Class	**Implements**	**Duplicates Allowed**	**Maintains Order**	**Sorted**	**Allows Nulls**	**Thread-Safe**	**Best Use Case**
ArrayList	List	✅ Yes	✅ Yes	❌ No	✅ Yes (1 null)	❌ No	Random access, fast read
LinkedList	List, Deque	✅ Yes	✅ Yes	❌ No	✅ Yes	❌ No	Frequent insert/delete
Vector	List	✅ Yes	✅ Yes	❌ No	✅ Yes	✅ Yes	Legacy synchronized list
Stack	Vector	✅ Yes	✅ Yes (LIFO)	❌ No	✅ Yes	✅ Yes	LIFO operations
HashSet	Set	❌ No	❌ No	❌ No	✅ Yes (1 null)	❌ No	Unique elements, fast lookup
LinkedHashSet	Set	❌ No	✅ Yes	❌ No	✅ Yes	❌ No	Unique with insertion order
TreeSet	NavigableSet	❌ No	✅ Yes (Sorted)	✅ Yes (Natural/C)	❌ No (no nulls)	❌ No	Sorted unique elements
HashMap	Map	✅ Keys: ❌ Duplicates ✅ Values	❌ No	❌ No	✅ One null key, multiple null values	❌ No	Key-value store, fast lookup
LinkedHashMap	Map	✅ Yes	✅ Yes	❌ No	✅ Yes	❌ No	Maintains insertion order in maps
TreeMap	NavigableMap	✅ Yes	✅ Yes (Sorted)	✅ Yes	❌ No (no null keys)	❌ No	Sorted key-value pairs
Hashtable	Map	✅ Yes	❌ No	❌ No	❌ No (no nulls)	✅ Yes	Legacy synchronized map
PriorityQueue	Queue	✅ Yes	❌ No (Heap order)	✅ Yes (Priority)	❌ No	❌ No	Priority-based processing

Class	Implements	Duplicates Allowed	Maintains Order	Sorted	Allows Nulls	Thread-Safe	Best Use Case
ArrayDeque	Deque	✅ Yes	✅ Yes	❌ No	❌ No	❌ No	Fast stack/queue with no capacity restrictions
EnumSet	Set	❌ No	✅ Yes (Enum order)	✅ Yes	❌ No	❌ No	Efficient set for enum types
WeakHashMap	Map	✅ Yes	❌ No	❌ No	✅ Yes	❌ No	GC-aware map (weak keys)
ConcurrentHashMap	Map	✅ Yes	❌ No	❌ No	❌ No null keys	✅ Yes	Thread-safe map without locking
CopyOnWriteArrayList	List	✅ Yes	✅ Yes	❌ No	✅ Yes	✅ Yes	Thread-safe list, good for read-heavy use

Feature	LinkedHashMap	TreeMap	Hashtable
Allows Duplicate Keys	❌ No (keys must be unique)	❌ No (keys must be unique)	❌ No (keys must be unique)
Allows Duplicate Values	✅ Yes	✅ Yes	✅ Yes
Key Order Maintained	✅ Yes (insertion order)	✅ Yes (sorted order - natural/comparator)	❌ No (no guaranteed order)

Feature	**LinkedHashMap**	**TreeMap**	**Hashtable**
Null Keys Allowed	✅ Yes (only 1 null key)	❌ No (throws NullPointerException)	❌ No (throws NullPointerException)
Null Values Allowed	✅ Yes (multiple null values)	✅ Yes	❌ No (throws NullPointerException)
Thread-Safe	❌ No	❌ No	✅ Yes (fully synchronized)
Use Case	Maintain key insertion order	Maintain sorted order	Legacy synchronized map

Here's a ✅ **complete list of core implementation classes** in the **Java Collection Framework**, organized by their respective interfaces (e.g., List, Set, Map, Queue, etc.).

✅ 1. List Interface – Ordered Collection (Allows Duplicates)

Implementation Class	Description
ArrayList	Resizable array, fast random access
LinkedList	Doubly linked list, efficient insert/delete
Vector (legacy)	Synchronized dynamic array
Stack (extends Vector)	Legacy LIFO stack
CopyOnWriteArrayList	Thread-safe list (java.util.concurrent)

✅ 2. Set Interface – No Duplicates

Implementation Class	Description
HashSet	Unordered set, backed by HashMap
LinkedHashSet	Maintains insertion order
TreeSet	Sorted set (Red-Black tree)
EnumSet	High-performance set for enums
CopyOnWriteArraySet	Thread-safe set (java.util.concurrent)

✅ 3. Queue & Deque Interfaces – FIFO, LIFO, and Priority

Implementation Class	Description
PriorityQueue	Elements ordered by priority
ArrayDeque	Resizable double-ended queue
LinkedList	Implements both Queue and Deque
ConcurrentLinkedQueue	Thread-safe queue (non-blocking)
LinkedBlockingQueue	Blocking queue (java.util.concurrent)
ArrayBlockingQueue	Bounded blocking queue

Implementation Class	Description
PriorityBlockingQueue	Thread-safe priority queue
DelayQueue	Elements become available after delay
SynchronousQueue	For thread handoff, no internal storage
LinkedTransferQueue	High-performance concurrent queue

✅ 4. Map Interface – Key-Value Pairs

Implementation Class	Description
HashMap	Fast lookup via hash table
LinkedHashMap	Maintains insertion order
TreeMap	Sorted map (Red-Black tree)
Hashtable (legacy)	Synchronized map
WeakHashMap	Keys are weakly referenced
IdentityHashMap	Compares keys by reference (==)
EnumMap	Efficient map for enum keys
ConcurrentHashMap	Thread-safe, high concurrency

✅ 5. Other Specialized Implementations

Class	Implements/Supports
Properties	Subclass of Hashtable, used for configs
Collections (Utility)	Static utility methods for collections
Arrays (Utility)	Static methods for arrays

In Java, the **Collection Framework** provides various interfaces and implementation classes. The most common implementation classes include:

- ArrayList
- LinkedList
- Vector
- HashSet
- LinkedHashSet
- TreeSet
- HashMap
- LinkedHashMap
- TreeMap
- Stack
- PriorityQueue

Each of these classes implements specific interfaces like List, Set, Queue, or Map and provides its own set of methods (many inherited from their interfaces).

🔷 1. ArrayList (implements List)

java

CopyEdit

add(E e)

add(int index, E element)

get(int index)

set(int index, E element)

remove(int index)

remove(Object o)

contains(Object o)

indexOf(Object o)

size()

clear()

isEmpty()

toArray()

🔷 2. LinkedList (implements List, Deque)

java

CopyEdit

add(E e)

addFirst(E e)

addLast(E e)

getFirst()

getLast()

removeFirst()

removeLast()

offer(E e)

poll()

peek()

push(E e)

pop()

🔷 3. Vector (implements List)

java

CopyEdit

add(E e)

add(int index, E element)

get(int index)

remove(int index)

set(int index, E element)

capacity()

size()

isEmpty()

clear()

🔷 4. Stack (extends Vector)

java

CopyEdit

push(E item)

pop()

peek()

search(Object o)

empty()

🔷 5. HashSet (implements Set)

java

CopyEdit

add(E e)

remove(Object o)

contains(Object o)

isEmpty()

size()

clear()

iterator()

🔷 6. LinkedHashSet (extends HashSet)

Inherits all methods from HashSet, maintains insertion order.

🔷 7. TreeSet (implements NavigableSet)

java

CopyEdit

add(E e)

ceiling(E e)

floor(E e)

first()

last()

higher(E e)

lower(E e)

pollFirst()

pollLast()

🔷 8. PriorityQueue (implements Queue)

java

CopyEdit

add(E e)

offer(E e)

poll()

peek()

remove()

size()

clear()

🔷 9. HashMap (implements Map)

java

CopyEdit

put(K key, V value)

get(Object key)

remove(Object key)

containsKey(Object key)

containsValue(Object value)

keySet()

values()

entrySet()

size()

clear()

🔷 10. LinkedHashMap (extends HashMap)

Inherits all methods from HashMap, maintains insertion order.

🔷 11. TreeMap (implements NavigableMap)

java

CopyEdit

put(K key, V value)

firstKey()

lastKey()

ceilingKey(K key)

floorKey(K key)

pollFirstEntry()

pollLastEntry()

subMap(K fromKey, K toKey)