

Overview

The program uses the data structure such as list and array to solve the crossword puzzle.

A summary of the formatting requirements are in the csci 3901 course assignment #4 information in the course's brightspace space.

Files and external data

There are two main files:

- FillInPuzzleUI.java -- main for the program that prompts the user for the operation to be performed.
- FillInPuzzle.java -- it acts as an interface for the function load puzzle, solve, print and choices.

Data structures and their relations to each other

This program uses lists and array to solve the puzzle
puzzlematrix -- it is used to store the characters of the words

integer_array -it is used to stores 1's in the indexes where a letters can be put and 0's where letters cannot be put.

row --it is used to store the total no of rows the puzzle box can have.

column -- it is used to store the total no of the columns the puzzle box can have.

n -- it is used to store the the word length

Assumptions

A summary of the assumptions to be made are in the csci 3901 course assignment #4 information in the course's brightspace space.

Choices

The user can invoke either loadPuzzle or solve or choices or print function.

Design elements

The main aim of the program is to solve the crossword puzzle

loadPuzzle() - it is used to create puzzle box and add letters to it

solve() - it is used to solve the crossword puzzle .

print() -- it is used to print the solution to the output stream file.

Choices() - it gives the no of ways the puzzle can be solved.

Strategy and Algorithm

Initially the crossword puzzle box is created.

The words array contains the words to be put in the puzzle box and matrix is the layout of the puzzle box. The initial index is zero. The comparison between the row and column is made. If the row value is greater, then maxsize value is set to row value otherwise it is set to column value.

The words array, matrix, initial index and maxsize is sent as parameters to solve function.

In solve function two iteration loops are used. One loop is to check the words that can align vertically. The other loop is to check the words that can align horizontally.

When running these loops, the words from the words array are checked and if it has to be aligned vertically or horizontally is checked. If the word has to be aligned vertically then the index value given is used to fit in the word in the puzzle box. If it can be fit then it is placed in the particular index value. If not then backtracking is used to find the other words that can be put into a particular index.

The same steps are repeated until all the words are filled in the crossword.

Limitations

The size of the grid is limited to the maximum size of the integer datatype.

References

[1] <https://www.geeksforgeeks.org/solve-crossword-puzzle/>

[2] <https://www.geeksforgeeks.org/java-io-bufferedreader-class-java/>

[3] <https://www.geeksforgeeks.org/io-bufferedwriter-class-methods-java/>

