***Project #2 Word statistics***

**Project Description:**

The purpose of this project is to develop a program with a graphical user interface (GUI) that reads all text files from a specified directory. The program should provide various word statistics, including the number of words per file and directory, occurrences of specific words ("is," "are," "you"), longest and shortest words per file and directory. The program should have the capability to search for text files in subdirectories as well.

**What we Covered:**

Developed a program with a GUI to read text files from a specified directory.

Implemented functionality to search for text files in subdirectories.

Calculated and displayed word statistics in real-time, including the number of words per file and directory, occurrences of specific words, and longest and shortest words per file and directory.

We used Multithread to perform the operations on each file.

We use synchronization mechanisms to avoid race conditions and ensure the consistency of the data.

DOCS

This documentation provides an overview of the Java application for word statistics. The application is designed to analyze text files within a specified directory, counting words and providing statistics such as the longest and shortest words.

Classes

1. **FrameGUI.java**

* Purpose: Creates the main GUI frame for the application.
* Components:
* `JFrame Frame`: The main frame of the application.
* `ImageIcon icon`: Icon for the frame.
* `BottomPanel bottom`: Instance of the BottomPanel class.

2. **TopPanel.java**

* **Purpose**: Provides a top panel in the GUI for user input.
* **Components**:
* `JTextField directoryName`: Displays the selected directory name.
* `JTextField directoryPath`: Displays the selected directory path.
* `JCheckBox selectDirectoryCheckBox`: Checkbox to include subdirectories.
* `File Directory`: Selected directory.
* `boolean Subdirectories`: Indicates whether to include subdirectories.
* `JButton browseButton`: Button to browse and select a directory.

**3. BottomPanel.java**

* **Purpose**: Contains the bottom panel with buttons, a table, and statistical labels.
* **Components**:
* `JButton start`: Button to initiate file processing.
* `DefaultTableModel tableModel`: Table model for displaying file statistics.
* `JTable resultTable`: Table displaying file statistics.
* `JScrollPane scrollPane`: Scroll pane for the table.
* `JPanel BottomPanel`: Main panel.
* `JPanel row1, row2, row3`: Sub-panels for organizing components.
* `JLabel Longest, Shortest`: Labels for displaying longest and shortest words.
* `File[] files`: Array of files selected for processing.
* `Semaphore semaphore`: Semaphore for thread synchronization.
* `int longestLength, shortestLength`: Static variables for tracking longest and shortest word lengths.

4. **Files.java**

* **Purpose**: Handles file-related operations.
* **Methods**:
* `ArrayList<File> txtFiles`: ArrayList to store text files.
* `getFiles(File directory, boolean includeSubdirectories)`: Retrieves text files from a directory.

5. **FileThread.java**

* **Purpose**: Implements a thread for processing individual files.
* **Components**:
* `Path filePath`: Path of the file to be processed.
* `DefaultTableModel tableModel`: Table model for updating statistics.
* `int rowIndex`: Index of the current row in the table.
* `int num\_of\_words, num\_of\_is, num\_of\_you, num\_of\_are`: Counters for different word types.
* `int longestWordLength, shortestWordLength`: Variables for tracking word lengths.
* `JLabel Longest, Shortest`: Labels for displaying longest and shortest words.
* `Semaphore semaphore`: Semaphore for thread synchronization.
* `void processFile(Path filePath)`: Processes the content of a file.

6. **WordStatistics.java**

* **Purpose:** Contains the main method to launch the application.
* **Components**:
* `FrameGUI Frame`: Instance of the main frame.

How to Use the Application

1. Launch the application by running the `WordStatistics.java` file.
2. In the top panel, click the "Browse" button to select a directory.
3. Optionally, check the "Include subdirectory" checkbox to include files from subdirectories.
4. Click the "Start Processing" button to initiate file processing.
5. View the processed file statistics in the table, including word counts and lengths.
6. The longest and shortest words are displayed in the bottom panel.

Notes

* The application supports processing multiple files concurrently using threads.
* Word statistics are displayed in real-time as files are processed.
* The longest and shortest words are dynamically updated during processing.
* File selection and processing status are displayed in the GUI.
* The application provides a responsive and user-friendly interface for analyzing word statistics in text files.