* **List**
* Java中数组用来存储数据的局限性
* List集合类中**元素有序**、**且可重复**，集合中的每个元素都有其对应的顺序索引。
* List容器中的**元素都对应一个整数型的序号记载其在容器中的位置**，可以根据序号存取容器中的元素。
* JDK API中的List接口的实现类常用的有：**ArrayList**、**LinkedList**和**Vector**。
* **List集合里添加了一些根据索引来操作集合元素的方法。**
* void add(int index, Object ele)——在指定索引位置index添加元素ele。
* boolean addAll(int index, Collection eles)——
* Object get(int index)——获取指定索引位置的元素。
* int indexOf(Object obj)——返回obj在集合中首次出现的位置，没有返回-1。
* int lastIndexOf(Object obj)——返回obj在集合中最后一次出现的位置。
* Object remove(int index)——删除指定索引位置的元素。
* Object set(int index, Object ele)——修改指定索引位置的元素为ele。
* List subList(int fromIndex, int toIndex)——返回从fromIndex到toIndex结束的一个子list。（左闭右开）
* **List常用方法：增（add()） 删（remove()） 改（set()） 查（get()） 插（add()） 长度（size()）**
* **List实现类之一：ArrayList——主要实现类**
* ArrayList是List接口的典型实现类。
* 本质上，**ArrayList是对象引用的一个变长数组**。
* ArrayList是线程不安全的，而Vector是线程安全的，即使为保证List集合线程安全，也不推荐使用Vector。
* Arrays.asList(...)方法返回的List集合既不是ArrayList实例，也不是Vector实例，**Arrays.asList(...)返回值是一个固定长度的List集合**。
* **List实现类之二：LinkedList**
* 对于**频繁的插入或删除元素**的操作，建议使用LinkedList类，效率较高。
* 新增方法：
  + void addFirst(Object obj)
  + void addLast(Object obj)
  + Object getFirst()
  + Object getLast()
  + Object removeFirst()
  + Object removeLast()
* **List实现类之三：Vector**
* Vector是一个古老的集合，JDK1.0就有了。大多数操作与ArrayList相同，区别之处在于Vector是线程安全的。
* 在各种list中，最好把ArrayList作为缺省选择。当插入、删除频繁时，使用LinkedList；Vector总是比ArrayList慢，所以尽量避免使用。
* 新增方法：
  + void addElement(Object obj)
  + void insertElementAt(Object obj,int index)
  + void setElementAt(Object obj,int index)
  + void removeElement(Object obj)
  + void removeAllElements()
* ListIterator接口
* List额外提供了一个listIterator()方法，该方法返回一个ListIterator对象，ListIterator接口继承了Iterator接口，提供了专门操作List的方法：
* void add()
* boolean hasPrevious()
* Object previous()
* Boolean hasNext()
* Object next()
* Iterator和ListIterator主要区别（了解）
  + ListIterator和Iterator都有hasNext()和next()方法，可以实现**顺序向后遍历**。但是ListIterator有hasPrevious()和previous()方法，可以实现**逆向（顺序向前）遍历**。Iterator就不可以。
  + ListIterator可以定位当前的索引位置，nextIndex()和previousIndex()可以实现。Iterator没有此功能。
  + ListIterator有add()方法，可以向List中插入对象，而Iterator不能。
  + 都可实现删除对象，但是ListIterator可以实现对象的修改，set()方法可以实现。Iterator仅能遍历，不能修改。因为ListIterator的这些功能，可以实现对LinkedList等List数据结构的操作。