**What is a Method present in Collection interface, list interface, Set interface, Queue interface.**

The Collection interface is a member of the [Java Collections Framework](https://www.geeksforgeeks.org/collections-in-java-2/). It is a part of java.util package. It is one of the root interfaces of the Collection Hierarchy. The Collection interface is not directly implemented by any class. However, it is implemented indirectly via its subtypes or subinterfaces like [List](https://www.geeksforgeeks.org/list-interface-java-examples/), [Queue](https://www.geeksforgeeks.org/queue-interface-java/), and [Set](https://www.geeksforgeeks.org/set-in-java/).

For Example, the [HashSet](https://www.geeksforgeeks.org/hashset-in-java/" \l ":~:text=Last%20Updated%3A%2009%2D09%2D,class%20permits%20the%20null%20element.) class implements the Set interface which is a subinterface of the Collection interface. If a collection implementation doesn’t implement a particular operation, it should define the corresponding method to throw UnsupportedOperationException.

**List**: This is a child interface of the collection interface. This interface is dedicated to the data of the list type in which we can store all the ordered collection of the objects. This also allows duplicate data to be present in it. This list interface is implemented by various classes like [ArrayList](https://www.geeksforgeeks.org/arraylist-in-java/), [Vector](https://www.geeksforgeeks.org/java-util-vector-class-java/), [Stack](https://www.geeksforgeeks.org/stack-class-in-java/), etc.

**Set**: A set is an unordered collection of objects in which duplicate values cannot be stored. This collection is used when we wish to avoid the duplication of the objects and wish to store only the unique objects. This set interface is implemented by various classes like [HashSet](http://www.geeksforgeeks.org/hashset-in-java/), [TreeSet](https://www.geeksforgeeks.org/treeset-in-java-with-examples/), [LinkedHashSet](https://www.geeksforgeeks.org/linkedhashset-in-java-with-examples/), etc.

**Queue**: As the name suggests, a queue interface maintains the FIFO(First In First Out) order similar to a real-world queue line. This interface is dedicated to storing all the elements where the order of the elements matter. For example, whenever we try to book a ticket, the tickets are sold at the first come first serve basis. Therefore, the person whose request arrives first into the queue gets the ticket. There are various classes like [PriorityQueue](https://www.geeksforgeeks.org/priority-queue-class-in-java-2/), [Deque](https://www.geeksforgeeks.org/deque-set-1-introduction-applications/), [ArrayDeque](https://www.geeksforgeeks.org/arraydeque-in-java/), etc.

**What is a Callection framework library besides the concepts covered in sessions.**

The Collection framework represents a unified architecture for storing and manipulating a group of objects. It has:

1. Interfaces and its implementations, i.e., classes
2. Algorithm

### ****Hierarchy of Collection Framework****

Let us see the hierarchy of Collection framework. The **java.util** package contains all the [classes](https://www.javatpoint.com/object-and-class-in-java) and [interfaces](https://www.javatpoint.com/interface-in-java) for the Collection framework.