|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Collection Type** | **Class/Interface** | **Ordered** | **Sorted** | **Allows Duplicates** | **Null Allowed** | **Thread-safe** | **Special Features** | | **List** | ArrayList | ✅ Yes | ❌ No | ✅ Yes | ✅ Yes | ❌ No | Fast random access | |  | LinkedList | ✅ Yes | ❌ No | ✅ Yes | ✅ Yes | ❌ No | Fast insert/delete | | **Set** | HashSet | ❌ No | ❌ No | ❌ No | ✅ Yes | ❌ No | Backed by HashMap | |  | LinkedHashSet | ✅ Yes | ❌ No | ❌ No | ✅ Yes | ❌ No | Maintains insertion order | |  | TreeSet | ❌ No | ✅ Yes | ❌ No | ❌ No | ❌ No | Sorted set (Red-Black Tree) | | **Queue** | PriorityQueue | ❌ No | ✅ Yes | ✅ Yes | ❌ No | ❌ No | Sorted by priority | |  | ArrayDeque | ✅ Yes | ❌ No | ✅ Yes | ❌ No | ❌ No | Double-ended queue | | **Map** | HashMap | ❌ No | ❌ No | ❌ Keys | ✅ One key, multiple values | ❌ No | Fast lookup | |  | LinkedHashMap | ✅ Yes | ❌ No | ❌ Keys | ✅ One key, multiple values | ❌ No | Maintains insertion/access order | |  | TreeMap | ✅ Yes | ✅ Yes | ❌ Keys | ❌ No | ❌ No | Sorted by key | |  | Hashtable | ❌ No | ❌ No | ❌ Keys/Values | ❌ No | ✅ Yes | Legacy synchronized map | |  | ConcurrentHashMap | ❌ No | ❌ No | ❌ Keys | ❌ No | ✅ Yes | Thread-safe, modern | |

**💡 Tips for Choosing:**

* Use ArrayList when you need fast random access.
* Use LinkedList when frequent insertions/deletions are needed.
* Use HashSet for uniqueness without order.
* Use LinkedHashSet when insertion order matters.
* Use TreeSet for sorted unique elements.
* Use HashMap for key-value pairs with fast lookup.
* Use LinkedHashMap when you need ordered key-value pairs.
* Use TreeMap when keys must be sorted.
* Use ConcurrentHashMap for thread-safe operations