fail-safe vs fail-fast Iterator in Java

Difference between fail-safe and fail-fast Iterator is becomingfavorite core java interview questions day by day, reason it touches concurrency a bit and interviewee can go deep on it to ask *how fail-safe or fail-fast behavior is implemented*. In this article article we will see **what is fail-safe and fail fast iterators in java** and differences between fail-fast and fail-safe iterators . Concept of fail-safe iterator are relatively new in Javaand first introduced with Concurrent Collections in Java 5 like`ConcurrentHashMap` and `CopyOnWriteArrayList`.

Difference between fail-fast Iterator vs fail-safe Iterator in Java

fail-fast Iterators in Java

As name suggest **fail-fast Iterators** fail as soon as they realized that *structure of Collection has been changed since iteration has begun*. Structural changes means adding, removing or updating any element from collection while one thread is Iterating over that collection. fail-fast behavior is implemented by keepinga modification count and if iteration thread realizes the change in modification count it throws`ConcurrentModificationException`.

Java doc says this is not a guaranteed behavior instead its done of "best effort basis", So application programming can not rely on this behavior. Also since multiple threads are involved while updating and checking modification count and this check is done without synchronization, there is a chance that Iteration thread still sees a stale value and might not be ableto detect any change done by parallel threads. Iterators returned by most of JDK1.4 collection are fail-fast including Vector,ArrayList, HashSet etc. to read more about Iterator see my post What is Iterator in Java.

fail-safe Iterator in java

Contrary to fail-fast Iterator, **fail-safe iterator** doesn't throw any Exception if Collection is modified structurally while one thread is Iterating over it because they work on clone of Collection instead of original collection and that's why they are called as fail-safe iterator. Iterator of `CopyOnWriteArrayList` is an example of fail-safe Iterator also iterator written by `ConcurrentHashMap` keySet is also fail-safe iterator and never throw`ConcurrentModificationException` in Java.

That's all on **difference between fail-safe vs fail-fast Iterator in Java**, As I said due to there confusing or not to easy differentiation they are quickly becoming popular java collection questions asked on various level of java interview. Let me know if you are aware of any other difference between fail-fast and fail-safe iterator.