**Real Time Java Interview Questions and Answers**

1. JVM Architecture?

The JVM (Java Virtual Machine) architecture is the underlying infrastructure that enables Java programs to be executed on different hardware and operating systems. It provides a layer of abstraction between the Java application and the underlying system, allowing Java programs to be platform-independent and portable. Here's an overview of the JVM architecture:

**1. Class Loader Subsystem:**

- **Class Loader:** Responsible for loading Java class files into memory. It performs tasks like locating and loading class files from the file system, network, or other sources.

- **Bytecode Verifier:** Verifies the integrity and correctness of bytecode during the loading process to ensure that it doesn't violate any security or structural constraints.

**2. Runtime Data Area:**

**- Method Area:** Stores class-level data, including metadata, constant pool, field and method information, and static variables.

**- Heap:** Memory area where objects are allocated. It is divided into young generation (Eden space, Survivor spaces) and old generation (Tenured space) regions, managed by garbage collection.

- Java Stack: Each thread in a Java program has its own stack, which stores method frames containing local variables, operand stacks, and other data related to method execution.

- Native Method Stack: Similar to the Java stack, but specific to native (non-Java) method execution.

- PC (Program Counter) Registers: Each thread has its own PC register, which stores the address of the currently executing instruction.

3. Execution Engine:

- Just-In-Time (JIT) Compiler: Translates bytecode into native machine code on the fly, optimizing performance by analyzing and compiling frequently executed sections of code.

- Interpreter: Executes bytecode line by line when the code is first encountered. It's used initially before the JIT compiler has optimized and compiled the code.

4. Native Method Interface (JNI):

- Allows Java code to interact with native code written in languages such as C or C++. It provides a way to call native methods and access native libraries.

5. Native Method Libraries:

- Contains the native libraries required by the JVM to provide platform-specific functionality and access to the underlying operating system.

6. Java Native Interface (JNI) Libraries:

- Provide a standard set of native methods that enable Java programs to perform tasks not directly supported by the Java language, such as accessing hardware devices or interacting with the operating system.

The JVM architecture allows Java programs to be executed in a controlled and secure environment. It manages memory, provides automatic garbage collection, and offers various optimization techniques to improve performance. Additionally, it ensures platform independence by abstracting the underlying hardware and operating system details, making Java applications highly portable across different environments.



2. What is platform independence, why java is called platform independent?

3. What is STREAM API in java 8? Explain methods? How to sort employee with age > 30

4. Explain Overriding and Overloading?

5. Explain Collection API? diff HashMap and LinkdedHashMap, TreeMap, HashSet

6. Explain internal workings of HashMap? What is the DS used?

7. Explain Exception Handling?

8. difference between final, finally, finalize?

9. What is Garbage Collector in Java?

10. What is Abstract?

11. What is Interface?

12. What is static variable?

13. What is instance variable?

14. Why we need empty for loop? explain reason

15. Multithreads in java

16. What is Dependency in Injection? types of DI

17. Explain Spring MVC Model?

18. How to connect from spring to database?

19. List out and Explain Annotations in Spring boot?

20. Difference between String, StringBuffer, StringBuilder?

21. How to reverse a string without using inbuilt reverse method?

22. Swapping of two variables without using third variable?

23. What is super class method throws IOException sub class method thows Exception in overriding?

24. Access specifiers? in case of method overriding

25. Difference between @Component and @RequestMapping

26. Difference between exit and break?

27. Explain Servlet methods?

28. Servlet vs JSP?

29. What if class implement two interfaces that are having same method name called display?

30. Explain HTTP methods GET, POST, PUT and DELETE?

31. Explain Wrapper classes?

32. What is the need of interfaces?

33. Explain try with resources?

34. How to change the port number in spring?

35. Spring Modules?

36. Why is Hibernate?

37. How to connect to a DB from Hibernate? What is the file used?

38. How to create custom immutable class in java?

39. Why Spring Boot?

40. What is Spring Data JPA? Named query

41. What are the advantages of functional interfaces in java?

42. Difference between == and equals method

43. Stream API how to sort employee with age > 30

JVM Architecture?

44. How HashMap internally works? What is the data structure used?

**IRIS**

1. What are all the annotations used in spring ?
2. @qulifier annotation
3. Stream api adding an element and removing the last character from that element and storing it inside the new list
4. Hibernate get and load method? If its not available What exception we get?
5. What is CuncurrentModificationException?
6. To avoid that which collection class we need to use?
7. What is polymorphism?
8. What is mokito?
9. How to get second highest salary in SQL

JVM stands for Java Virtual Machine.​

JVM is the virtual machine on which Java code executes.​

JVM is responsible for converting byte code into machine specific code.​

Method Area - Method area stores data for each and every class like fields, constant pool, method's data and information​

Heap - Heap is place where all objects are stored in JVM​

Java Threads (Java thread Stacks) - Whenever new method is called new stack frame is created and it is pushed on top of that thread's stack.​

PC Registers - the address of instructions currently and next address being executed.​

Native Method Stack - For every thread, a separate native stack is created. It stores native method information.

Stream API?

Roles and responsibilities in current project?

Difference between Delete and truncate?

Difference between view and materialized views?

Difference between procedure and functions?

Can we call procedure inside functions?

Constraints and it's types?

Triggers and it's types?

Can we recover the data after truncate?

What is foreign key?

Cursors and it's types?

Write a procedure to calculate nu.of experience of each employees?

Which are date functions which you know?

What are the collection types in Oracle? And differentiate it

What is bulk collection

Can we use commit inside trigger?

What is difference between row level and statement level trigger

What is pragma in Oracle?

How union set operator is different than union all?

Performance tuning related questions

How will you identify whether query is taking too much time for execution and how to optimise running queries?

Difference between Group by and having clause?

What is rank(),dense\_rank() and row\_number() analytical functions and they are differ by their functionality?

What are main advantages of packages?

How packages improves the performance of database?

Write a query to find department wise total salary?

Write a query to count only specified characters from given string String: GOOGLE Total nu of "O" from GOOGLE

Write a query to find who is more earning than Chetan from employees table

Which are the types of subquery?

What is exists and not exists operator in correlated subquery?

What are the differences between Java and other programming languages?

What are the different types of Java collections?

What are the different types of Java exceptions?

What is the difference between a class and an interface?

What is the difference between a static and an instance method?

What is the difference between a synchronized and an unsynchronized method?

What is the difference between a local variable and an instance variable?

What is the difference between a final and a non-final variable?

What is the difference between a transient and a non-transient variable?

What are the different ways to implement a singleton in Java?

What is the difference between a hashmap and a hashset?

What is the difference between a synchronized block and a synchronized method?

What is the difference between a transient and a volatile variable?

What is the difference between a stack and a heap?

What is the difference between an abstract class and an interface?

What is the difference between an inner class and an anonymous class?

What is the difference between a static and a non-static inner class?

What is the difference between a local variable and a parameter?

**Interview Questions for IRIS**

1. What is self-join and Write a query of self-join?

A self-join is a type of SQL query where a table is joined with itself to compare different rows within the same table. This can be useful when you have a table with related information that needs to be compared or analyzed.

Let's consider a simple example with a hypothetical table called "employees" that stores information about employees in a company. The table has the following columns:

* employee\_id (primary key)
* employee\_name
* supervisor\_id (foreign key referencing the employee\_id of the supervisor)

**Interview Questions for Josh Software**

**Interview Question Josh Software**

Difference between ArrayList and LinkedList

How to resolve the merge conficts in git?

What are all the commands for git workflow?

What is Optional class in java?

A[] = {1, 2, 3, 4, 5}

B[] = {7, 1, 3, 4, 6}

What is A-B[] ?? write the program?

@RequestMapping, @Requestmapping path annotation,

HTTP error codes explain? What is success code?

Java8 features?

What are functional interface why do we need it?

Difference between static and default methods inside interface?

What is your daily tasks in the company?

What is singleton?