CollectionUtils工具类

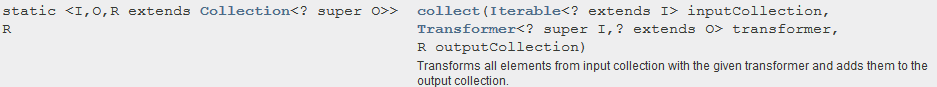
1. CollectionUtils是一个 工具类，主要是用来方便处理java.util中的集合。直接继承于java.lang.Object类，存在于包org.apache.commons.collections4中。
2. 既然是一个工具类，那么CollectionUtils类本身就不需要创建对象，所以CollectionUtils类对外不提供构造方法，也没有静态方法返回本身对象。

CollectionUtils类作为工具类，主要有两个功能：

1. 产生一些集合对象，对集合进行包装处理；
2. 提供一些方法，直接调用即可。
3. CollectionUtils中的成员方法都是**静态方法**，用于操作集合。
4. **collect方法**：有很多重载。返回一个集合对象。

如传入一个集合和一个转换器。

即collect（集合，转换器）。



注意：Collection接口继承于Iterable<E>接口，可以利用Iterable接收任何一个集合对象。

1. 等等。
2. 对集合的操作：集合间的**并集、交集、差集**。

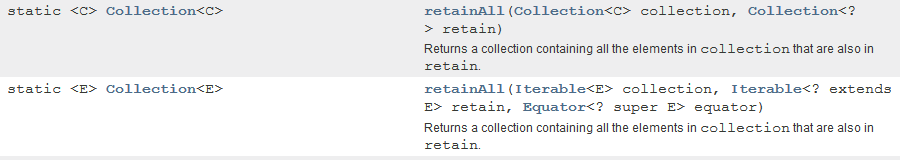
利用CollectionUtils里提供的方法实现集合间的并、交、差等操作。

1. 并集：**union方法**：返回一个Collection对象。

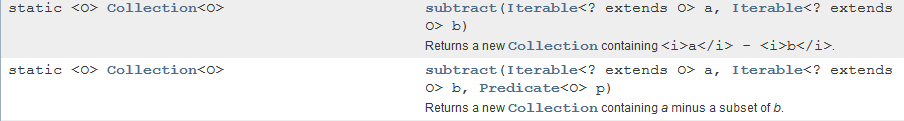


1. 交集：intersection方法或retainAll方法。





1. **差集：substract方法**。



1. 示例：

**public** **static** **void** main(String[] args) {

Set<Integer> set1 = **new** HashSet<Integer>();

Set<Integer> set2 = **new** HashSet<Integer>();

set1.add(111);

set1.add(222);

set1.add(333);

set2.add(222);

set2.add(333);

set2.add(444);

Collection<Integer> unionCollection = CollectionUtils.*union*(set1, set2);

System.***out***.println("\*\*\*\*\*并集\*\*\*\*\*\*\*\*\*\*\*");

*IteratorOutputPrint*(unionCollection);

Collection<Integer> inter = CollectionUtils.*intersection*(set1, set2);

inter = CollectionUtils.*retainAll*(set1, set2);

System.***out***.println("\*\*\*\*\*交集\*\*\*\*\*\*\*\*\*\*\*");

*IteratorOutputPrint*(inter);

Collection<Integer> subs = CollectionUtils.*subtract*(set1, set2);

System.***out***.println("\*\*\*\*\*差集\*\*\*\*\*\*\*\*\*\*\*");

*IteratorOutputPrint*(subs);

}

**private** **static** **void** IteratorOutputPrint(Collection<Integer> coll) {

Iterator<Integer> iterator = coll.iterator();

**while** (iterator.hasNext()) {

System.***out***.println(iterator.next());

} }

结果：