

Interview Theory Questions

1. What are threads? How do they differ from processes?

Threads are the smallest unit of execution within a process. They share the same memory space and resources of the process, whereas processes have their own memory space and resources.

2. Explain the two primary ways to create threads in Java.

Threads can be created in Java by either extending the Thread class or implementing the Runnable interface.

3. What is the difference between extending the Thread class and implementing the Runnable interface for creating threads?

Extending the Thread class binds the thread code with a specific thread instance, limiting the reusability. Implementing Runnable promotes better separation of concerns, allowing the same runnable code to be executed by different threads.

4. What is the purpose of the start() method when creating threads? Why shouldn't you call the run() method directly?

The start() method initializes the thread and schedules it for execution by the operating system. Calling run() directly would execute the thread's logic in the current thread, not creating a separate execution context.

5. List and explain the various thread states in Java.

The thread states are NEW, RUNNABLE, BLOCKED, WAITING, TIMED_WAITING, and TERMINATED. A thread progresses through these states during its lifecycle.

6. What is the difference between the NEW state and RUNNABLE state?

In the NEW state, the thread is created but not yet started. In the RUNNABLE state, the thread is ready for execution and waiting for the CPU time.

7. Describe a scenario where a thread might enter the WAITING state.

A thread might enter the WAITING state when it's waiting indefinitely for another thread to perform an action, such as when it's waiting on a wait() call within a synchronized block.

8. Why is it generally recommended to implement Runnable rather than extending Thread?

Implementing Runnable is recommended because Java supports multiple interfaces, enabling better code organization and reusability. It also avoids the single inheritance issue that arises when extending the Thread class.

9. How can a thread transition from the RUNNABLE state to the BLOCKED state?

A thread transitions from the RUNNABLE state to the BLOCKED state when it attempts to acquire a lock that is currently held by another thread.

10. What is the difference between the WAITING, TIMED_WAITING, and TERMINATED thread states?

In the WAITING state, a thread is waiting indefinitely for another thread's notification. In the TIMED_WAITING state, a thread is waiting for a specific time period. In the TERMINATED state, the thread has completed its execution.