public static <T extends Comparable<? super T>> void sort(List<T> list)

This is a static generic method that declares type T is of type Comparable or a subclass of Comparable. The Comparable interface is also generic which accepts any class that type T is a subclass of.

The return type is void, nothing is returned. The method name is “sort” and the only parameter is of type “List<T>” where T is Comparable or extended from Comparable.

public static <T> void sort(List<T> list, Comparator<? super T> c)

This is a static generic method where the generic type is defined as type T.

The return type is void, nothing is returned. The method name is “sort” accepting two parameters. The first parameter is of type “List<T>” where T is the type T defined. The second parameter is a lower-bounded wildcard type of Comparator where any type is a parent class of T.

public static <T> int binarySearch(List<? extends Comparable<? super T>> list, T key)

This is a static generic method where the generic type is defined as type T.

The return type is an integer. The method name is “binarySearch” and accepts two parameters. The first parameter accepts a List with contents of types that are a subclass of Comparable. The Comparable type has a lower bound type T where accepted types are a super of T. The second parameter is of type T.

public static void shuffle(List<?> list)

This is a static method where there are no generic types defined.

The return type is void, nothing is returned. The method name is “shuffle” accepting one parameter. The parameter is the type “List<?>” where the contents do not have any type restrictions. There are no boundaries and it will accept any type.

public static <T> void copy(List<? super T> dest, List<? extends T> src)

This is a static generic method where the generic type is defined as type T.

The return type is void, nothing is returned. The method name is “copy” accepting two parameters. The first parameter is of type “List<? super T>”, where the accepted types for List contents are a parent class of T. The second parameter is of type “List<? extends T>”, where the accepted types for List contents are a subclass of T.