

[Evgen959](#) / [Advanced_Backend](#) Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)[Advanced_Backend](#) / [Lesen 044](#) / [code](#) / [d06_12_1](#) / [src](#) / [Main2.java](#) ...

Evgen959 newLes44

76d693f · 10 hours ago



24 lines (18 loc) · 1.04 KB

[Code](#) [Blame](#)[Raw](#)

```
1  import java.util.*;
2  import java.util.stream.Collectors;
3
4  ✓ public class Main2 {
5  ✓      public static void main(String[] args) {
6
7          List<String> list = List.of("Jack", "John", "Mark", "Ann", "Lena");
8
9          List<String> collect = list.stream().toList();
10         Set<String> collect1 = list.stream().collect(Collectors.toSet());
11
12         LinkedList<String> collect2 = list.stream().collect(Collectors.toCollection(() -> new LinkedList<>()));
13         TreeSet<String> collect3 = list.stream().collect(Collectors.toCollection(() -> new TreeSet<>()));
14
15         //-----MAP
16         Map<String, Character> collect4 = list.stream().collect(Collectors.toMap(s -> s, s -> s.charAt(0)));
17         System.out.println(collect4);
18
19         Map<Character, String> collect5 = list.stream().collect(Collectors.toMap(s->s.charAt(0), s->s, (s1,s2)->s1+" "+s2));
20         System.out.println(collect5);
21         Map<Character, String> collect6 = list.stream().collect(Collectors.toMap(s -> s.charAt(0), s -> s, (s1,s2) ->s1));
22         System.out.println(collect6);
23     }
24 }
```

