

**Word Statistics**

Project Description:

The "Word Statistics" program processes text files within a chosen directory, providing word-related statistics per file and for the entire directory. It includes functionalities like:

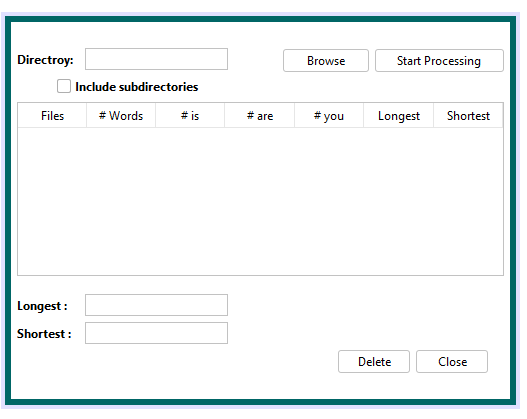
* **File Analysis:** Reads text files from a specific directory, optionally including subdirectories.
* **Word Metrics:** Calculates statistics like the number of words per file and directory, longest and shortest words, and counts occurrences of specific words ("is," "are," "you").
* **Real-time Updates:** Displays file names being processed and continually updates statistics during runtime analysis.

This program offers comprehensive insights into word-related data within text files, updating information dynamically as it processes files, and providing an ongoing summary of the analyzed data.

What We Have Done:

1. GUI (Graphical User Interface), done by: Mariam Hossam.
2. Functions Count Is, Are and You in file, done by: Andrew Gameel Bishry.
3. Functions Longest and Shortest word in file, done by: Rehab Sayed Ahmed.
4. Threads Code, done by: Mohamed Medhat.
5. Functions Longest and Shortest word in directories and All methods in conn class without show data method , done by: Jana Emad EL Deen.
6. Function Count in all files, Show data method and Video viewing the project, done by: Mariam Saad Mohamed.
7. Documentation (which involved the idea/description of the project),

It was done by: Hana Salama Osman.



The GUI features an interface with input and output sections:

**Input Section:**

* + **Directory Selection:** Allows users to input a directory path or choose one via a browse button.
  + **Checkbox Option:** Provides the choice to include subdirectories for analysis.

**Output Section:**

* + **Tabular Display:** Presents word-related statistics in table form, comprising:
  + Total word count (#words)
  + Occurrences of "is," "are," and "you" within the files analyzed.

The GUI interface simplifies directory selection and analysis options while presenting essential word statistics in a clear, tabulated format for user review and analysis.

It is done by: Mariem Hossam.

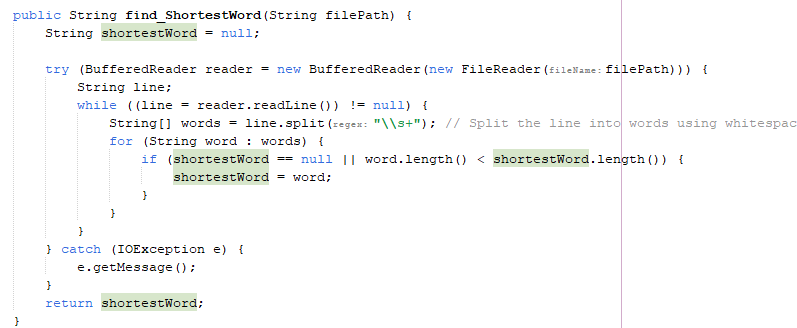
1. Code Implementation,

It was done by: Jana Emad EL Deen Ahmed, Mohamed Medhat, Rehab Sayed Ahmed, Mariam Saad Mohamed, Andrew Gameel Bishry.

Code Documentation:

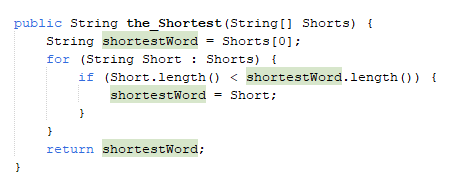
**Shortest word/file:**

find\_ShortestWord, reads a file specified by the given file Path and finds the shortest word in the file. Here's a breakdown of the function:



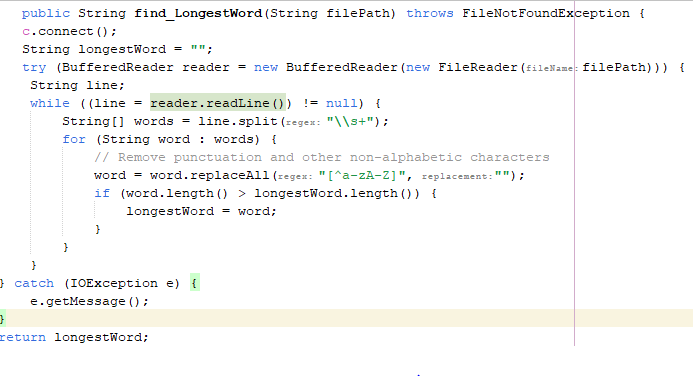
**Shortest word/directories:**

the\_Shortest, reads a directory specified by the given directory path and finds the shortest word in the directory. Here's a breakdown of the function:



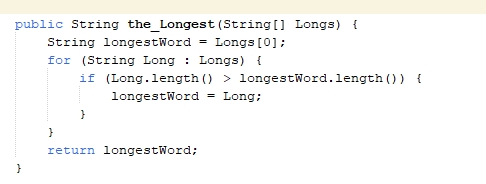
**Longest word/file:**

find\_LongestWord, reads a file specified by the given file Path and finds the longest word in the file. Here's a breakdown of the function:



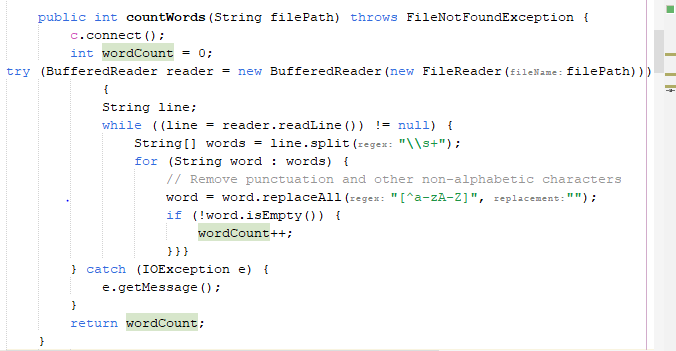
**Longest word/directories:**

the\_Longest, reads a directory specified by the given directory path and finds the longest word in the directory. Here's a breakdown of the function:



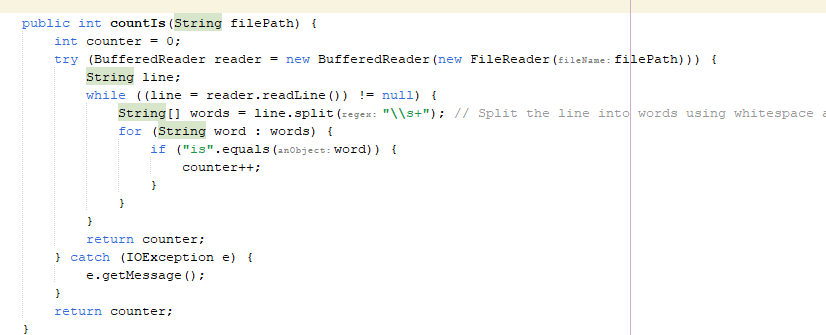
**Words’ Count:**

countWords, reads a file specified by the given file path and counts the total number of words in the file. Below is a breakdown of the function:



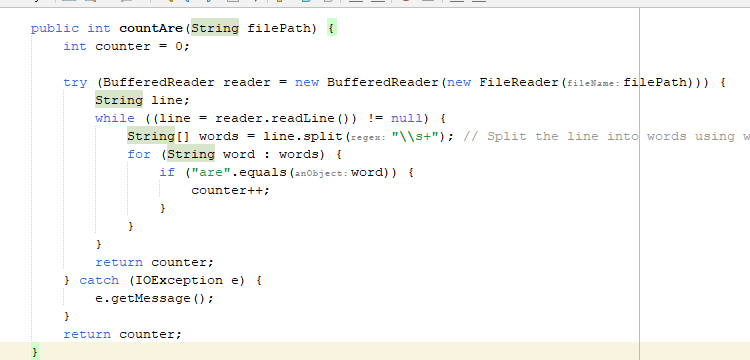
**“Is” Count:**

countIs, reads a file specified by the given filePath and counts the "is” words in the file. Below is a breakdown of the function:



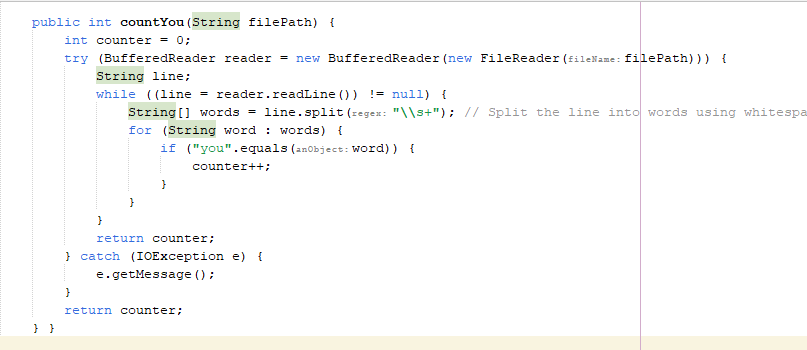
**“Are” Count:**

countAre, reads a file specified by the given file path and counts the "are” words in the file. Below is a breakdown of the function:



**“You” Count:**

countYou, reads a file specified by the given file path and counts the "you” words in the file. Below is a breakdown of the function:



**Word Thread:**

**Word**, extends the **Thread** class and seems to perform various word-related operations on a file.

Here's a summary of what each method does:

**Constructor:**

Initializes the class with a file path (filePath).

**run method:**

Connects to a database using c.connect().

Invokes several methods to perform various word-related operations, including counting words, finding the longest and shortest words, and counting occurrences of specific words.

Catches FileNotFoundException and prints an error message if the specified file is not found.

**countWords method:**

Connects to a database using c.connect().

Reads the content of the file specified by filePath.

Counts the total number of words in the file (words are defined as sequences of alphabetic characters).

**find\_LongestWord method:**

Connects to a database using c.connect().

Reads the content of the file specified by filePath.

Finds and returns the longest word in the file (ignores non-alphabetic characters).

**find\_ShortestWord method:**

Reads the content of the file specified by filePath.

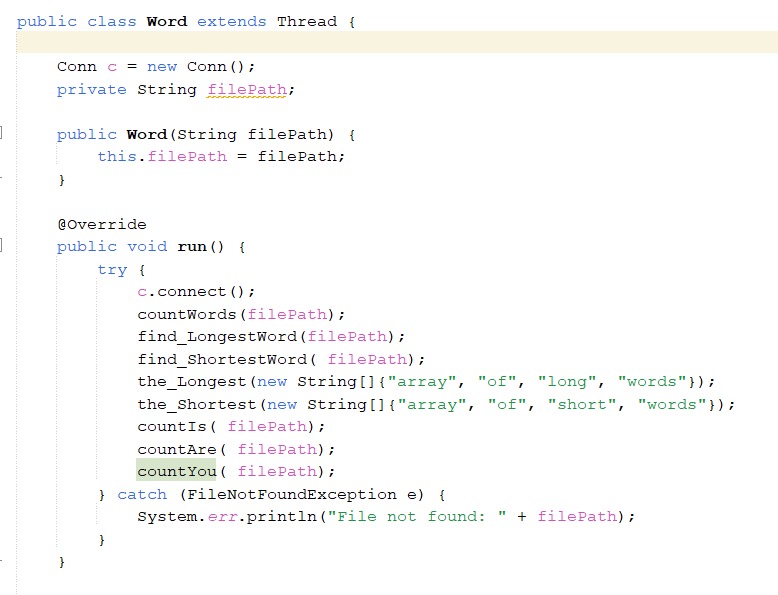
Finds and returns the shortest word in the file (ignores non-alphabetic characters).

the\_Longest and the\_Shortest methods:

Find and return the longest and shortest words, respectively, from an array of strings.

**countIs, countAre, and countYou methods:**

Count the occurrences of the words "is," "are," and "you," respectively, in the file specified by filePath.



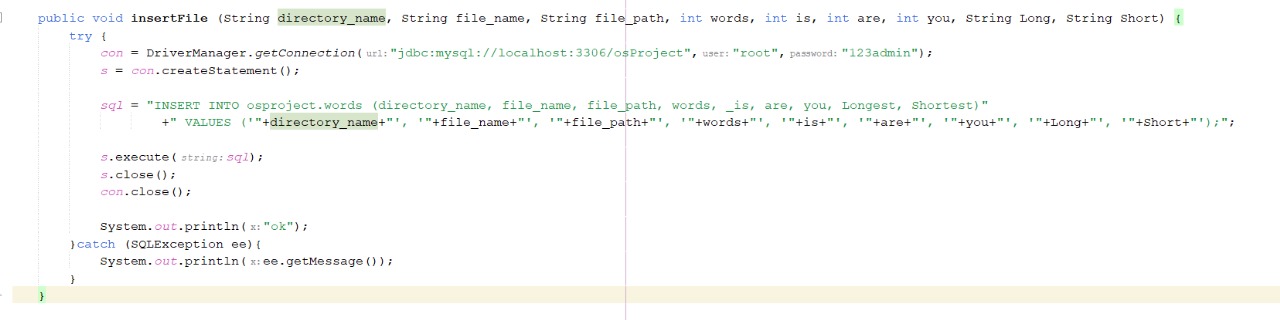
**“Conn” Function:**

Conn, appears to handle database operations related to an "osProject" involving word analysis.



**“InsertFile” Function:**

To insert data in to database.



**“ShowShortest” Function:**

The ShowShortest function in the provided Conn class retrieves data from the words table in a MySQL database, specifically extracting values from the "Shortest" column.



**“ShowLongest” Function:**

The ShowLongest function in the provided Conn class retrieves data from the words table in a MySQL database, specifically extracting values from the "Longest" column.



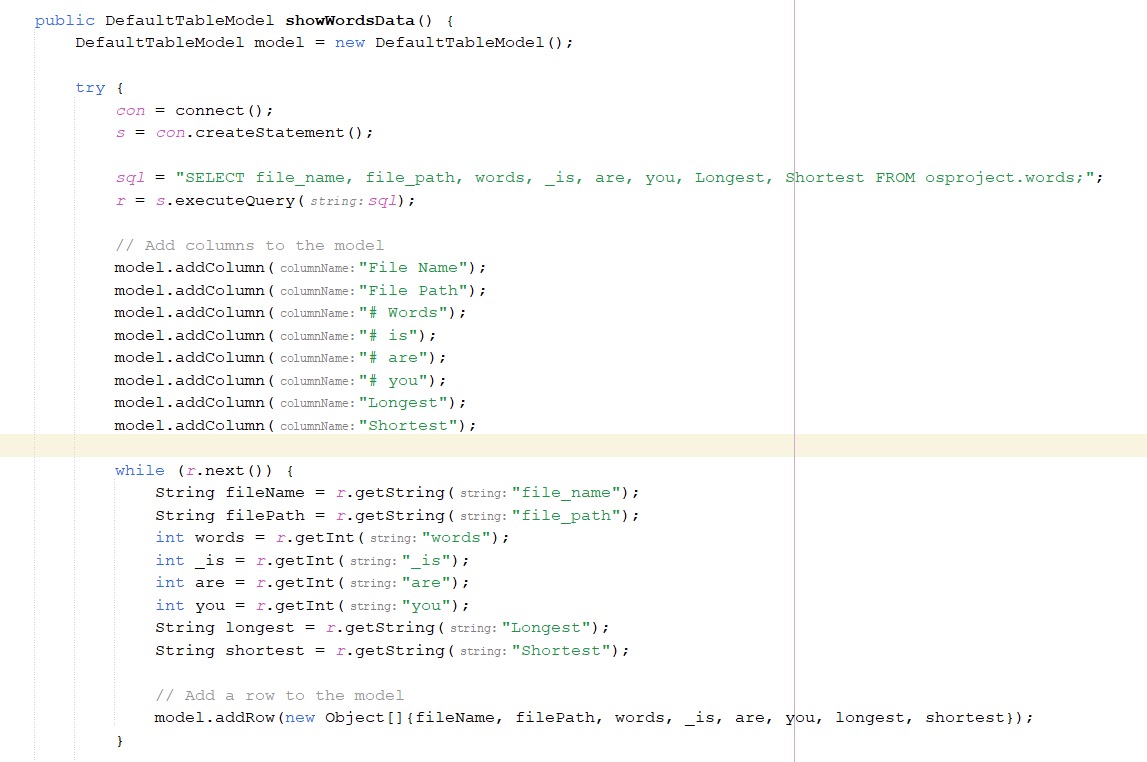
**“deleteRecordFromDatabase” Function:**

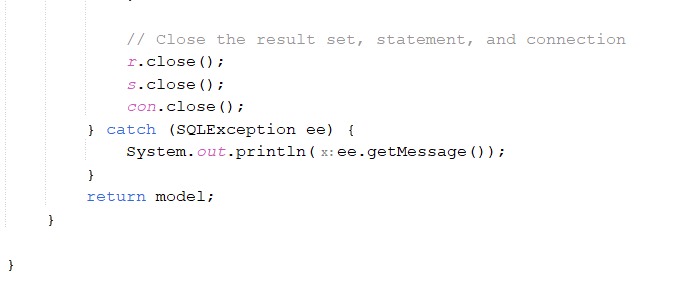
The deleteRecordFromDatabase function in the provided Conn class deletes records from the words table in a MySQL database named "osProject" based on a specified directory name.



**“DefaultTableModel showWordsData()” Function:**

The showWordsData function in the provided Conn class retrieves data from the words table in a MySQL database named "osProject" and returns it as a DefaultTableModel.





Thank You