Uttara InfoSolutions
www.uttarainfo.com

Prototype of Dictionary Project:

When the dictionary solution is run, the following should be displayed:

Press 1 to Create Dictionary

Press 2 to Load Dictionary

Press 3 to Exit

Enter choice:

If 1 is given as input,

a) you should first ask the user to enter name of Dictionary.

You should validate whether this dicionary name is already used by another as a business validation.

b) Then you should display the following menu:

Press 1 to Add a word

Press 2 to Edit a word

Press 3 to Remove a word

Press 4 to List the words

Press 5 to Search words

Press 6 to Go back

If 1 is selected,

i) you should ask the user to input a word text (this has to be unique)

ii) then display a list of semantics like this:

Choose 1 for Verb

Choose 2 for Noun

Choose 3 for Adjective...

...

Choose n to Goback

iii) you should validate that the user has not given the same semantic

earlier for the same word

iv) After user chooses a semantic, you should ask the user to input a meaning and then an example.

v) After this, you should display the Choose Semantic menu again (until user elects to goback)

Implementation tip: When Goback is selected, you should write to a file the following:

word_text = semantic1:semantic2:...::meaning1:meaning2:...::ex1:ex2:...

which are given as inputs for the word.

vi) When 2 is selected (Edit word), you should allow the user to a) remove semantic with its meaning and example or b) add a new semantic, meaning and example (again unique check to be performed)

vii) A user can remove a word that exits only. All its semantics,meanings,ex will be removed from dictionary

viii) When 4 is selected (List words), the user should be shown the following submenu:

Press 1 to list words alphabetically

Press 2 to list words by word length (ascending)

Press 3 to Goback

Depending on user input, the sorting of words have to be done.

ix) When 5 is selected (Search), you should ask the user to input a string to search.

This string has be searched in the entire contents of the Dictionary(word text, meanings & examples) and the following should be displayed:

Total number of occurances : <num>

Number of occurances in Word text: <num>

Matches found:

<word text1>

<word text2>

```
...

Number of occurances in Meanings text: <num>

<meaningtext1 - word it belongs to>

<meaningtext2 - word it belongs to>

...

Number of occurances in Examples text: <num>

<extext1 - word it belongs to>

<extext2 - word it belongs to>

...
```

In the outer menu, if 2 is given as input (Load Dictionary), you should ask the user to input the name

of dictionary (which should exist) and then you will display the same inner menu where in the user can

now add,edit,remove,list,search from the loaded dictionary.

Implementation tips:

a) First iteration: Implement the dictionary solution without using any OO techniques. Simply create

a class with main() method that interacts with user and provides functionality. No need to create any other class.

When a user inputs a Dictionary name as input, create a File instance with the dictionary name as constructor input,

verify if this file exists and if not, you can create the file.

It is into this file that you will write the contents once the user adds/edits/removes a word.

Use a FileWriter encapsulated in a BufferedWriter to do file writing (Pass true as its second parameter to append to the file).

In the file, one line represents one words input.

You can use BufferedWriters newLine() method to insert a new line in the file.

For removing a line (a word), you will have to copy the contents of the file temporarily to memory except the one line and

then write back to the same file, overwriting the contents.

b) Second iteration: Implement the Dictionary project by implmenting MVC design pattern, OO techniques and storage in both file and database.