Concurrency concepts and concurrency collections

Amo Samuel

Concurrency

Concurrency is the ability of a system to handle multiple tasks or operations simultaneously. In programming, concurrency allows different parts of a program to run independently and at the same time, potentially improving efficiency and responsiveness.

Threads and Processes

In Java, a thread is the smallest unit of execution within a process. While a process is an instance of a program, a thread is a path of execution within that process. Multithreading allows multiple threads to execute simultaneously, sharing the same resources, which can significantly improve performance.

Race Conditions

A race condition occurs when two or more threads access shared data and try to change it simultaneously. The final outcome depends on the timing of their execution, leading to unpredictable results.

Synchronization

Synchronization is crucial in multithreaded environments to ensure that only one thread accesses a critical section of code at a time. This prevents race conditions and ensures data consistency. In Java, synchronization can be achieved using synchronized blocks or methods.

Deadlocks

Deadlocks occur when two or more threads are waiting for each other to release resources, resulting in a standstill. Avoiding deadlocks requires careful design, such as maintaining a consistent order of resource acquisition or using timed locks.

Some concurrent collections in java

ConcurrentHashMap: A ConcurrentHashMap is a thread-steady implementation of a Map interface. It permits a couple of threads to have a look at and write without blocking off each other.

CopyOnWriteArrayList: CopyOnWriteArrayList is a thread-stable listing wherein adjustments create a modern day duplicate of the underlying array. It ensures that the actual list stays unaltered in some unspecified time in the future of generation via other thread.

BlockingQueue: BlockingQueue is a concurrent queue that is able to block off operations for getting added and doing away with factors. It is generally implemented for producer-consumer cases.

ConcurrentLinkedQueue: ConcurrentLinkedQueue is a non-blocking off thread-safe queue. It's an inexperienced choice for plenty of producer-consumer situations.

Sources

- javapoint.com
- Medium