

 Evgen959 / Advanced\_Backend Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)[Advanced\\_Backend](#) / [Lesen 032](#) / [code](#) / [hw\\_na\\_les031\\_1](#) / [src](#) / [Main.java](#) 

Evgen959 newLes32

74dd8bb · 10 hours ago



53 lines (47 loc) · 2.62 KB

Code

Blame

Raw



```
1  /*
2  2. Дан List<Employee>. У каждого Employee задано имя и год,
3  с которого сотрудник работает в фирме. Руководство решило премировать N сотрудников,
4  которые работают дольше всех.
5  Напишите метод, который вернет новый List<Employee> премируемых сотрудников.
6  Реализовать тесты.
7  Если нет идей, подсказка на следующей странице.
8  Хорошей идеей может быть отсортировать список работников по стажу
9  3. В предыдущей задачи учтите, что если например, нужно премировать 3 сотрудника,
10 но при этом в компании 5 старейших сотрудников с одинаковым стажем,
11 необходимо премировать всех пятерых.
12 Или например, есть 2 старейших сотрудника со стажем 10 лет,
13 и еще 5 со стажем 9 лет - премируем всех семерых. Реализовать тесты.
14 */
15
16 import java.util.ArrayList;
17 import java.util.Collections;
18 import java.util.Comparator;
19 import java.util.List;
20
21 public class Main {
22     public static void main(String[] args) {
23
24
25
26     }
27
28     public static List<Employee> selectEmployeeForBonus(List<Employee> list, int N){
29         List<Employee> result = new ArrayList<>();
30         List<Employee> sortedList = createSortedCopy(list,new ComparatorEmployeeByYear());
31         int i=0;
32         int year = 0;
33         for (i = 0; i < Math.min(N,sortedList.size()); i++) {
34             Employee emp = sortedList.get(i);
35             result.add(emp);
36             year = emp.getYear(); // год в последней ячейки нового листа
37         }
38         for ( ;i< sortedList.size();i++){
39             Employee employee = sortedList.get(i);
40             if (employee.getYear() == year){ //сравниваем год с последним добавленным в новый лист
41                 result.add(employee);
42             } else {
```

```
43         break;
44     }
45 }
46 return result;
47 }
48 ✓ public static List<Employee> createSortedCopy(List<Employee> list, Comparator<Employee> comparator) {
49     ArrayList<Employee> result = new ArrayList<>(list); // result - копия листа list
50     Collections.sort(result, comparator); // сортируем result
51     return result;
52 }
53 }
```