TextAnalyzer

External Documentation

This program is based on 10 modules:

- 8 classes:

    - Main: This class launches the application by creating a Window object.

    - Window: This class sets the GUI and other visual aspects of the program. It creates 10 functional buttons, 3 text fields, 2 labels, and a scroll pane. All this held by a frame. This class also works with all the functionalities dealing with choosing-opening-portraying a file, converting the File into text (String), extracting the text from the textbox for later analysis, performing word search from the text box based on user input. This module also holds the Wordstractor and passes it all the text from the chosen File or the text extraction.

        - Wordstractor: This module takes text and extracts all the individual words from it, which then are set into unordered lists that are to be sorted afterwards. In the extraction process the text is first sanitized from any punctuation sign. The resulting text is then given to a series of specialized list. The following lists compose this series.

        - List: This module represents a linked-list of nodes that hold individual words. This module is able to output an unordered String representation of all its nodes and the node's content.

            - MergeSortList: This module is an extension of List and its main functionality is to sort the collection of nodes held by List. This collection is sorted based on alphabetical order through comparison-based algorithm merge sort. This module gives the functionality to sort things in reverse alphabetical order.

            - OccurrenceList: This module is an extension of List and its main functionality is to sort the collection of node held by List. The collection is sorted based on the times the content of a node is repeated. This module takes the original List in order to make a customized copy (where certain nodes are eliminated and some are edited) that is suitable to the algorithm applied.

- LengthList: This module is an extension of List and its main functionality is to sort the collection of node held by List. The collection is sorted based on the length of the content held by a node.

- Node: This module represents the container where objects added to the list will be held. This node is the main unit of the list and has the capability to point to its neighboring nodes (linking).

- 1 Interface

- Comparable: This interface imposes a total ordering on the objects of each class that implements it. This ordering is referred to as the class's natural ordering, and the class's compareTo method is referred to as its natural comparison method. This interface is mainly used to compare the content of the nodes in the list for sorting purposes.

1 Exception class

- ListException: This exception interface is used to alert and catch any abnormalities that might have resulted from the List class. It indicates conditions or erroes that a reasonable application might want to catch.