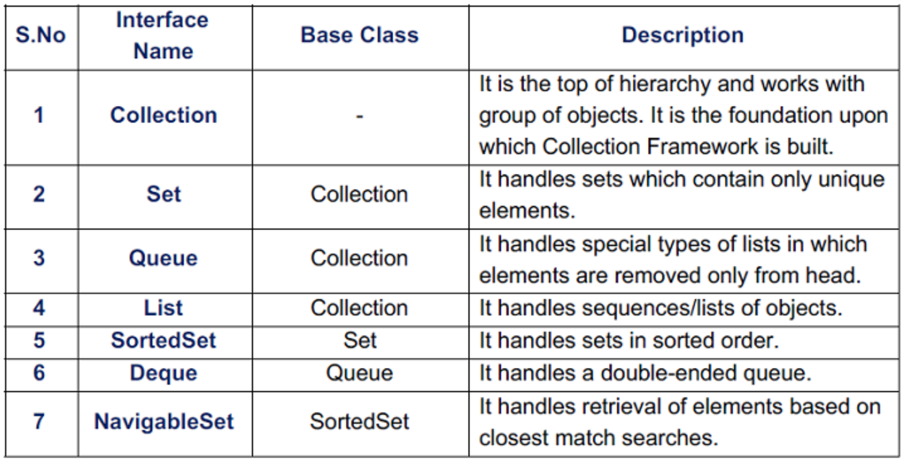
Java Collections Tutorial Part - 2

6

In the previous tutorial, we have seen the introduction to JCF and Iterators. In this tutorial, lets look into the details of Collection Interfaces and Classes.

THE COLLECTION INTERFACES

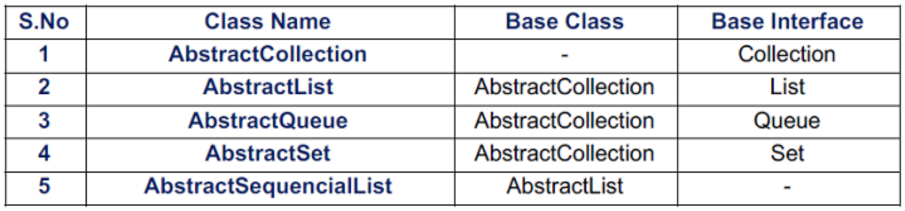
The Collection Classes implementing these interfaces have the functionality of these interfaces. Therefore, these interfaces define the core behavior of a Collection class.



S.No	Interface Name	Base Class	Description
1	Collection	-	It is the top of hierarchy and works with group of objects. It is the foundation upon which Collection Framework is built.
2	Set	Collection	It handles sets which contain only unique elements.
3	Queue	Collection	It handles special types of lists in which elements are removed only from head.
4	List	Collection	It handles sequences/lists of objects.
5	SortedSet	Set	It handles sets in sorted order.
6	Deque	Queue	It handles a double-ended queue.
7	NavigableSet	SortedSet	It handles retrieval of elements based on closest match searches.

THE COLLECTION ABSTRACT CLASSES

The Abstract Classes provide skeletal implementations that are used as starting points for creating concrete Collections.



S.No	Class Name	Base Class	Base Interface
1	AbstractCollection	-	Collection
2	AbstractList	AbstractCollection	List
3	AbstractQueue	AbstractCollection	Queue
4	AbstractSet	AbstractCollection	Set
5	AbstractSequencialList	AbstractList	-

THE COLLECTION CLASSES

These standard classes provide full implementation of the Collection that can be used as-is.

The below table shows various differences between the Collection classes. The terms used in the table are -

1) AD - Allow Duplicates: It tells whether that particular collection allows duplicate values to be inserted.

2) AN - Allow Null: It tells whether null objects can be inserted into that particular collection.

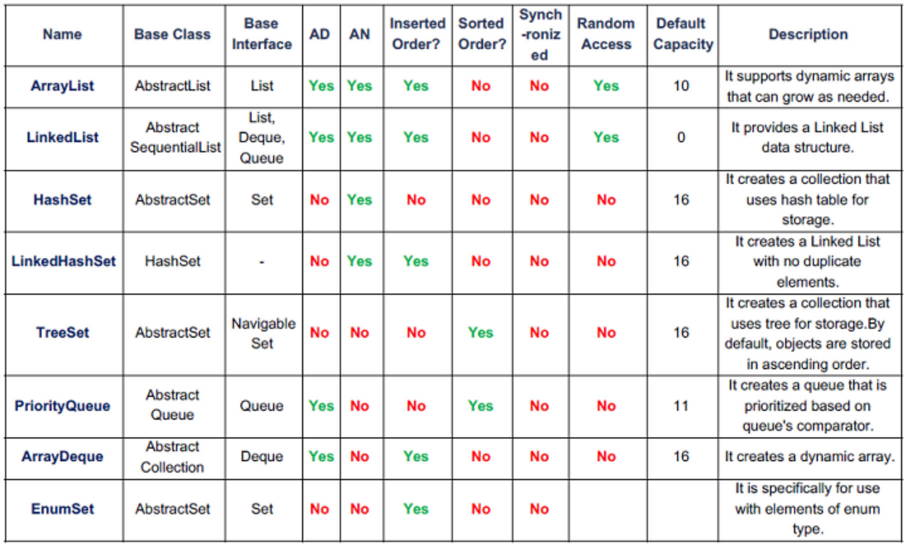
3) Inserted Order: It tells whether the objects are stored in the same order in which they were inserted.

4) Sorted Order: It tells whether the objects are stored in sorted order.

5) Synchronized: It tells whether the collection is thread-safe or not.

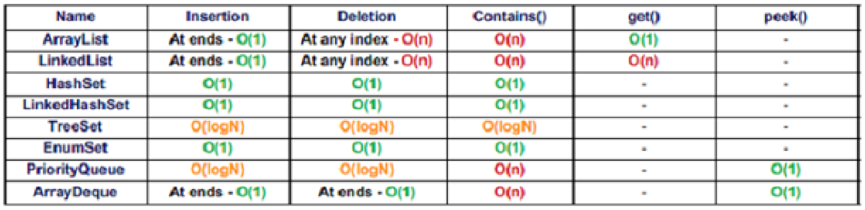
6) Random Access: It tells whether the collection has a get() method to returns the index of an object or return the object using an index.

7) Default capacity: The initial capacity of the collection when it is created using an empty constructor.



Name	Base Class	Base Interface	AD	AN	Inserted Order?	Sorted Order?	Synch-ronized	Random Access	Default Capacity	Description
ArrayList	AbstractList	List	Yes	Yes	Yes	No	No	Yes	10	It supports dynamic arrays that can grow as needed.
LinkedList	Abstract SequentialList	List, Deque, Queue	Yes	Yes	Yes	No	No	Yes	0	It provides a Linked List data structure.
HashSet	AbstractSet	Set	No	Yes	No	No	No	No	16	It creates a collection that uses hash table for storage.
LinkedHashSet	HashSet	-	No	Yes	Yes	No	No	No	16	It creates a Linked List with no duplicate elements.
TreeSet	AbstractSet	Navigable Set	No	No	No	Yes	No	No	16	It creates a collection that uses tree for storage.By default, objects are stored in ascending order.
PriorityQueue	Abstract Queue	Queue	Yes	No	No	Yes	No	No	11	It creates a queue that is prioritized based on queue's comparator.
ArrayDeque	Abstract Collection	Deque	Yes	No	Yes	No	No	No	16	It creates a dynamic array.
EnumSet	AbstractSet	Set	No	No	Yes	No	No			It is specifically for use with elements of enum type.

The Time Complexities of basic operations are given below:



Name	Insertion	Deletion	Contains()	get()	peek()
ArrayList	At ends - O(1)	At any index - O(n)	O(n)	O(1)	-
LinkedList	At ends - O(1)	At any index - O(n)	O(n)	O(n)	-
HashSet	O(1)	O(1)	O(1)	-	-
LinkedHashSet	O(1)	O(1)	O(1)	-	-
TreeSet	O(logN)	O(logN)	O(logN)	-	-
EnumSet	O(1)	O(1)	O(1)	-	-
PriorityQueue	O(logN)	O(logN)	O(n)	-	O(1)
ArrayDeque	At ends - O(1)	At ends - O(1)	O(n)	-	O(1)

Click here to go to next tutorial

Java Collections Tutorial Series:

Java Collections Tutorial Part - 1

Java Collections Tutorial Part - 2

Java Collections Tutorial Part - 3

Java Collections Tutorial Part - 4