

导航

博客园
首页
新随笔
联系
订阅
管理

XML

<	2018年9月						>
日	一	二	三	四	五	六	
26	27	28	29	30	31	1	
2	3	4	5	6	7	8	
9	10	11	12	13	14	15	
16	17	18	19	20	21	22	
23	24	25	26	27	28	29	
30	1	2	3	4	5	6	

公告

昵称：乌云上
园龄：1年9个月
粉丝：1
关注：6
+加关注

搜索

<input type="text"/>	找找看
<input type="text"/>	谷歌搜索

常用链接

guava快速入门（二）

Guava工程包含了若干被Google的 Java项目广泛依赖 的核心库，例如：集合 [collections]、缓存 [caching]、原生类型支持 [primitives support]、并发库 [concurrency libraries]、通用注解 [common annotations]、字符串处理 [string processing]、I/O 等等。

guava类似Apache Commons工具集

Collection

不可变集合

不可变对象有很多优点，包括：

- 当对象被不可信的库调用时，不可变形式是安全的；
- 不可变对象被多个线程调用时，不存在竞态条件问题
- 不可变集合不需要考虑变化，因此可以节省时间和空间。所有不可变的集合都比它们的可变形式有更好的内存利用率（分析和测试细节）；
- 不可变对象因为有固定不变，可以作为常量来安全使用。

JDK也提供了Collections.unmodifiableXXX方法把集合包装为不可变形式，但：

- 笨重而且累赘：不能舒适地用在所有想做防御性拷贝的场景；
- 不安全：要保证没人通过原集合的引用进行修改，返回的集合才是事实上不可变的；
- 低效：包装过的集合仍然保有可变集合的开销，比如并发修改的检查、散列表的额外空间，等等。

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

日常积累(110)
Java(26)
数据库(24)
SqlServer(22)
环境搭建(21)
SQL(19)
jQuery(18)
php(16)
Node.js学习(15)
JavaScript(15)
更多

随笔分类

.NET(6)
Ajax(1)
BAT(1)
CASCO卡斯柯
Dos命令(1)
Excel(2)
Game(1)
Java(27)
jQuery(17)
jQuery easyUI(1)
Node.js(16)
PHP(14)
Python(15)
SQL(26)
WeUI(10)
总结(4)

随笔档案

2018年9月 (19)

创建不可变集合方法：

- copyOf方法，如ImmutableSet.copyOf(set);
- of方法，如ImmutableSet.of("a", "b", "c")或 ImmutableMap.of("a", 1, "b", 2);
- Builder工具

```
1 import com.google.common.collect.ImmutableList;
2 import com.google.common.collect.ImmutableSet;
3
4 public class ImmutableDemo {
5     public static void main(String[] args) {
6         ImmutableSet<String> set = ImmutableSet.of("a", "b", "c", "d");
7         ImmutableSet<String> set1 = ImmutableSet.copyOf(set);
8         ImmutableSet<String> set2 = ImmutableSet.<String> builder().addAll(set).add("e").build();
9         ImmutableList<String> list = set.asList();
10
11         System.out.println("set:"+set);
12         System.out.println("set1:"+set1);
13         System.out.println("set2:"+set2);
14         //set.add("f");//java.lang.UnsupportedOperationException
15     }
16 }
```

返回：

```
1 set:[a, b, c, d]
2 set1:[a, b, c, d]
3 set2:[a, b, c, d, e]
```

新型集合类

Multiset

2018年8月 (54)
2018年7月 (6)
2018年6月 (43)
2018年5月 (13)
2018年4月 (6)
2018年3月 (1)
2018年1月 (1)

积分与排名

积分 - 5839
排名 - 58324

最新评论

1. Re:如何把ASP.NET MVC
项目部署到本地IIS上

@~雨落忧伤~引用别人能访问吗? 可以的, 前提是, 访问者和你的电脑在同一个局域网网上, 并且知道你的电脑ip地址, 比如, 你用手机连上你电脑开出来的wifi, 然后再手机的浏览器中输入ip地址也是可以访问的。.....

--乌云上

2. Re:如何把ASP.NET MVC
项目部署到本地IIS上
别人能访问吗?

--~雨落忧伤~

阅读排行榜

1. 常见浏览器User-Agent
大全(773)
2. Python——第一个
python程序
helloworld(654)
3. jquery里判断数组内是否
包含了指定的值或元素的方法(368)

Multiset可统计一个词在文档中出现了多少次

[+ View Code](#)

返回:

```
1 [a x 5, lucas x 2]
2 5
3 7
4 2
5 []
```

Multimap

Multimap可以很容易地把一个键映射到多个值。换句话说, Multimap是把键映射到任意多个值的一般方式。

```
1 import java.util.Collection;
2 import java.util.Map;
3
4 import com.google.common.collect.HashMultimap;
5 import com.google.common.collect.Multimap;
6
7 public class MultiMapDemo {
8
9     public static void main(String[] args) {
10         Multimap<String, Integer> map = HashMultimap.create(); // Multimap是把键映射到任意多个值的一般方式
11         map.put("a", 1); // key相同时不会覆盖原value
12         map.put("a", 2);
13         map.put("a", 3);
14         map.put("b", 1);
15         map.put("b", 2);
16         System.out.println(map); // {a=[1, 2, 3]}
17         System.out.println(map.get("a")); // 返回的是集合
```

4. 如何把ASP.NET MVC项目部署到本地IIS上(329)
5. jquery操作radio单选按钮, 实现取值, 动态选中, 动态删除的各种方法(328)

评论排行榜

1. 如何把ASP.NET MVC项目部署到本地IIS上(2)

```

18         System.out.println(map.size()); // 返回所有"键-单个值映射"的个数,而非不同键的个数
19         System.out.println(map.keySet().size()); // 返回不同key的个数
20         Map<String, Collection<Integer>> mapView = map.asMap();
21         System.out.println(mapView);
22     }
23 }

```

返回:

```

1  {a=[1, 2, 3], b=[1, 2]}
2  [1, 2, 3]
3  5
4  2
5  {a=[1, 2, 3], b=[1, 2]}

```

BiMap

BiMap

```

1  import com.google.common.collect.BiMap;
2  import com.google.common.collect.HashBiMap;
3
4  public class BitMapDemo {
5
6      public static void main(String[] args) {
7          BiMap<String, String> biMap = HashBiMap.create();
8          biMap.put("sina", "sina.com");
9          biMap.put("qq", "qq.com");
10         biMap.put("sina", "sina.cn"); // 会覆盖原来的value
11         /*
12          * 在BiMap中,如果你想把键映射到已经存在的值, 会抛出IllegalArgumentException异常
13          * 如果对特定值,你想要强制替换它的键, 请使用 BiMap.forcePut(key, value)
14          */

```

```
15 // 抛出异常java.lang.IllegalArgumentException: value already present: qq.com
16 //biMap.put("tecent", "qq.com");
17 biMap.forcePut("tecent", "qq.com"); // 强制替换key
18 System.out.println(biMap);
19 System.out.println(biMap.inverse()); // 翻转value-key
20 System.out.println(biMap.inverse().get("sina.cn")); // 通过value找key
21 System.out.println(biMap.inverse().inverse() == biMap); // true
22
23 }
24
25 }
```

返回:

```
1 {sina=sina.cn, tecent=qq.com}
2 {sina.cn=sina, qq.com=tecent}
3 sina
4 true
```

Table

Table它有两个支持所有类型的键：“行”和“列”。

```
1 import java.util.Set;
2
3 import com.google.common.collect.HashBasedTable;
4 import com.google.common.collect.Table;
5
6 public class TableDemo {
7
8     public static void main(String[] args) {
9         // 记录学生在某门课上的成绩
10         Table<String, String, Integer> table = HashBasedTable.create();
```

```
11     table.put("jack", "java", 100);
12     table.put("jack", "c", 90);
13     table.put("mike", "java", 93);
14     table.put("mike", "c", 100);
15     Set<Table.Cell<String, String, Integer>> cells = table.cellSet();
16     for (Table.Cell<String, String, Integer> cell : cells) {
17         System.out.println(cell.getRowKey() + " " + cell.getColumnKey() + " " + cell.getValue());
18     }
19     System.out.println("-----");
20     System.out.println(table.row("jack"));
21     System.out.println(table);
22     System.out.println(table.rowKeySet());
23     System.out.println(table.columnKeySet());
24     System.out.println(table.values());
25 }
26
27 }
```

返回:

```
1  mike java 93
2  mike c 100
3  jack java 100
4  jack c 90
5  -----
6  {java=100, c=90}
7  {mike={java=93, c=100}, jack={java=100, c=90}}
8  [mike, jack]
9  [java, c]
10 [93, 100, 100, 90]
```

Collections2

filter () : 只保留集合中满足特定要求的元素

```
1  import java.util.Collection;
2  import java.util.List;
3
4  import com.google.common.collect.Collections2;
5  import com.google.common.collect.Lists;
6
7  public class FilterDemo {
8
9      public static void main(String[] args) {
10         List<String> list = Lists.newArrayList("moon", "dad", "refer", "son");
11         Collection<String> palindromeList = Collections2.filter(list, input -> {
12             return new StringBuilder(input).reverse().toString().equals(input); // 找回文串
13         });
14         System.out.println(palindromeList); // palindromeList 回文, 也叫做镜文, 正写倒写一样, 如mam
15     }
16
17 }
```

返回:

```
1  [dad, refer]
```

transform () : 类型转换

```
1  import java.text.SimpleDateFormat;
2  import java.util.Collection;
3  import java.util.Set;
4
5  import com.google.common.base.Function;
6  import com.google.common.collect.Collections2;
7  import com.google.common.collect.Sets;
8
9  public class TransformDemo {
```

```
10
11     public static void main(String[] args) {
12         Set<Long> times = Sets.newHashSet();
13         times.add(91299990701L);
14         times.add(9320001010L);
15         times.add(9920170621L);
16         Collection<String> timeStrCol = Collections2.transform(times, new Function<Long, String>() {
17
18             @Override
19             public String apply( Long input) {
20                 return new SimpleDateFormat("yyyy-MM-dd").format(input);
21             }
22         });
23         System.out.println(timeStrCol);
24     }
25
26 }
```

返回:

```
1 [1970-04-19, 1970-04-26, 1972-11-23]
```

多个Function组合

```
1 import java.util.Collection;
2 import java.util.List;
3
4 import com.google.common.base.Function;
5 import com.google.common.base.Functions;
6 import com.google.common.collect.Collections2;
7 import com.google.common.collect.Lists;
8
9 public class TransformDemo2 {
10
11     public static void main(String[] args) {
```



```
12     List<String> list = Lists.newArrayList("abcde", "good", "happiness");
13     // 确保容器中的字符串长度不超过5
14     Function<String, String> f1 = new Function<String, String>() {
15
16         @Override
17         public String apply( String input) {
18             return input.length() > 5 ? input.substring(0, 5) : input;
19         }
20     };
21     // 转成大写
22     Function<String, String> f2 = new Function<String, String>() {
23
24         @Override
25         public String apply( String input) {
26             return input.toUpperCase();
27         }
28     };
29     Function<String, String> function = Functions.compose(f1, f2);
30     Collection<String> results = Collections2.transform(list, function);
31     System.out.println(results);
32 }
33
34 }
```

返回:

```
1 [ABCDE, GOOD, HAPPY]
```

集合操作: 交集、差集、并集

```
1 import java.util.Set;
2
3 import com.google.common.collect.Sets;
```

```
4
5 public class CollectionsDemo {
6
7     public static void main(String[] args) {
8         Set<Integer> set1 = Sets.newHashSet(1, 2, 3, 4, 5);
9         Set<Integer> set2 = Sets.newHashSet(3, 4, 5, 6);
10        Sets.SetView<Integer> inter = Sets.intersection(set1, set2); // 交集
11        System.out.println(inter);
12        Sets.SetView<Integer> diff = Sets.difference(set1, set2); // 差集,在A中不在B中
13        System.out.println(diff);
14        Sets.SetView<Integer> union = Sets.union(set1, set2); // 并集
15        System.out.println(union);
16    }
17
18 }
```

返回:

[+ View Code](#)

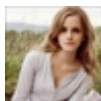
分类: [Java](#)

标签: [日常积累](#), [Java](#), [guava](#), [guava基础](#)

好文要顶

关注我

收藏该文



乌云上

关注 - 6

粉丝 - 1

[+加关注](#)

0

0

[« 上一篇: 13个开发者技能必知必会!](#)[» 下一篇: guava快速入门（三）](#)

posted on 2018-06-20 16:48 乌云上 阅读(16) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。



最新IT新闻:

- 约谈整改28天后探访：北京租房仍存虚假房源 中介自爆商业内幕
 - 华为手机芯片麒麟980发布：投入3亿美元 用7纳米工艺
 - 太空殖民拯救人类？灭绝可能性或上升 还需三思而行
 - 二手车交易平台乱象丛生：隐形高收费 刷单造繁荣
 - 谷歌拒派最高领导参加听证会 美参议院：深感失望
- » 更多新闻...



华为全联接大会 | 上海 | 2018.10.10-12

[大会门票+云服务器] 专属套餐0.35折起



最新知识库文章:

- 如何招到一个靠谱的程序员
- 一个故事看懂“区块链”
- 被踢出去的用户
- 成为一个有目标的学习者
- 历史转折中的“杭派工程师”
- » 更多知识库文章...

Powered by:

博客园

Copyright © 乌云上