



链滴

java 中 list 集合的操作 (交集, 并集, 差集)

作者: [zspmsir](#)

原文链接: <https://ld246.com/article/1658454679473>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

import static java.util.stream.Collectors.toList;
import java.util.ArrayList;
import java.util.List;
/**
 * @ClassName: T
 * @Author: 技术solo
 * @Date: 2020-01-04
 * @Description: list集合的操作(交集, 并集, 差集)
 */
public class T {

    public static void main(String[] args) {
        List<String> list1 = new ArrayList();
        list1.add("1111");
        list1.add("2222");
        list1.add("3333");

        List<String> list2 = new ArrayList();
        list2.add("3333");
        list2.add("4444");
        list2.add("5555");

        // 交集
        List<String> intersection = list1.stream().filter(item -> list2.contains(item)).collect(toList(
);
        System.out.println("---得到交集 intersection---");
        intersection.parallelStream().forEach(System.out :: println);

        // 差集 (list1 - list2)
        List<String> reduce1 = list1.stream().filter(item -> !list2.contains(item)).collect(toList());
        System.out.println("---得到差集 reduce1 (list1 - list2)---");
        reduce1.parallelStream().forEach(System.out :: println);

        // 差集 (list2 - list1)
        List<String> reduce2 = list2.stream().filter(item -> !list1.contains(item)).collect(toList());
        System.out.println("---得到差集 reduce2 (list2 - list1)---");
        reduce2.parallelStream().forEach(System.out :: println);

        // 并集
        List<String> listAll = list1.parallelStream().collect(toList());
        List<String> listAll2 = list2.parallelStream().collect(toList());
        listAll.addAll(listAll2);
        System.out.println("---得到并集 listAll---");
        listAll.parallelStream().forEach(System.out :: println);

        // 去重并集
        List<String> listAllDistinct = listAll.stream().distinct().collect(toList());
        System.out.println("---得到去重并集 listAllDistinct---");
        listAllDistinct.parallelStream().forEach(System.out :: println);

        System.out.println("---原来的List1---");
        list1.parallelStream().forEach(System.out :: println);
        System.out.println("---原来的List2---");
        list2.parallelStream().forEach(System.out :: println);
    }
}

```

```
    // 一般有filter 操作时, 不用并行流parallelStream ,如果用的话可能会导致线程安全问题  
  }  
}
```