Collections工具類

1. 作用： *操作Collection、Map的工具类*
2. 常用方法：

*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  
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*



說明:ArrayList和HashMap都是線程不安全的，如果程序要求線程安全，我們可以將ArrayList和HashMap轉為線程安全的。使用*synchronizedList(List list)和synchronizedList(Map map)*

public void test2(){  
 List list = new ArrayList();  
 list.add(123);  
 list.add(43);  
 list.add(765);  
 list.add(-97);  
 list.add(0);  
*//报异常：IndexOutOfBoundsException("Source does not fit in dest")  
// List dest = new ArrayList();  
// Collections.copy(dest,list);  
 //正确的：* List dest = Arrays.*asList*(new Object[list.size()]);  
 System.*out*.println(dest.size());*//list.size();* Collections.*copy*(dest,list);  
 System.*out*.println(dest);  
 */\*  
 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  
 \*/  
 //返回的list1即为线程安全的List* List list1 = Collections.*synchronizedList*(list);  
 }

1. 面試題：

*Collection 和 Collections的区别？*