Today, you will construct anti-plagiarism program. The aim is to detect plagiarism among a group of students working on a programming assigment. You need to check whether any scripts in the working directory, containing several scripts, are too similar to each other. Obviously they do not have to be identical, only a part could have been copied, or some empty lines could have been inserted.

First of all, you need to be able to quantify the similarity of two strings. The most simple measure is the Hamming Distance between two strings. It is the number of positions at which these strings vary. In more technical terms, it is a measure of the minimum number of changes required to turn one string into another. For example, one string is **medication**, the other one is **meditation**, in this case the Hamming distance is 1. It is 0 when the strings are identical. The strings can be of different length, for example **speed**, the other one is **speeding**, in this case the Hamming distance is 3.

To compare the strings, provide a class `Hamming` with the public method `compare` which compares two strings and returns the Hamming distance. It should throw an exception, when one of the strings is empty. Check the method with few different examples.

The strings could differ only by whitespaces, tabulators, etc. To handle this situation provide a class `HammingCleared` that is a child of the `Hamming` class. `HammingCleared` class should contain the private method `clear` which removes from the string all charactes such as whitespaces, tabulators, underlines. `HammingCleared` class should also contain the `compare` method, that overrides the method from the parent class. This method should clear both strings first, and then compare them using the method from the parent class. Use the keyword super within the overridden method in the child class to use the parent class' method.

...possible to be continued in the next week if there's no enough time today...

Once you can quantify the string similarity, construct a class `CheckPlagiarism`. The class should have a public method `CompareFiles` that loads two files and compare their lines using the Hamming Distance. Obviously the similar lines can be located at places in the files. In order to detect them, start with the file having more lines, and compare any line from the first file, to all the lines of the second file, recording the minimal Hamming Distance. Then, calculate the average of the minimal Hamming Distances you calculated. If this average value is below some threshold (represented by some static variable) you may classify both files as "Plagiarism detected". Report what is the average value of the minimal Hamming Distance, and how many lines are identical in both files. If this average value is zero, than both files are identical. You need to properly hadle the exceptions in case you have empty lines.

Finally, chceck for plagiarism for the all the files in some working directory. You may use your fellow students code to check the similarities, and to set the proper threshold. Also you may copy some of the files introducing few changes to see if the plagiarism is detected. Use `listFiles()` method on the `java.io.File` object to list the files in the working directory.