

遍历List方法一：普通for循环

最慢

1 2 3 4	<pre>for(int i=0;i<list.size();i++){//list为集合的对象名 String temp = (String)list.get(i); System.out.println(temp); }</pre>
<div>◀</div> <div>▶</div>	

遍历List方法二：增强for循环(使用泛型!)

1 2 3	<pre>for (String temp : list) { System.out.println(temp); }</pre>
<div>◀</div> <div>▶</div>	

遍历List方法三：使用Iterator迭代器(1)

1 2 3 4	<pre>for(Iterator iter= list.iterator();iter.hasNext();){ String temp = (String)iter.next(); System.out.println(temp); }</pre>
<div>◀</div> <div>▶</div>	

遍历List方法四：使用Iterator迭代器(2)

1 2 3 4 5 6	<pre>Iterator iter =list.iterator(); while(iter.hasNext()){ Object obj = iter.next(); iter.remove();//如果要遍历时，删除集合中的 这种方式！ System.out.println(obj); }</pre>
<div>◀</div> <div>▶</div>	

遍历Set方法一：增强for循环

1 2 3	<pre>for(String temp:set){ System.out.println(temp); }</pre>
<div>◀</div> <div>▶</div>	

遍历Set方法二：使用Iterator迭代器

1 2 3 4	<pre>for(Iterator iter = set.iterator();iter.hasNext();){ String temp = (String)iter.next(); System.out.println(temp); }</pre>
<div>◀</div> <div>▶</div>	

遍历Map方法一：根据key获取value

1 2 3 4 5	<pre>Map<Integer, Man> maps = new HashMap<Integer, Man>(); Set<Integer> keySet = maps.keySet(); for(Integer id : keySet){ System.out.println(maps.get(id).name); }</pre>
<div>◀</div> <div>▶</div>	

遍历Map方法二：使用entrySet

1 2 3 4	<pre>Set<Entry<Integer, Man>> ss = maps.entrySet(); for (Iterator iterator = ss.iterator(); iterator.hasNext();) { Entry e = (Entry) iterator.next(); System.out.println(e.getKey("+e.getValue()); }</pre>
<div>◀</div> <div>▶</div>	

排序

有两种排序方式：

- 自然排序：

我们的元素必须实现 Comparable 接口.可比较的.实现 compareTo 方法.

- 比较器排序：

我们需要自定义类,实现Comparator接口,这个类就是比较器实现 compare 方法.