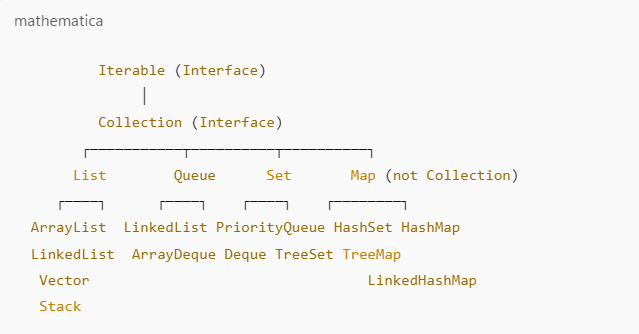
**Java Collections Framework Hierarchy**

****

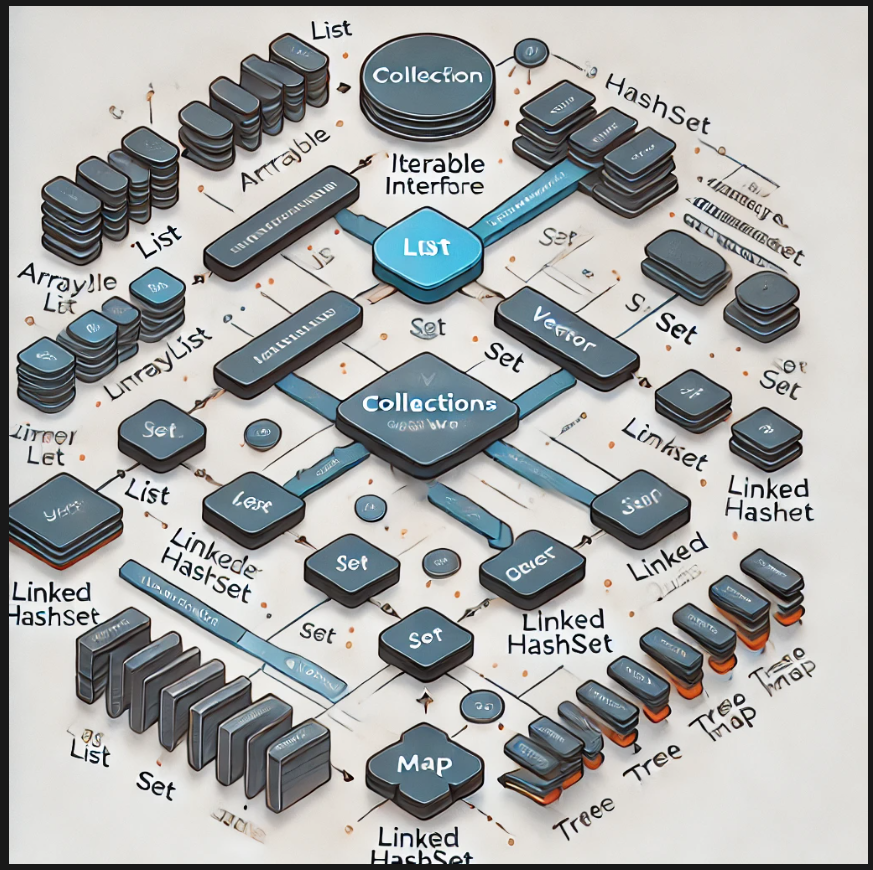
**Explanation of the Java Collections Hierarchy:**

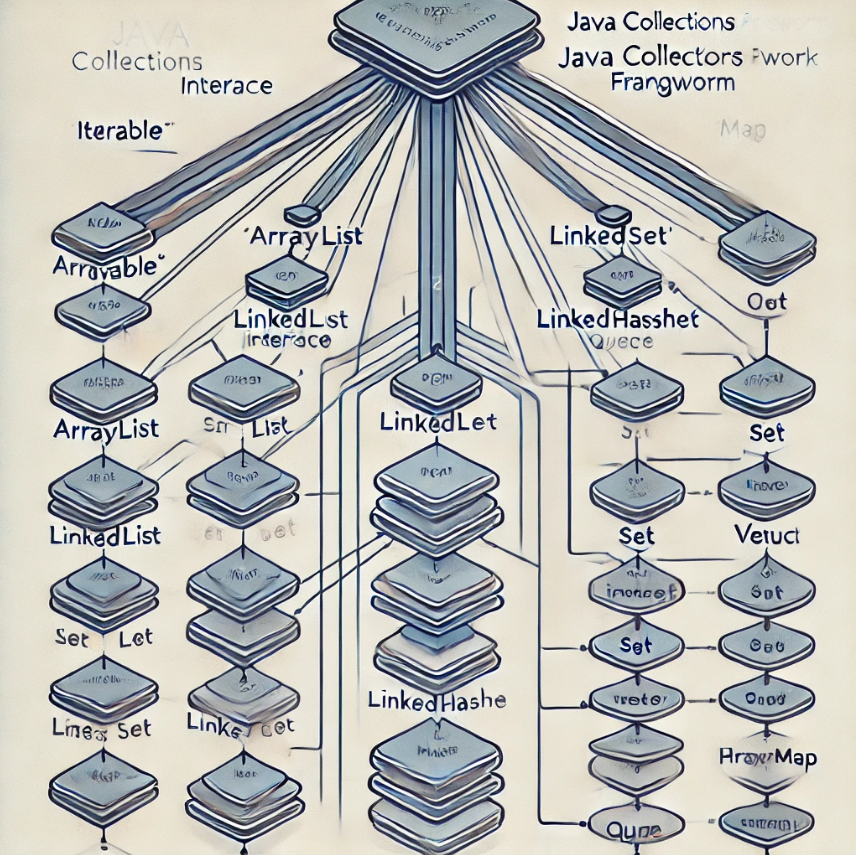
** Iterable Interface**

* Root interface that allows iteration using a loop.
* Implemented by Collection and other sub interfaces like List, Set, and Queue.

** Collection Interface**

* Extends Iterable and is the parent of all collection types.
* Methods: add(), remove(), contains(), size(), isEmpty(), etc.





**1. List (Ordered, Allows Duplicates):**

* **Implementations:**
  + ArrayList (Fast Random Access, Slower Insert/Delete)
  + LinkedList (Fast Insert/Delete, Slow Random Access)
  + Vector (Thread-Safe, Legacy)
  + Stack (LIFO, Legacy)

**2. Set (Unique Elements, No Duplicates):**

* **Implementations:**
  + HashSet (Unordered, Fast Lookups)
  + LinkedHashSet (Maintains Insertion Order)
  + TreeSet (Sorted Elements)

**3. Queue (FIFO):**

* **Implementations:**
  + PriorityQueue (Sorted Order, Heap-Based)
  + ArrayDeque (Deque, Fast Insert/Delete from Both Ends)

**4. Map (Key-Value Pairs, not a Collection)**

* **Implementations:**
  + HashMap (Unordered, Fast)
  + LinkedHashMap (Maintains Insertion Order)
  + TreeMap (Sorted Keys)

1. List Interface (Ordered, Allows Duplicates)

A List in Java is an **ordered collection** that allows **duplicate elements**. Elements in a List can be accessed by index, and insertion order is preserved.

