|  |  |  |
| --- | --- | --- |
| METHODS OF COLLECTION INTERFACE | | |
| **No.** | **Method** | **Description** |
| 1 | **public boolean add(E e)** | It is used to insert an element in this collection. |
| 2 | **public boolean addAll(Collection<? extends E> c)** | It is used to insert the specified collection elements in the invoking collection. |
| 3 | **public boolean remove(Object element)** | It is used to delete an element from the collection. |
| 4 | **public boolean removeAll(Collection<?> c)** | It is used to delete all the elements of the specified collection from the invoking collection. |
| 5 | **default boolean removeIf(Predicate<? super E> filter)** | It is used to delete all the elements of the collection that satisfy the specified predicate. |
| 6 | **public boolean retainAll(Collection<?> c)** | It is used to delete all the elements of invoking collection except the specified collection. |
| 7 | **public int size()** | It returns the total number of elements in the collection. |
| 8 | **public void clear()** | It removes the total number of elements from the collection. |
| 9 | **public boolean contains(Object element)** | It is used to search an element. |
| 10 | **public boolean containsAll(Collection<?> c)** | It is used to search the specified collection in the collection. |
| 11 | **public Iterator iterator()** | It returns an iterator. |
| 12 | **public Object[] toArray()** | It converts collection into array. |
| 13 | **public <T> T[] toArray(T[] a)** | It converts collection into array. Here, the runtime type of the returned array is that of the specified array. |
| 14 | **public boolean isEmpty()** | It checks if collection is empty. |
| 15 | **default Stream<E> parallelStream()** | It returns a possibly parallel Stream with the collection as its source. |
| 16 | **default Stream<E> stream()** | It returns a sequential Stream with the collection as its source. |
| 17 | **default Spliterator<E> spliterator()** | It generates a Spliterator over the specified elements in the collection. |
| 18 | **public boolean equals(Object element)** | It matches two collections. |
| 19 | **public int hashCode()** | It returns the hash code number of the collection. |

|  |  |
| --- | --- |
| Constructor | Description |
| ArrayList() | It is used to build an empty array list. |
| ArrayList(Collection<? extends E> c) | It is used to build an array list that is initialized with the elements of the collection c. |
| ArrayList(int capacity) | It is used to build an array list that has the specified initial capacity. |
|  |  |
| Method  Description | |
| [void add(int index, E element)](https://www.javatpoint.com/java-arraylist-add-method) | It is used to insert the specified element at the specified position in a list. |
| [boolean add(E e)](https://www.javatpoint.com/java-arraylist-add-method) | It is used to append the specified element at the end of a list. |
| [boolean addAll(Collection<? extends E> c)](https://www.javatpoint.com/java-arraylist-addall-method) | It is used to append all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's iterator. |
| [boolean addAll(int index, Collection<? extends E> c)](https://www.javatpoint.com/java-arraylist-addall-method) | It is used to append all the elements in the specified collection, starting at the specified position of the list. |
| [void clear()](https://www.javatpoint.com/java-arraylist-clear-method) | It is used to remove all of the elements from this list. |
| void ensureCapacity(int requiredCapacity) | It is used to enhance the capacity of an ArrayList instance. |
| E get(int index) | It is used to fetch the element from the particular position of the list. |
| boolean isEmpty() | It returns true if the list is empty, otherwise false. |
| [Iterator()](https://www.javatpoint.com/java-arraylist-iterator-method) |  |
| [listIterator()](https://www.javatpoint.com/java-arraylist-listiterator-method) |  |
| int lastIndexOf(Object o) | It is used to return the index in this list of the last occurrence of the specified element, or -1 if the list does not contain this element. |
| Object[] toArray() | It is used to return an array containing all of the elements in this list in the correct order. |
| <T> T[] toArray(T[] a) | It is used to return an array containing all of the elements in this list in the correct order. |
| Object clone() | It is used to return a shallow copy of an ArrayList. |
| boolean contains(Object o) | It returns true if the list contains the specified element. |
| int indexOf(Object o) | It is used to return the index in this list of the first occurrence of the specified element, or -1 if the List does not contain this element. |
| E remove(int index) | It is used to remove the element present at the specified position in the list. |
| [boolean remove(Object o)](https://www.javatpoint.com/java-arraylist-remove-method) | It is used to remove the first occurrence of the specified element. |
| [boolean removeAll(Collection<?> c)](https://www.javatpoint.com/java-arraylist-removeall-method) | It is used to remove all the elements from the list. |
| boolean removeIf(Predicate<? super E> filter) | It is used to remove all the elements from the list that satisfies the given predicate. |
| [protected void removeRange(int fromIndex, int toIndex)](https://www.javatpoint.com/java-arraylist-removerange-method) | It is used to remove all the elements lies within the given range. |
| void replaceAll(UnaryOperator<E> operator) | It is used to replace all the elements from the list with the specified element. |
| [void retainAll(Collection<?> c)](https://www.javatpoint.com/java-arraylist-retainall-method) | It is used to retain all the elements in the list that are present in the specified collection. |
| E set(int index, E element) | It is used to replace the specified element in the list, present at the specified position. |
| void sort(Comparator<? super E> c) | It is used to sort the elements of the list on the basis of the specified comparator. |
| Spliterator<E> spliterator() | It is used to create a spliterator over the elements in a list. |
| List<E> subList(int fromIndex, int toIndex) | It is used to fetch all the elements that lies within the given range. |
| int size() | It is used to return the number of elements present in the list. |
| void trimToSize() | It is used to trim the capacity of this ArrayList instance to be the list's current size. |