R E A D M E

How to execute:

Preconditions:

→Make sure java directory path is set in your PC's environment variable

→Create a new folder in your PC.

→Copy the jar file "Ashwin_1800CodingChallenge.jar" in a newly created folder in your PC. Copy the dictionary data file in your PC, using the same directory is recommended, however not mandatory.

Use below command from your command prompt to execute the program:

*$>**Java – jar Ashwin_1800CodingChallenge.jar –d <dictionary file>***

The program takes one command line argument to set dictionary file

-d <Dictionary data file> : This will set the dictionary for the program to look up.

Source Code:

→ Extract Ashwin_Aconex1800_Challenge_SRC.zip as newly created folder. Java source file can be found under Aconex1800_challenge/src folder. Aconex1800_challenge/data folder contains the sample dictionary file used by me to test this program.

Why I chose this program:

I found this program challenging because this was not about search but finding all possible combinations of numbers and looking up for the exact match from the dictionary. This interests me because it is really interesting lively and practical.

Design:

Dictionary Class: I designed a dictionary class, which is also my main repository of dictionary data stored in a HashMap in a key value pair. Where key is the corresponding user defined keymap number and value is set of words matching with this number.

KeypadDictionary Class: This extends Dictionary class and allows user to define his own keypad by writing conversion method. This class has methods to read data from external file and prepare data in Hashmap format we require.

PhoneNumber Class: It contains set of phone numbers program is currently working on. Current implementation works on number at a time as per requirements, however there is a provision and abstract method available to implement later to find all combinations.

PhoneNumberParser Class: This is a supporting class to parse combinations of the number and finding results.

PhoneNumberOperation Class: This class implements methods to break the number in to all possible combinations which can be mapped to words in the dictionary. It uses a link list of Object PhoneNumberParser and iterates each elements of it to find match in the dictionary. It also moves the result to finalParsedList link from running parserList.

Assumptions:

Since 1,0 is not present in keymap it is considered as not matched. Two consecutive numbers can't remain unchanged, however 2 numbers can appear in different places in the combination result lists.

Note: If above assumptions are required to be altered, it can be easily.