**Overview**

This Program reads the input from Buffered Reader object which creates an array of characters to forms a grid and set of words need to fill in grid.

Several operations are conducted on grid and comes with output. Here are the functions written to deal with that scenarios:

* **loadPuzzle() Method :** This method comes with the buffered reader object as argument and creates a grid layout with set of strings which gets filled in grid.
* **solve() Method :** This method fills the grid with all the set of strings by checking all the combinations and directions of placeholders.
* **print() Method :** This method prints the formatted input to PrintWritter object in grid format with proper space.
* **choices() Method :** This method returns number of failed attempts while filling the grid with all combinations of strings.

**Files and External data:**

* There are no external files used while executing the program.

**Classes:**

1. **FillInPuzzle:**

* This Class has a sort of primary class which open ups a a character array for the grid. The loadPuzzle method reads input as Buffered Reader as objects in this class and designs the grid with given dimensions and stores all the set of strings.
* This class takes help of other class called “CrossWord.java” where the crossword object is instantiated, and the puzzle is solved.

1. **CrossWord:**

* This class contains methods like fitVertical, fitHorizantal, set,get, unfill which acts on the grid and words which makes out all combinations to fill the grid correctly.

**Data structures Used:**

* The data given by the user is stored in Character array, List of strings in a Set and which is implemented in different way by adding features of top and down.

**Assumptions:**

* No word is repeated in the set of input words.
* The puzzle is case invariant.
* The words in the puzzle all fill from left to right or from top to bottom.