

```
interface
java.util.List
(First, methods that must be implemented.
This list omits any methods added after JDK 8.):

contains
containsAll
equals
get
hashCode
indexOf
isEmpty
iterator
lastIndexOf
listIterator 2 signatures:
    listIterator()
    listIterator(int index)
size
subList
default spliterator
toArray 2 signatures:
    toArray()
    toArray(T[] a)

(Methods that are optional, and Clojure throws
UnsupportedOperationException when called,
by design.):

add 2 signatures:
    add(E e)
    add(int index, E element)
addAll 2 signatures:
    addAll(Collection<? extends E> c)
    addAll(int index, Collection<? extends E> c)
clear
remove 2 signatures:
    remove(Object o)
    remove(int index)
removeAll
default replaceAll
retainAll
set
default sort
```

```
interface
java.util.Collection
- means in List, but not in Collection
+ means not in List, but in Collection

(First, methods that must be implemented.
This list omits any methods added after JDK 8.):

contains
containsAll
equals
- get
hashCode
- indexOf
isEmpty
iterator
- lastIndexOf
- listIterator 2 signatures:
    - listIterator()
    - listIterator(int index)
+ default parallelStream
size
- subList
default spliterator
+ default stream
+ toArray 3 signatures:
    toArray()
    toArray(T[] a)
+ default toArray(IntFunction<T[]> generator)

(Methods that are optional, and Clojure throws
UnsupportedOperationException when called,
by design.):

add 1 signature:
    add(E e)
- add(int index, E element)
addAll 1 signature:
    addAll(Collection<? extends E> c)
- addAll(int index, Collection<? extends E> c)
clear
remove 1 signature:
    remove(Object o)
- remove(int index)
removeAll
+ default removeIf
- default replaceAll
retainAll
- set
- default sort
```

```
interface
java.lang.Iterable
iterator
default forEach
default spliterator
```