Digital14 Coding Task

Please read the below given problem statement and design a working solution for it. Please make sure to take care of below pointers while designing and coding your solution:

* Provide a working solution.
* Design should be extensible to take care of future needs.
* Unit test cases for the working solution.
* Use of Design principles and patterns to design your solution

**Problem Statement:**

There are two kinds of Writers: StringWriter which writes to a String and FileWriter that writes to a file (these are not related to those classes provided in Java). There can be other types of Writers (like SocketWriter that writes to a Socket) in the future.

A Writer simply writes to a target (like string or file) until close is called. Any effort to write after the call to close is simply ignored.

You can find out what was written to a StringWriter by calling a function. You can find out what was written to a file using a FileWriter by reading the file written to.

There are several types of operations that can be performed on these two (or other types added in the future) writers. For example:

**lower case**: This converts the string being written to all lower case

**upper case**: This converts the string being written to all upper case

**stupid remover**: This replaces the word stupid (only in lower case) to s\*\*\*\*\*

**duplicate remover**: This removes consecutive duplicated words. For example,

"This is is it" will replaced by "This is it" when this function is applied.

Design so that other such functions may be added in the future, but without changing any existing class.

The user of your design will pick and choose what kinds of operations they want to use or combine to use.

User may be interested in combining the stupid remover and the lower case operation when writing to a String. I may also be interested in combining the duplicate remover and the upper case operation when writing to a file. I may chose to use any combination of these operations at will.

**For example:**

You can imagine an operation like this (pseudocode):

writeToWriter(Writer writer) {

writer.write("This is really really stupid!!!")

writer.close();

}

If I had combined the operations of duplicate remover, stupid remover,

and if the writer is targeting a file (say myfile.dat), then the file

myfile.dat would contain the content

This is really s\*\*\*\*\*!!!