Java → Functional streams → Stream pipelines

Java → The ten most frequent words

Hard 🕒 8 minutes ?

56 users solved this problem. Latest completion was **6 days ago**.

Code Challenge — Write a program

Write a program that reads a text (in the UTF-8) from the standard input. The program must count the frequency of words in the text and print the 10 most frequent words.

A word is a sequence of characters consisting only of digits and letters. For example, the string "Functions bring happiness!" has three words: "Functions", "brings", "happiness".

The counting words should be case-insensitive, i.e. "Functions", "functions" and "FUNCTIONS" are the same word. Output words in the lower case.

If the text has less than 10 unique words, output as many as there are.

If some words in the text have the same frequency, order them lexicographically as well. For details, see here (https://en.wikipedia.org/wiki/Lexicographical_order).

The problem has a beautiful solution using streams without any loops and conditional operators. Try to write it.

Sample Input 1:

```
Functions bring happiness!
```

Sample Output 1:

```
bring
functions
happiness
```

Sample Input 2:

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sodales consectetur purus at faucibus. Donec
mi quam, tempor vel ipsum non, faucibus suscipit massa. Morbi lacinia velit blandit tincidunt efficitur.
Vestibulum eget metus imperdiet sapien laoreet faucibus. Nunc eget vehicula mauris, ac auctor lorem. Lorem
ipsum dolor sit amet, consectetur adipiscing elit. Integer vel odio nec mi tempor dignissim.
```

Sample Output 2:

```
consectetur
faucibus
ipsum
lorem
adipiscing
amet
dolor
eget
elit
mi
```

Code Editor IDE

Solve in IDE

✓ IDE is opened

If you don't see your IDE opened, switch to it manually

✓ **Correct**

Thanks for your feedback!

Write here how we could improve this problem

Continue

Reference solution β

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.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.Arrays;
import java.util.stream.Collectors;

public class Main {

    public static void main(String[] args) {
        new BufferedReader(new InputStreamReader(System.in))
                .lines()
                .flatMap(line -> Arrays.stream(line.split("[\\p{Punct}\\s]+")))
                .collect(Collectors.toMap(w -> w.toLowerCase(), w -> 1, Integer::sum))
                .entrySet()
                .stream()
                .sorted((x, y) -> y.getValue() == x.getValue()
                        ? x.getKey().compareTo(y.getKey())
                        : (y.getValue() - x.getValue()))
                .limit(10)
                .forEach(x -> System.out.println(x.getKey()));
    }
}
```

Time limit: 8 seconds Memory limit: 256 MB

Comments (8) Hints (2) Useful links (0) Solutions (1)

Share something, Sergey Kubatko

Post Please do not post solutions here

Sort by: Last posted ▾

OR **Oleksandr Rodiuk** 21 days ago Report

wild task..

♡ 0 Reply

LA **LAURENT APICELLA** about 1 month ago Report

Not convinced by this paradigm

♡ 0 Reply

U1 **User 1074039** 3 months ago Report

Cool task!

♡ 0 Reply

KP **Konstantin Proskurnya** 4 months ago Fixed

Typo - A word is a sequence of characters consisting only of digits of letters

♡ 0 Show all

DG **Dmitrii Gushcha** 4 months ago Report

That was nice

♡ 0 Reply

I **Icskatinggn** 5 months ago Report

Holy shit I did it. I've really enjoyed these sections on Functional Programming in Java. They've really made me think!

♡ 0 Reply

MS **Mateusz Szafarz** 7 months ago Report

@Florin Barbuceanu I'm guessing that the sequence is not right. The description clearly says: "If some words in the text have the same frequency, order them lexicographically as well.".

♡ 0 Reply

FB **Florin Barbuceanu** 8 months ago Report

I think the test for "Functions bring happiness!" is broken.
I suggest it tests that all the expected tokens are present, not that the order is the same.
This is what I see currently:

Failed test #1. Wrong answer

♡ 0 Show all Reply