Top 10 Interview Q&A for Java Lead Position

-By SivaReddy

1) **What are SOLID design principles?**
   - Single Responsibility Principle : One class should have one and only one responsibility
   - Open Closed Principle : Software components should be open for extension, but closed for modification
   - Liskov's Substitution Principle: Derived types must be completely substitutable for their base types
   - Interface Segregation Principle: Clients should not be forced to implement unnecessary methods which they will not use
   - Dependency Inversion Principle: Depend on abstractions, not on concretions

   In simple Terms, If you want to remember

   - Reusability
   - Upgradability with Zero or minimal downtime
   - Scalability
   - Easy for Maintenance
   - Loose coupling

2) **What are the advantages and Disadvantages of Junit?**
   Advantages:
   - Problems are discovered early.
   - Integration testing is easier.
   - Acts as documentation of the system.
   - Extremely accurate unit test coverage.
   - The unit test itself is used to verify the design

   Disadvantages:

   - Time consuming.
   - Hard to set up realistic, useful tests.
   - Need to have a review process for test case failures.
   - Test program cannot be run in actual deployment environment.
   - Test code is likely to be at least as buggy as the code it is testing.
   - Value and accuracy of unit tests can be diminished if initial conditions are not set correctly
3) **Did you use any caching framework in your applications? What are the advantages and disadvantages of it?**

   Caching is a technique of storing frequently useddata/information in memory, so that, when the samedata/information is needed next time, it could be directly retrieved from the memory instead of being generated by the application

Different Open Source Cache Frameworks: Java Caching System (JCS),OSCache,Java Object Cache (JOCache),SwarmCache,JBossCache,IronEye Cache, EHCache, Infinispan,

What is the Data structure implemented for Caching?
HashMap/LinkedHashMap/ConcurrentHashMap

Advantages and Disadvantages of Caching

Two types of cache exist in Java development: local cache and cluster cache.

Local cache

- **Advantages:**
- Data is directly taken from Java Virtual Machine (JVM)
- Access speed is very fast
- Uses very low threshold
- Well-developed plug-ins like ehcache, oscache, and guava cache are easily available in the market. (The ehcache and oscache already provide the cluster cache.)

Disadvantages:

- Inconsistency and disparity problems may occur between different machines in the server cluster
- When the local cache is too large, it is easy to trigger Garbage Collection (GC) of servers

Cluster cache

Advantages:

- The data consistency is guaranteed, and the capacity expansion is convenient, transparent, and unperceivable to users.
- It engages more mature middleware tools like Tair, Redis, and Memcached.

Disadvantages:

- You need to select mature and business-complying cluster caching middleware and cluster framework to establish and maintain a set of cache cluster servers
- The threshold for use is high, and cache hotspot issues may occur, like the Tair hotspot issues.

4) What is TDD and its advantages and disadvantages?

- TDD is a software development methodology whereby you write and run a set of tests before you write code.

- The idea is that those tests will fail at first and then you start to write enough code to try to get all the tests to pass. Having all the tests pass could be a measure of the done criteria (dev-done) and also increases confidence in the quality of the code.

Advantages:

- TDD helps you learn, understand, and internalize the key principles of good modular design.
- TDD also forces good architecture
- Helps prevents defects
- Helps programmers really understand their code
- It helps to clarify requirements because you have to figure out concretely what inputs you have to feed and what outputs you expect.

Disadvantages:

- Hard to apply to existing legacy code
- Hard to start working this way. Especially if you have many years of working the other way.
- You sometimes have to mock a lot of things or things that are difficult to mock.
- Unless everyone on the team correctly maintains their tests, the whole system can quickly degrade.

5) **What are the additional responsibilities you take as Lead?**
    - Involved in Business requirements gathering from client
    - Preparing the use case, Class, activity, sequence diagrams, high level and low-level design documents, Review the design with the architects
    - Provide technical guidance to Team members
    - Sprint Planning, Functionalities break down into tasks, assigned to Scrum team, Code review and mark sure team follow the best practices to maintain the overall system quality
    - Adopt the tools like SonarQube, Jenkins , Junit, Mock tool to optimize the development efforts
6) **How do you onboard new team member joining into your team/project?**
    - Prepare the KT(Knowledge Transition) plan
    - Provide Technical design documents, SOP, setup the Local development environment
    - KT on the Project overview, technical designs and provide all the guidance to accustom to the environment
    - If there are tools specific to the project and candidate does not know about it, then provide the basic training on the modules to pick up things his/her own
    - Give some exercise takes to understand the project better
7) **What is the approach do you follow to fix the production blocker issue?**
    - Identify the application server logs and find the exception/error details
    - Identify the Java class and line of the code having the issue
    - Troubleshoot the code to find RCA, it might may be due to corrupted data or code did not handle to catch the exception like Null pointer exception, ArrayIndexOutOfBoundException etc..
    - In case of error due to OutOfMemoryError then increase the heap Size or monitor the memory consumption using tools like JConsole/JMC

8) **Do you have experience in Application Support, Ticket tracking tool and how do you resolve issues?**
   - ServiceNow, Bugzilla, JIRA, TRSM are tools used to track Issues or Change Requests.
   - There SLAs based on severity of the issue, P0, P1, P2, P3,P4, Importance : Blocker/Critical/Medium/Normal/Low
   - In supporting applications, communication plays key role while resolving the issue

9) **What is agile process and how did you implement agile in your project?**
   - Daily Scrum meetings
   - Sprint Planning (Product Owner, Scrum Team)
   - Sprint execution(Implementation)
   - Sprint Introspection
   - Sprint Closure

10) **What is best framework you ever worked on and how do you choose which framework suitable for the application and What critical/Complex issue you encountered and how do you resolve?**

ALL THE BEST