Collections工具类：

```
ArrayList<String> arrayList = new ArrayList<>();
```

向集合中添加元素

```
Collections.addAll(arrayList,"张四","李四","王麻子","李四");
System.out.println(arrayList);
```

反向指定元素顺序

```
Collections.reverse(arrayList);
System.out.println(arrayList);
```

指定某个位置的元素进行替换

```
Collections.replaceAll(arrayList,"李四","替换");
System.out.println(arrayList);
```

生成空的Set List Map

```
List<Object> emptyList = Collections.emptyList();
Map<Object, Object> emptyMap = Collections.emptyMap();
Set<Object> set = Collections.emptySet();
System.out.println(arrayList);
```

加入多个元素

```
Set<Integer> set1 = new HashSet<>();
Collections.addAll(set1,10,20,15,18,29,26);
System.out.println(set1);
```

比大小

```
Integer max =Collections.max(set1);
Integer min = Collections.min(set1);
System.out.println("max = " + max);
System.out.println("min = " + min);
```

集合轮换

```
System.out.println(arrayList);
Collections.rotate(arrayList,1);
System.out.println(arrayList);
```

交换两个元素的位置

```
Collections.swap(arrayList,0,1);
System.out.println(arrayList);
```

打乱集合中元素

```
Collections.shuffle(arrayList);
System.out.println(arrayList);
```

关于泛型数据类型

list中的元素是任意类型
```
static void test(List list) {
    System.out.println(list);
}
```
list中的元素是任意类型
```
static void test(List<?> list) {
    System.out.println(list);
}
```
lis中的元素是Girl或Girl的子类
```
static void test(List<? extends Girl> list) {
    System.out.println(list);
}
```
lis中的元素是Girl或Girl的父类
```
static void test(List<? super Girl> list) {
    System.out.println(list);
}
```