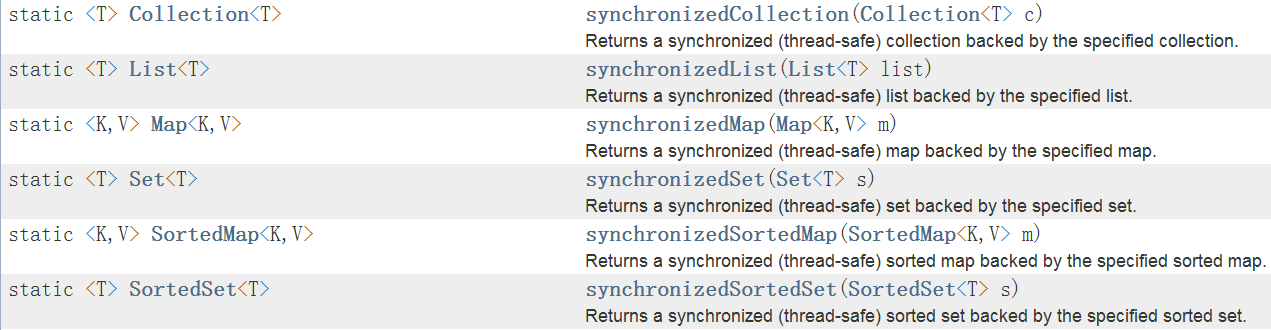
* **操作集合的工具类——Collections**
* Collections是一个操作Set、List和Map等集合的工具类。
* Collections中提供了一系列**静态的方法**对集合元素进行**排序**、**查询**和**修改**等操作，还提供了对集合对象**设置不可变**、对集合对象实现**同步控制**等方法。
* **排序操作：（均为static方法）**
* reverse(List)——反转List中元素的顺序。
* shuffle(List)——对List集合元素进行随机排序。
* sort(List)——根据元素的自然顺序对指定List集合元素按升序排序。
* sort(List，Comparator)——**根据指定的Comparator产生的顺序对List集合元素进行排序**。
* swap(List，int，int)——将指定list集合中的i处元素和j处元素进行交换。

@Test  
public void test1() {  
 List list = new ArrayList();  
 list.add(123);  
 list.add(456);  
 list.add(12);  
 list.add(78);  
 System.***out***.println(list);  
  
 // 通过类名直接调用该类的静态方法。  
 // 反转  
 Collections.reverse(list);  
 System.***out***.println(list);  
  
 // 随机排序  
 Collections.shuffle(list);  
 System.***out***.println(list);  
  
 // 升序排序  
 Collections.sort(list);  
 System.***out***.println(list);  
  
 // 交换元素  
 Collections.swap(list, 0, 2);  
 System.***out***.println(list);  
}

* **查找、替换**
* Object max(Collection)——根据元素的自然顺序，返回给定集合中的最大元素。
* Object max(Collection，Comparator)——根据Comparator指定的顺序，返回给定集合中的最大元素。
* Object min(Collection)
* Object min(Collection，Comparator)
* int frequency(Collection，Object)——返回指定集合中指定元素的出现次数。
* void copy(List dest,List src)——将src中的内容复制到dest中。
* boolean replaceAll(List list，Object oldVal，Object newVal)——使用新值替换List对象的所有旧值。

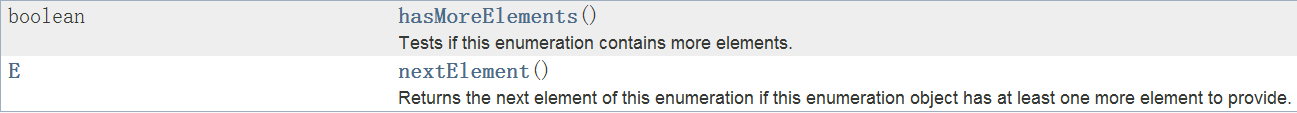
@Test  
public void test2() {  
 List list = new ArrayList();  
 list.add(123);  
 list.add(456);  
 list.add(12);  
 list.add(78);  
 list.add(456);  
  
 // 查找最大元素  
 Object max = Collections.max(list);  
 System.***out***.println(max);  
  
 // 查找最小元素  
 Object min = Collections.min(list);  
 System.***out***.println(min);  
  
 // 返回指定元素的个数  
 int count = Collections.frequency(list,4567);  
 System.***out***.println(count);  
  
 // 将src中的内容复制到dest中  
 //List list1 = new ArrayList();//错误的方式  
 List list1 = Arrays.asList(new Object[list.size()]);  
 Collections.copy(list1,list);  
 System.***out***.println(list1);  
}

* **同步控制**
* Collections类中提供了多个synchronizedXxx()方法，该方法可使**将指定集合包装成线程同步的集合**，从而可以解决多线程并发访问集合时的线程安全问题。



// 线程安全  
List list2 = Collections.synchronizedList(list);

* **Enumeration（了解即可）**
* Enumeration接口是Iterator迭代器的“古老版本”。



@Test  
public void test3() {  
 Enumeration stringEnum = new StringTokenizer("a-b\*c-d-e-g", "-");  
 while (stringEnum.hasMoreElements()) {  
 Object obj = stringEnum.nextElement();  
 System.***out***.println(obj);  
 }  
  
}