

# woonu

业精于勤，荒于嬉；行成于思，毁于随。——韩愈

博客园 首页 新随笔 联系 订阅 管理

随笔-32 文章-0 评论-2

## 集合框架学习之Guava Collection

开源工具包：

- Guava : Google Collection
- Apache: Commons Collecton

### 1.1 Google Collections

Guava:google的工程师利用传说中的“20%时间”开发的集合库，它是对jdk提供的扩展，提供了很多使用的类来简化代码

jar包: <https://code.google.com/p/guava-libraries/>

源码下载：

- 下载git工具：（易于本地增加分支和分布式的特性）
  - msysgit:<http://code.google.com/p/msysgit/downloads/list>

- 命令: `git clone 网络路径本地文件夹`（不存在或空文件夹）



/\*\*

\* Guava集合框架

\* @author qjc

\*

\* 2016-3-12

## 公告

昵称: woonu

园龄: 2年6个月

粉丝: 19

关注: 12

+加关注

|    |         |    |    |    |    |    |   |
|----|---------|----|----|----|----|----|---|
| <  | 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  |   |

## 搜索

 谷歌搜索

## 常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

## 最新随笔

1. 记录weiye项目上线遇到的一些问题
2. layer弹框插件使用
3. 使用Ztree新增角色和编辑角色回显

```

*/
public class Demo {
    /**
     * 只读设置
     */
    @Test
    public void testGuava1() {
        List<String> list = new ArrayList<>();
        list.add("a");
        list.add("b");
        list.add("c");
        //对原有的list进行包装, 相当于原有List的一个视图, 快照, 不够安全
        List<String> readList = Collections.unmodifiableList(list);
        //java.lang.UnsupportedOperationException
        // readList.add("d");
        list.add("d"); //改变原有List 试图也一起改变
        //对比查看初始化List guava对只读设置 安全可靠, 并且相对简单
        List<String> immutableList = ImmutableList.of("a", "b", "c");
        immutableList.add("d"); //java.lang.UnsupportedOperationException
    }
    /**
     * 函数式编程:过滤器
     */
    @Test
    public void testGuava2() {
        //创建List 静态初始化
        List<String> list = Lists.newArrayList("moon", "son", "dad", "refer");
        //找出回文 palindronme backwoods mirror words
        Collection<String> palindromeList = Collections2.filter(list, new
        Predicate<String>() {
            @Override
            public boolean apply(String input) {
                //业务逻辑
                return new StringBuilder(input).reverse().toString().equals(input);
            }
        });
    }
}

```

4. Linux下安装JDK7和TomCat7
5. PLSQL程序设计(Oracle)
6. 存储过程, 存储函数(Oracle)
7. 【深入spring】IoC容器的实现
8. 【spring基础】spring声明式事务详解
9. 【spring基础】spring与jdbc整合详解
10. 【spring基础】AOP概念与动态代理详解

## 随笔分类(30)

Java并发编程

Java基础知识(14)

JVM(1)

Linux(2)

框架知识(8)

设计模式(2)

数据结构及算法

数据库(3)

网络编程

## 随笔档案(32)

2016年10月 (3)

2016年9月 (1)

2016年6月 (2)

2016年4月 (1)

2016年3月 (24)

2015年9月 (1)

## 积分与排名

积分 - 22683

排名 - 20201

## 最新评论

1. Re:存储过程, 存储函数(Oracle)

简单明了啊 可惜不是PHP写的语法, 只能知道有这么回事

--贝斯学长

2. Re:struts2基础之配置详解

```
        }
    });
    for(String temp : palindromeList){
        System.out.println(temp);
        //输出结果: dad refer
    }
}
/**
 * 函数式编程:转换
 */
@Test
public void testGuava3(){
    Set<Long> timeSet = Sets.newHashSet();
    //类型转换
    timeSet.add(19990701L);
    timeSet.add(20080808L);
    timeSet.add(20161212L);
    Collection<String> timeStrCol = Collections2.transform(timeSet, new
Function<Long,String>(){
        @Override
        public String apply(Long input) {
            return new SimpleDateFormat("yyyy-MM-dd").format(input);
        }
    });
    for(String temp : timeStrCol){
        System.out.println(temp);
    }
}
/**
 * 组合式函数编程
 * 确保容器中的字符串长度不超过5, 超过进行截取, 然后全部大写
 */
@Test
public void testGuava4(){
```

很详细, 值得收藏, 以后开发中如果遇到可以回来查询。

--冷豪

### 阅读排行榜

1. JVM核心之JVM运行和类加载全过程(15932)
2. 存储过程, 存储函数(Oracle)(10494)
3. 【spring基础】spring与jdbc整合详解(2836)
4. JDBC事务与事务隔离级别详解(2277)
5. 【spring基础】AOP概念与动态代理详解(1854)

### 评论排行榜

1. 存储过程, 存储函数(Oracle)(1)
2. 【struts2基础】配置详解(1)

### 推荐排行榜

1. 【struts2基础】配置详解(3)
2. PLSQL程序设计(Oracle)(3)
3. 【spring基础】AOP概念与动态代理详解(2)
4. 存储过程, 存储函数(Oracle)(1)
5. JDBC事务与事务隔离级别详解(1)

```
List<String> list = Lists.newArrayList("abcde", "good", "happiness");
//确保容器中的字符串长度不超过5
Function<String, String> f1 = new Function<String, String>() {
    @Override
    public String apply(String input) {
        return input.length() > 5 ? input.substring(0, 5) : input;
    }
};
//转成大写
Function<String, String> f2 = new Function<String, String>() {
    @Override
    public String apply(String input) {
        return input.toUpperCase();
    }
};
//String = f2(f1(String))
Function<String, String> f = Functions.compose(f1, f2);
Collection<String> resultCol = Collections2.transform(list, f);
for (String str : resultCol) {
    System.out.println(str);
    /*
     * 输出结果:
     *
     *      ABCDE
     *      GOOD
     *      HAPPY
     */
}
}
/**
 * 加入约束：非空、长度验证
 */
@Test
public void testGuava5() {
    Set<String> sets = Sets.newHashSet();
```

```
//创建约束
Constraint<String> constraint = new Constraint<String>() {

    @Override
    public String checkElement(String element) {
        //非空验证
        Preconditions.checkNotNull(element);
        //长度验证 5~20位字符串
        Preconditions.checkArgument(element.length()>=5 && element.length()
<20);

        return element;
    }
};

Set<String> cs = Constraints.constrainedSet(sets, constraint);
//      cs.add(null);      //java.lang.NullPointerException
//      cs.add("abcd"); //java.lang.IllegalArgumentException
cs.add("abcde");//ok
}
/**
 * 集合操作：交集、差集、并集
 */
@Test
public void testGuava6(){
    Set<Integer> set1 =Sets.newHashSet(1,2,3,4);
    Set<Integer> set2 =Sets.newHashSet(3,4,5,6);
    //交集
    SetView<Integer> inter = Sets.intersection(set1, set2);
    //差集
    SetView<Integer> diff = Sets.difference(set1, set2);
    //并集
    SetView<Integer> union = Sets.union(set1, set2);
    //遍历输出:交集 : 3,4 差集: 1, 2, 5,6 并集: 1,2,3,4,5,6,
}
/**
```

```
* 统计单词出现的次数
* 1、HashMap 分拆存储+面向对象思维 --->判断
* 2、Multiset:无序可重复 .count() 增强可读性+操作简单
*/

@Test
public void testGuava7(){
    String str = "this is a cat and that is mice where is the food";
    //分割字符串
    String[] arr = str.split(" ");
    //存储到Multiset中
    Multiset<String> set = HashMultiset.create();
    for(String ss:arr){
        set.add(ss);
    }
    //获取所有的单词set
    Set<String> letters = set.elementSet();
    for(String temp:letters){
        System.out.println(temp+": "+set.count(temp));
    }
}

/**
* 分析查看每个教师教授的没门课程
* Multimap key也可以重复
*/

@Test
public void testGuava8(){
    Map<String, String> map = new HashMap<String, String>();
    //加入测试数据
    map.put("改革开放", "邓爷爷");
    map.put("三个代表", "江主席");
    map.put("科学发展观", "胡主席");
    map.put("和谐社会", "胡主席");
    map.put("八荣八耻", "胡主席");
    //Multimap
```

```

Multimap<String, String> teacher = ArrayListMultimap.create();
//迭代器
Iterator<Map.Entry<String, String>> it = map.entrySet().iterator();
while(it.hasNext()){
    Map.Entry<String, String> entry = it.next();
    String key = entry.getKey(); //课程
    String value = entry.getValue(); //教师
    //教师--->课程
    teacher.put(value, key);
}
//查看Multimap
Set<String> keySet = teacher.keySet();
for(String key:keySet){
    Collection<String> col = teacher.get(key);
    System.out.println(key+": "+col);
}
}
/**
 * HashMap 键唯一,值可以重复
 * BiMap:双向Map(bidirectional Map) 键与值不能重复(unique -valued map)
 */
@Test
public void testGuava9(){
    BiMap<String, String> biMap = HashBiMap.create();
    biMap.put("sina", "@sina.com");
    biMap.put("qq", "@qq.com");
    //通过邮箱找用户
    String user = biMap.inverse().get("@sina.com");
    System.out.println(user);
    System.out.println(biMap.inverse().inverse()==biMap);
}
/**
 * 双键的Map -->Table --->rowKey+columnKye+value
 */

```

```
@Test
public void testGuava10(){
    Table<String, String, Integer> table = HashBasedTable.create();
    //测试数据
    table.put("龙傲天", "java", 50);
    table.put("龙傲天", "oracle", 60);
    table.put("福尔康", "java", 70);
    table.put("福尔康", "oracle", 100);
    //所有的行数据
    Set<Cell<String, String, Integer>> cells = table.cellSet();
    for(Cell<String, String, Integer> temp : cells){

        System.out.println(temp.getRowKey()+":"+temp.getColumnKey()+":"+temp.getValue());
    }
    /*
    龙傲天:java:50
    龙傲天:oracle:60
    福尔康:java:70
    福尔康:oracle:100*/
    System.out.println("=====学生查看成绩=====");
    System.out.print("学生\t");
    //所有的课程
    Set<String> cours = table.columnKeySet();
    for(String t : cours){
        System.out.print(t+"\t");
    }
    System.out.println();
    //所有的学生
    Set<String> stus = table.rowKeySet();
    for(String stu:stus){
        System.out.print(stu+"\t");
        Map<String,Integer> scores = table.row(stu);
        for(String c:cours){
            System.out.print(scores.get(c)+"\t");
        }
    }
}
```



```
        }  
        System.out.println();  
    }  
    /*  
    学生      java      oracle  
    龙傲天    50        60  
    福尔康    70        100*/  
}  
}
```



### 小结:

1.只读设置:immutableList

2.函数式编程:解耦

1) predicate

2)Function

工具:

Collections2.filter() 过滤器

Collections2.transfer() 转换

Functions.compose() 组合式函数编程

3. 加入约束条件:非空 长度验证

Constraint

Preconditions

4. 集合的操作:交集差集并集

Sets.intersection()

Sets.difference()

Sets.union()

5. Multiset    Multimap    BiMap

6. Table

分类: [Java基础知识](#)

好文要顶

关注我

收藏该文



[woonu](#)

[关注 - 12](#)

[粉丝 - 19](#)

[+加关注](#)

0

0

« 上一篇: [集合框架学习之其他容器](#)

» 下一篇: [线程基础](#)

posted @ 2016-03-12 19:44 woonu 阅读(1743) 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](#)。

腾讯云

# 学生服务器体验套餐

## 10元/月

· 1核2G · 1M带宽 · 50GB存储

[立即抢购](#)

**最新IT新闻:**

- 为什么是张勇 这个接班马云的人到底有多牛?
  - ofo融资迷局: 滴滴、阿里、戴威的三角博弈
  - 京东红太阳
  - 走近后厂村程序员的真实生活: 拿命换钱
  - 滴滴仍给无营运证的黑车司机派单 客服: 无证被扣车
- » 更多新闻...

**最新知识库文章:**

- 为什么说 Java 程序员必须掌握 Spring Boot ?
  - 在学习中, 有一个比掌握知识更重要的能力
  - 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
- » 更多知识库文章...

Copyright ©2018 woonu