

stream流附带集合

1月份

扩展 equals和deepequals

```
System.out.println(a==b);
System.out.println(a.equals(b));
```

打印两个值相等==后面的数值打印不出来 注意点

关于deepequals和lequals的总结 (equals的比的是==和equals) -> 对于字符串而言 比较的是地址和内容 对于对象而言比较的是地址 对于基本数据类型比较的是数值 (deepequals) -> 比equals多了对应内容的比较, 前提是数据类型一样

"字符串".charat(index)返回的是字符 不能和字符串比较 所以返回的是false

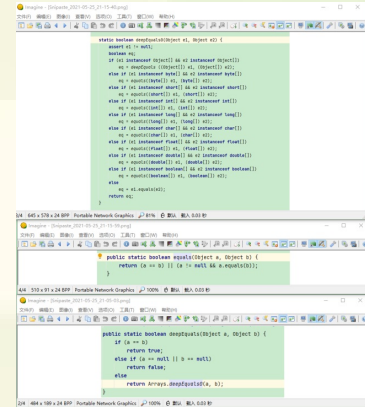
Collections.addAll(list, "打广告", "大苏打", "撒旦", "的", "撒"); 代替list多次添加 注意 map不是collection 所以不能用Collections.addAll, map只能乖乖用put

Map.ofEntries(Map.entry("张三",12),Map.entry("李四",15)) 创建不可变map 其他类似如List.of("撒",2)

数组则是使用Araays.stream(arr(具体数组名称))来获取stream流, 但是使用arr.stream() 按下alt+enter会让idea自动补全成Araays.stream(arr(具体数组名称))

只有list和set有stream流 所以map转换为stream前需要 map(实体类).entryset 获取set集合 或者 keyset以及values获取Collection集合

扩展: domain都是贫血模型 改为com.xxx.entity 引用http://irwinx.github.io/2017/03/04/%E7%B8%B6%E6%80%9D%E6%9E%81%E6%81%90-%E4%BD%A0%E7%9C%9F%E7%9A%84%E4%BC%9A%E5%86%99java%E5%90%97/



2月份

小补充: map是大括号 set和list是中括号 collection只包括list和set

带大括号 记得加; 尽量保证例如 (s) 有这个括号

重要: equals和==的区别 1 首先 查看地址值 System.identityHashCode 打印的是十进制的原始hashCode 并非是直接打印, 直接打印的是包名+类名+@分隔符+hashCode十六进制值, 然而hashCode值可以重写, 记住, 就算重写了hashCode内存地址还是不会变 所以==或者equals都会false 对象这两个比较的都是地址值, equals的源代码就是==比较. 若是重写equals则可以为true, 跟hashCode是否相等并无关联

skip(2)表示跳过前两个 limit(2)表示就要前两个 两个加起来组成所有的数

数组, list和set和map用.stream()方式获取流 其他的用stream.of()获取

这里的map函数用来映射 所以打印的就是aa bb cc

distinct() 去掉重复数据

续: 直接打印的地址值可以用toString重写值, 先打印的是包名+类名+@分隔符+十六进制hashCode, 若是不重写toString 可以打印重写过后的十进制转十六进制的hashCode 比如hashCode重写十进制200, 就是包+类+@+c8 (十六进制) 若是重写toString 就是toString的值了

sorted () 用来排序有默认构造还有比较器的参数, 所以return b-a就可以把大的放后, 小的放前

s2和s3值相等, stream.of后面可以接多个数值, 案例并非是数组或者集合

```
Sorted() 升序排序
Sorted() 降序排序
Sorted() 自定义比较器
Sorted() 自定义比较器
```

stream流的sorted和lmap等用法要记住后面进行操作类似于filter过滤进行排序以及映射等等这类操作, 需要在sorted () 这个括号内return b-a 返回倒序, 集合set也适用

4月份

字符串.split ("", "") 可以分开例如"zhangsan, 12"这两个数, 返回的是数组, 用【0】 以及 【1】 可以获得两边的数值

new arraylist (list.of这样的不可变集合) 可以将list.of的不可变性变为可变

a

collect属性

collect也可以对元素可以进行类似的遍历 主要作用是收集 顾名思义 兼顾每一个元素, 例如下图 主要是将集合进行转换为流进行输出, 因此每一个元素都会兼顾到

分支主题