List, Queue & Stack

```
interface List<E>

   boolean        add(E obj)
      void        add(int index, E obj)
         E        remove(int index)
       int        size()
         E        get(int index)
         E        set(int index, E obj)
Iterator<E>       iterator()
ListIterator<E>   listIterator()
```

```
interface Iterator<E>

   boolean  hasNext()
         E  next()
      void  remove()
```

```
interface ListIterator<E> extends Iterator<E>

      void  add(E obj)
      void  set(E obj)
```

```
interface Queue<E>

   boolean   add(E obj)
         E   remove()
         E   peek()
   boolean   isEmpty()
```

```
class ArrayList<E> implements List<E>
```

```
class LinkedList<E> implements List<E>, Queue<E>

      void  addFirst(E obj)
      void  addLast(E obj)
         E  getFirst()
         E  getLast()
         E  removeFirst()
         E  removeLast()
```

```
class Stack<E>

         E   push(E item)
         E   pop()
         E   peek()
   boolean   isEmpty()
```

```
class PriorityQueue<E>

   boolean   isEmpty()
   boolean   add(E x)
         E   remove ()
         E   peek ()
```

usually implemented with a min- or max-heap (i.e. a type of binary tree)