache is considered global.

Q43. What are different types of second level cache ?

Ans. 1. EHCACHE (Easy Hibernate)

2. OSCache (Open Symphony)

3. Swarm Cache (JBoss)

4. Tree Cache (JBoss)

Q44. What are new features introduced with Java 8 ?

Ans. Lambda Expressions , Interface Default and Static Methods , Method Reference , Parameters Name , Optional , Streams, Concurrency.

Q45. What things you would care about to improve the performance of Application if its identified that its DB communication that needs to be improved ?

Ans. 1. Query Optimization (Query Rewriting , Prepared Statements)

2. Restructuring Indexes.

3. DB Caching Tuning (if using ORM)

4. Identifying the problems (if any) with the ORM Strategy (If using ORM)

Q46. If I try to add Enum constants to a TreeSet, What sorting order will it use ?

Ans. Tree Set will sort the Values in the order in which Enum constants are declared.

Q47. Why do we need Inner classes ? Can't we just work with outer classes wherever we implement Inner classes ?

Ans. Yes, we can substitute outer classes wherever we need to have inner classes but Inner classes have advantage in certain cases and hence preferred -

Ease - Why to implement a class outside if its objects are only intended to be part of an outer object. Its easy to define the class within another class if the use is only local.

Protection - Making a call an outer exposes a threat of it being used by any of the class. Why should it be made an outer class if its object should only occur as part of other objects.

For example - You may like to have an class address whose object should have a reference to city and by design that's the only use of city you have in your application. Making Address and City as outer class exposes City to any of the Class. Making it an inner class of Address will make sure that its accessed using object of Address.

Q48. What things you will look for if you get following exception while making DB call ?

table or view does not exist

Ans. First will check if the table or view actually exist in the DB

If it does , Will make sure to see that the application has rights on the schema that holds the respective Table.

Will then make sure that we have prefixed the schema with the table name while accessing it.

Will then make sure that its not DB Cache that's causing it as the table DDL might have been created recently.

Q49. What is a transitive dependency in Maven ? Can we override Transitive Dependency version and If Yes, how ?

Ans. Transitive dependency is the dependencies not defined directly in the current POM but the POM of the dependent projects.

Yes we can override transitive dependency version by specifying the dependency in the current POM.

Q50. Which UML diagrams you usually use for design ?

Ans. Use Case Diagram, Component Diagram for High level Design and Class Diagram , Sequence Diagram for low level design.

Q51. How do you coordinate and communicate with the team developers ?

Ans. We as a team of developers , testers , analyst , lead and architect sit close to each other. Most of the time I would just jump to their seat and talk to them (if required). We have daily stand up where we discuss things that needs team attention.

Q52. What kind of software architecture your organization follow ?

Ans. We have multi tier architecture with multiple layers , We have series of web servers and applications in application tier, infrastructure libraries at middle tier and Database servers at the lower tier. We are using Oracle as Database, ESB (Enterprise service Bus) for asynchronous communication and Rest Web Services.

Q53. Which web application framework you like and Why ?

Ans. I like Spring as it comes with inbuilt Dependency Injection framework. It has great online community and support and is proven to work well with ORMs like Hibernate. If we are not working with ORM and DI, Struts 2 is also good.

