




You solved this problem of the day **about 1 hour ago**

Using streams many times

 Hard  6 minutes 

39 users solved this problem. Latest completion was **8 days ago**.

Code Challenge — Write a program

Sometimes after mapping and filtering a stream, you need to use it more than one time. For example, find the maximum and count all the elements of a resulted sequence.

You need to implement a method that solves this problem. The method should save the stream and create a supplier, that can return this stream over and over again.

For example, this code should output 4, 8, 2 on separate lines:

```
Stream<Integer> stream = Arrays.stream(new Integer[]{1, 2, 3, 4, 5, 6, 7, 8});
Supplier<Stream<Integer>> saved = saveStream(stream.filter(e -> e % 2 == 0));

System.out.println(saved.get().count());
System.out.println(saved.get().max(Integer::compareTo).get());
System.out.println(saved.get().min(Integer::compareTo).get());
```

Sample Input 1:

```
1 2 3 4 5 6 7 8
```

Sample Output 1:

```
4
8
2
```

[Code Editor](#)

[IDE](#)

Java

```
1 import java.util.function.*;
2 import java.util.stream.*;
3
4 class FunctionUtils {
5
6     private static List list;
7
8     public static <T> Supplier<Stream<T>> saveStream(Stream<T> stream) {
9         list = stream.collect(Collectors.toList());
10        return () -> list.stream();
11    }
12
13 }
14
```

✓ **Correct**

Thanks for your feedback!

Write here how we could improve this problem

[Continue](#)

[Solve again](#)

Reference solution [Ⓐ]

These solutions are generated semi-automatically and may sometimes look too complicated or even bizarre. Please use it as a source of inspiration, not as a best possible solution for this problem. We are still improving the generation algorithm.

```
import java.util.*;

import java.util.function.*;
import java.util.stream.*;

class FunctionUtils {

    public static <T> Supplier<Stream<T>> saveStream(Stream<T> stream) {
        ArrayList<T> list = new ArrayList<>();
        stream.forEach(list::add);
        return list::stream;
    }

}
```

Time limit: 8 seconds Memory limit: 256 MB

[Comments \(3\)](#)

[Hints \(0\)](#)

[Useful links \(0\)](#)

[Solutions \(0\)](#)

Share something, Sergey Kubatko

Post

Please do not post solutions here

Sort by: **Last posted** ▼

KP **Konstantin Proskurnya** 4 months ago [Report](#)

This tutorial helped me a lot - <https://winterbe.com/posts/2014/07/31/java8-stream-tutorial-examples/>

♡ 1 [Reply](#)

SP **Shivam Pundir** 2 months ago [Report](#)

thanks that really helped

♡ 0 [Reply](#)

A **Andrew** 7 months ago [Fixed](#)

st.filter() should be stream.filter()

♡ 0