

# Java → Multifunctional mapper

Hard 25 minutes ?

40 users solved this problem. Latest completion was 21 days ago.

Code Challenge — Write a program

Write three functions:

1. **Multifunctional mapper (transformer)** that accepts a list of operators (mappers) and returns a new operator. The returned operator accepts a list of integer numbers and sequentially applies each mapper to each number in the list (performs multiple transformations). The result is a list with transformed values.
2. In terms of **the multifunctional mapper** define **an operator that multiplies by two each integer number and then add one to its**. The operator is applied to each number in the input list.
3. In terms of **the multifunctional mapper** define **an operator that squares each integer number and then calculates the next even number** following it. The operator is also applied to each number in the input list.

To simplify the problem all function (represented by objects) are declared, **you need to finish their realization**. Look carefully at definition of each function.

**Also there is an example:** identity operation that is defined in terms of **the multifunctional mapper**. It doesn't changes values in the input list. It repeats identity transformation three times just for example.

During testing all operators will be tested (including identity).

**Example 1.** An input list with integer numbers [1, 1, 1, 1].

- **identityTransformation** returns the result list [1, 1, 1, 1].
- **multTwoAndThenAddOneTransformation** returns the result list [3, 3, 3, 3].
- **squareAndThenGetNextEvenNumberTransformation** returns the result list [2, 2, 2, 2].

**Example 2.** An input list with integer numbers [1, 2, 3].

- **identityTransformation** returns the result list [1, 2, 3].
- **multTwoAndThenAddOneTransformation** returns the result list [3, 5, 7].
- **squareAndThenGetNextEvenNumberTransformation** returns the result list [2, 6, 10].

[Code Editor](#)

[IDE](#)

Java

```
1  /**
2   * The function accepts a list of mappers and returns an operator that accepts a list of integers
3   * and sequentially applies each mapper to each value (perform a transformation)
4   */
5  public static final Function<List<IntUnaryOperator>, UnaryOperator<List<Integer>>> multifunctionalMapper =
6      mappers ->
7          numbers -> numbers.stream()
8                          .map(mappers.stream().reduce(IntUnaryOperator.identity(), IntUnaryOperator:
9                          .collect(Collectors.toList()));
10 /**
11  * EXAMPLE: the operator transforms each number to the same number (perform the identity transformation)
12  * <p>
13  * It returns a list of the same numbers.
14  */
15  public static final UnaryOperator<List<Integer>> identityTransformation =
16      multifunctionalMapper.apply(Arrays.asList(x -> x, x -> x, x -> x));
17
18 /**
19  * The operator accepts an integer list.
20  * It multiplies by two each integer number and then add one to its.
21  * <p>
22  * The operator returns transformed integer list.
23  */
24  public static final UnaryOperator<List<Integer>> multTwoAndThenAddOneTransformation =
25      multifunctionalMapper.apply(Arrays.asList(x -> x * 2, x -> x + 1));
26
27 /**
28  * The operator accepts an integer list.
29  * It squares each integer number and then get the next even number following it.
30  * <p>
31  * The operator returns transformed integer list.
32  */
33  public static final UnaryOperator<List<Integer>> squareAndThenGetNextEvenNumberTransformation =
```

34

multifunctionalMapper.apply(Arrays.asList(x -> x \* x, x -> x % 2 == 0 ? x + 2 : x + 1));

35

✓ **Correct, but can be improved**

Thanks for your feedback!

Write here how we could improve this problem

Run

Continue

Solve again

Time limit: 8 seconds    Memory limit: 256 MB

Comments (3)

Hints (0)

Useful links (1)


Solutions (1)

Share something, Sergey Kubatko


Post

Please do not post solutions here

SK

**Sergey Kubatko** less than a minute ago 

<https://stackoverflow.com/questions/44520834/how-to-apply-a-list-of-functions-to-a-value-in-java-8>

 0 [Reply](#)