

**LAB PROJECT REPORT**  
**ALI MOHAMMED ABDULRAHIM**  
**202039600**  
**SEC 54**

**Making a dictionary program that reads a text file in java that is used as spell checker using AVL tree, Binary search tree and binary tree due to the low time complexity of “log n” comparing with linked list “n” for instance.**

**Dictionary class has:**

- 1) public Dictionary(File f) : Constructor that initializes the dictionary with words read from a file f using scanner by having a while loop that goes to each line and add to the AVL tree or rise an error if the file not found.**
- 2) public void addWord(String s) throws WordAlreadyExistsException : Adds a new word to the dictionary using BST insertion method ‘insert’ and throws an exception if the word already exists in the dictionary.**
- 3) public boolean findWord(String s) : Searches in the dictionary for a word using BST search method that returns true if it exists and false otherwise.**
- 4) public void deleteWord(String s) throws WordNotFoundException : Deletes a word from the dictionary using BST ‘deleteByCopying(s)’ method and throws an exception if the word does not exist in the dictionary.**
- 5) public String[] findSimilar(String s) : Returns an array of words that are similar to s by :
  - a) looping each char with each alphabet letter replace char with alphabet letters except the same char and if the resulting word is in the dictionary add to the ‘simwrds’ list.**
  - b) removes a char from s at index i in the same loop and checks if the resulting in the dictionary and if so add it.**
  - c) swap the adjacent and see if it is in the dictionary and add it, if it is.****

6) public void saveToFile(File f) : Saves the words in the dictionary to a file f using 'BufferedWriter' by:

start from the root node of the tree and if the current node has no left child write the current node's data to the file, then move to the right child.

However, if the current node has a left child traverse its left subtree and find the right node in the left of the current node.

Then, if the right child of the right node is null, set it to the current node then go to the left.

However, if the right child of the right node is not null, so it already traversed the left, and move to the right child so write the current node's data to the file then go to the right child.

WordAlreadyExistsException : a class made to make an exception if word already exist in the tree when adding.

WordNotFoundException : a class made to make an exception if word does not exist in the tree when removing.

Dictionary Driver class which test all the method from Dictionary class.

```
\ICS202_Project\bin> Dictionary_Driver
Enter filename> mydictionary.txt
Dictionary loaded successfully.

choose operation no. ( 1.add, 2.remove, 3.find, 4.similar, 5.save )> 3
check word> punter
Word not found.

choose operation no. ( 1.add, 2.remove, 3.find, 4.similar, 5.save )> 3
check word> robe
Word found.

choose operation no. ( 1.add, 2.remove, 3.find, 4.similar, 5.save )> 1
add new word> punter
Word added successfully.

choose operation no. ( 1.add, 2.remove, 3.find, 4.similar, 5.save )> 1
add new word> robe
Exception: Word already found.

choose operation no. ( 1.add, 2.remove, 3.find, 4.similar, 5.save )> 2
remove word> punter
Exception: Word not found.

choose operation no. ( 1.add, 2.remove, 3.find, 4.similar, 5.save )> 2
remove word> zoom
Word removed successfully.

choose operation no. ( 1.add, 2.remove, 3.find, 4.similar, 5.save )> 4
search for similar/s to> punter
punter.

choose operation no. ( 1.add, 2.remove, 3.find, 4.similar, 5.save )> 4
search for similar/s to> ali
eli, ami, ani, ala, alb, ale, all, alp, bali, mali, alai, alia, al, ail.

choose operation no. ( 1.add, 2.remove, 3.find, 4.similar, 5.save )> 4
search for similar/s to> blog
clog, flog, slog, blob, bloc, blot, blow, log, bog.

choose operation no. ( 1.add, 2.remove, 3.find, 4.similar, 5.save )> 5
Save Updated Dictionary (Y/N)> y
Enter filename> mydictionary2.txt
Dictionary saved successfully.
Exiting program.
PS D:\KFUPM\222\ICS 202\ICS202_Project> █
```