Mike Dumas

Apr 5 2013

Project 3 OOD Outline

Program Should Spells Check a File

1. use string objects throughout this project
2. Read the dictionary file named dict.txt containing a list of correctly spelt words
   1. Store the words in a hash table all lowercase
3. Read in the file that we want to spell check provided by parameter
   1. Tokenize fields by space
   2. Keep track of the line a token came from
   3. Determine if the tokenized field is a word
      * 1. is of length greater than or equal to 1 its a word
        2. it is a sequence of non-whitespace character its a word
        3. if **first char is a letter** and **last char is not** its a word
           1. and It is necessary to **chop off the last char**
        4. if **first char is not a letter** but **last char is a letter** its a word
           1. and it is necessary to **chop off the first char**
        5. If first and last char is a letter its a word
   4. if not a word do nothing
   5. If token was a word Check against hash
      1. If word is found in hash than the word is spelt correctly
      2. If word is not in hash
         1. Need to create a Node struct that contains the string object and the line number it occurred on.
         2. Create a Node for each word that was not found and add it to a Binary Search Tree. Upon completion Tree shall contain all words that were not found.
         3. Display the menu
         4. Will need to send the miss spelled word as a parameter to menu method
         5. Duplicate misspelled words should only be printed once, printing the most recent occurrence
            1. Binary Search Tree, has a find method, you send it the key and a Node, and it checks every node of the tree. If a word is in the tree already, it is a duplicate miss spelled word.
         6. Menu Options
            1. Add the word to the dictionary (lowercase)

add the word to our dictionary object Dictionary File

* + - * 1. Ignore the word

Add the word to another Dictionary intended for ignored words(lowercase)

We want to flag it, the same way Word ignores foreign names and no longer underlines it. So program should no longer bring that word to the users attention.

* + - * 1. go on to the next word

Simply Move on

* + - * 1. Search for a suggested spelling in the dictionary

Exchange all adjacent characters in the misspelled word until it is found in hash or it has been swapped enough times, whichever comes first.

If word is successfully fixed move on to next word

if not return to menu so user can chose another option

* + - * 1. Quit spell checking the file

End the program

* + - 1. As the the menu continues when it has not found any more words the program should print a message and terminate.
      2. Before Termination, using the Binary Search Tree class have program do a preorder traversal of all notfound words. Instead of outputting to the screeen, it should output to a textfile named notfound.txt. The printTree methods accept an output stream pointer.