`DocumentStatistics` Design Document

Description

The `DocumentStatistics` object will read documents, calculate specific ratios pertaining to the document, and attempt to guess the author of the document by calculating the "distance" between each author's stats and the document stats. The calculated ratios are the average word length, the type-token ratio, the Hapax Legomena ratio, the average number of words per sentence, and the average number of phrases per sentence.

Services

The constructor `DocumentStatistics()` can be used to construct a new `DocumentStatistics` object. This is where all the calculation methods are run and the main work is done for guessing the authors of the documents. The method prints out the author names as it identifies them. This method runs in `O(A * D + N * D)` time, where `A` is the number of authors whose statistics have to be checked, `D` is the number of documents that are being analyzed and identified, and `N` is the (average) length of the documents being parsed.

The method `analyzeAllDocs()` can be used to analyze the default documents indexed as `{1,2,3,4,5}`. This method is also overloaded to take in a 1D array of `int`s, which it uses as documents indices to analyze. The method calculates the 5 important ratios for the document and guess the author of the document. This method runs in `O(A * D + N * D)` time, where `A` is the number of authors whose statistics have to be checked, `D` is the number of documents that are being analyzed and identified, and `N` is the (average) length of the documents that are being parsed.

Internal Data Structures and State

The internal data structures used by `DocumentStatistics` objects are two 2D arrays of `double`s, a 1D array of `double`s, three 1D arrays of `String`s, a 1D array of `int`s, and a list of `Sentence`s that holds the parsed file. The two 2D arrays of `double`s are used to hold the calculated ratio metrics for the parsed files and the authors. The 1D array of `double`s is used to hold the weights used to identify the author for each document. The three 1D arrays of `String`s are used to hold the author names with normal capitalization and spacing, the author names in their directory forms (`firstName.LastName`), and the final guesses for the authors of each document. The 1D array of `int`s is used to hold the indices of the documents that were analyzed. Lastly, the list of `Sentence`s is used to hold the ordered set `Sentence`s that represents the parsed document.

The `DocumentStatistics` class also utilizes a `Document` private instance variable and a `Scanner` private instance variable. The `Document` is used to parse the document and transform it into a list of `Sentence`s. The `Scanner` is used by the `Document` in order to parse the document.

Test Plan

Both provided methods (`DocumentStatistics()` and `analyzeAllDocs()`) can be tested by running them on various documents and verifying that the methods always return the correct author for each one.