Reference Posts:

* <https://rathod-ajay.medium.com/your-guide-to-clear-java-developer-interview-in-2024-36a926ec6719>

1. ArrayList vs LinkedList

* ArrayList uses dynamic array, LinkedList uses doubly linked list
* ArrayList implements

1. Where are java objects created?
   * Java objects are created inside heap memory.
2. How HashMap internally works?
   * HashMap has 2 Data structures: Array, LinkedList
   * HashMap creates 16 buckets. It increases the bucket size when it reaches 75% size.
   * As per the new implementation, linked list will be converted to a tree to reduce the traversal time.
   * Earlier complexity was O(1+n), after converting to a binary search tree, it will be O(1+logn);
   * JVM will use compareTo() method to check the values.
3. HashMap vs HashTable

|  | **HashMap** | **HashTable** |
| --- | --- | --- |
|  | Non-synchronized. Multiple threads can access simultaneously, but synchronization must be externally handled for safe concurrent modification. | Synchronized. Only one thread can access at a time, ensuring thread safety. |
|  | Allows one null key and multiple null values. | Does not permit null keys or values. Inserting null can lead to a NullPointerException. |
|  | The iterator is fail-fast. Throws ConcurrentModificationException if modified by another thread during iteration. | Enumerator is not fail-fast. Due to internal synchronization, concurrent modification risks are minimized during enumeration. |
|  | Multiple threads can operate without waiting. Generally offers better performance in single-threaded scenarios due to lack of synchronization overhead. | Threads may need to wait due to synchronization. This can lead to performance overhead in multi-threaded scenarios. |
|  | Non-legacy. | Considered legacy; newer implementations like HashMap are generally recommended. |

1. Java 8 features:

Lambda Expressions

Functional Interfaces

Stream API

Date/Time API

CompletableFuture

Optional Classes

Default methods in interfaces

Method references

1. ConcurrentHashMap:

* Thread safe implementation of HashMap
* provides fine-grained locking, meaning that it locks only the portion of the map being modified, rather than the entire map
* Multiple threads can access it simultaneously without any synchronization issues
* The underlined data structure for ConcurrentHashMap is Hashtable.

1. Sdf
2. Asdf
3. As
4. df