

 Evgen959 / Advanced_Backend Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)[Advanced_Backend](#) / [Lesen 046](#) / [code](#) / [hw_na_les046_2](#) / [src](#) / [Main.java](#) 

...



Evgen959 newLes46

7cd990d · 6 minutes ago



65 lines (54 loc) · 2.58 KB

Code

Blame

Raw



```
1  /* 2. Дан текст (в рамках данной задачи текст не содержит знаков препинания,
2  слова разделены пробелом). Ваша задача написать метод, который сформирует частотный словарь,
3  т.е. получить map<String, Integer> где ключ - слово,
4  значение - сколько раз оно встречалось в тексте.*/
5
6  import java.util.Arrays;
7  import java.util.Collection;
8  import java.util.List;
9  import java.util.Map;
10 import java.util.function.Function;
11 import java.util.stream.Collectors;
12
13 public class Main {
14     public static void main(String[] args) {
15         String str = "anna jack lena anna john lena lena";
16         Map<String, Integer> map = countingWords(str);
17         System.out.println(map);
18         // {john=1, anna=2, lena=3, jack=1}
19
20         List<Person> people = List.of(
21             new Person("Jack", 10),
22             new Person("John", 20),
23             new Person("Jack", 10),
24             new Person("Jack", 14),
25             new Person("Jack", 19),
26             new Person("Jack", 19),
27             new Person("Lena", 19),
28             new Person("Lena", 25)
29         );
30         Map<Person, Integer> map1 = countingWords(people);
31         System.out.println(map1);
32         //{Person{name='Lena', age=25}=1, Person{name='Jack', age=14}=1,
33         // Person{name='Jack', age=19}=2, Person{name='Lena', age=19}=1,
34         // Person{name='John', age=20}=1, Person{name='Jack', age=10}=2}
35
36         Map<String, Integer> map2 = countingWords(people, p -> p.getName());
37         System.out.println(map2);
38         // {Lena=2, John=1, Jack=5}
39
40         Map<Integer, Integer> map3 = countingWords(people, p -> p.getAge());
41         System.out.println(map3);
42         // {19=3, 20=1, 25=1, 10=2, 14=1}
43
44     }
```

```
45
46     /*public static Map<String, Integer> countingWords(String str){
47         return Arrays.asList(str.split(" ")).stream()
48             .collect(Collectors.toMap(s -> s, s -> 1, (i1,i2)-> ++i1));
49     }*/
50
51     public static Map<String, Integer> countingWords(String str){
52         return countingWords(Arrays.asList(str.split(" ")));
53     }
54
55
56     public static <T> Map<T, Integer> countingWords(Collection<T> list){
57         return list.stream()
58             .collect(Collectors.toMap(s -> s, s -> 1, (i1,i2)-> ++i1));
59     }
60
61     public static <T, R> Map<R, Integer> countingWords(Collection<T> list, Function<T,R> keyFunction){
62         return list.stream()
63             .collect(Collectors.toMap(keyFunction, s -> 1, (i1,i2)-> ++i1));
64     }
65 }
```