Java/Kotlin Developer Homework

OVERVIEW

You will create a program, using Java or Kotlin, that provides summary statistics about the given input.

INPUT

Your program will be provided with a List of Strings.
Each element in the input will be a string that is one of the following:

- A number, like “50” or “-7.4”, to be treated as a number
- An instruction to repeat another element, in the format “Repeat element 12, 4 times”
    - This is treated as though that other element appeared this many times, instead of this repeat instruction element.
    - Element reference numbers start at 0. This instruction does not, itself, change any element’s reference number.
- Any other string, to be treated as a plain string

A sample input might be a list of:

- 4
- squid
- Repeat element 1, 7 times
- -4
- 3
- Repeat element 4, 3 times
- elephant

OUTPUT

Your program must create a result object from these inputs. The object must have the following methods:

- sum() - the total of all numbers in the input.
    - For the sample input, this would be 12.
- average() - the arithmetic mean of all numbers in the input.
    - For the sample input, this would be 2.

- contains(plainString) - whether the input contained the argument as a plain string, as an entire single element.
    - For the sample input, this would be true for “squid” or “elephant” and false for all other inputs.
- fractionNumeric() - the fraction of all elements in the input that are numbers.
    - For the sample input, this would be 0.4.
- String plainStringCounts() - an inventory of the plain strings present in the file, along with the number of times that plain string appeared. Must be in reverse alphabetical order.
    - For the sample input, this would be “squid:8; elephant:1”.

STARTER PROJECT

If you like, you may use your choice of the two starter Gradle projects (one Java, one Kotlin) that have been attached as zip files. They provide only a simple Gradle project you can build and run and a sample main method.
This is entirely optional - please feel free to create your own project instead.

ADDITIONAL INSTRUCTIONS

- Implement at least a few executable tests.
- Please do not use any dependencies other than your language’s standard library, or a test framework such as JUnit.
- These instructions may include some ambiguities. If you encounter an ambiguity, make whatever necessary assumptions you think are best, and document them.
- To submit your work, please send your project back to your contact as a ZIP archive. Please include only your source code, and not any .jar, .class, .bat or other executable files, which may be blocked by our email system.
    - If you are using the starter project, running the “submit” task will zip up your project for you, excluding those files, as “(your username) homework.zip”.