Assignment 5 : Hashtable Spell Checker

Given: Nov 29, 2012

Due Date: Dec 7, 2012

Time Due: Before Midnight (**no late assignments accepted, AND THIS TIME I MEAN MIDNIGHT**)

Hand in: A folder (named *A5 LastName FirstName*), submitted to the I: drive. Failure to properly do this will result in loss of marks.

Objectives:

- Using a Hash Table data structure to solve a problem
- Using java's HashTable but adding in separate chaining
- Implement a basic hash function

Assignment Overview

You are to write a program that will implement a very basic spell checker that uses a hash table to store a list of known wrong words.

The basic execution of the program is as follows:

1. Open a file that contains a list of known wrong words and how they should properly be spelled.
2. Fill a hash table based upon the wrong words in the file. Here the key will be the "*wrong word"* and the value stored will be the correct spelling for the word.
3. Open a text file that is to be checked for any wrong words
    a. Here you must check to see if the word exists in the Hash table you filled up in the previous step. If the wrong word exists in the hash table your program must add it to a list of found wrong word pairs.
4. The final list of found wrong words is then displayed on the console.
    a. For each wrong word it must indicated the line number the word was found on and the location in the line of the word. The location of the word is equal to the number of words that appear before it in the line plus 1. For example: "This is a test acheive". In this line "achieve" is spelt wrong and is the fifth word in the line.

Note:

- For this assignment you are given some basic classes that you must use along with a driver and skeleton class for the Spell Checker. More details about these classes are given later.

Provided Code:

- Driver.java – contains the main for the program
- SpellChecker.java – the skeleton class where you will insert your code.
    - Note: it would be a good idea to add other classes to create a good design.
    - Leave the current method definitions in the SpellChecker exactly as they are now so that the test driver used during marking can properly create and work with your class.
- WrongWordFound.java – A class used to return your final results.

Phase 1 : Fill the Hash Table with Wrong Words

For phase 1 of the project you have to open the file of known wrong words and fill the hash table. Once you have created the table using all the given wrong words you should print out the entire table to the console. Below are the basic steps you should follow to complete this phase:

1. Create a class to store the wrong word and its proper spelling.
    a. This class will be used to hold information for each element in the table
2. Next you will have to fill the hash table based upon the input file.
    a. You can use the Hashtable class (java.util) for this assignment. But you will have to add in the idea of separate chaining. You can use an ArrayList for the chains
        i. The key values will be in relation to a “wrong word”
    b. The name of the file containing the list of wrong words is passed into the constructor of the SpellChecker class. From here you must open the file and read in all the information and store it in the tree.
    c. Each line in the file has the following format:
        i. [wrongword][space][correct word]
        ii. Example: accross across
    d. You will not have to deal with any error checking of the input file. It will always be in this format
    e. As you read in each line you should break it into the wrong word and correct spelling and then create a new object to store this information. It should be stored in the class you created in step one.
3. Insert the word into your hash table
4. Once you have this phase completed in full and verified, comment out your print code before moving onto phase 2.

Hash Function design requirements

You will have to implement a polynomial hash function as discussed in class.

Considering the word to be an array *W* of characters of length *N* where the values of the letters are from 0 to 25 (i.e. int charVal(char ch){return ch - 'a';}), all letters in the word must be in lower case. Then the hash function you must implement is:

$$\text{Hash}(W) = W[0]a^{n-1} + W[1]a^{n-2} + \ldots + W[N\text{-}1]a^{0}$$

Where '*a*' > 1. For this project the value you will use for 'a' is determined in the following way:

1. Compute the following value: Your mount royal id number mod 4
2. If the result is 0 then your value of a will be 33
3. If the result is 1 then your value of a will be 37
4. If the result is 2 then your value of a will be 39
5. If the result is 3 then your value of a will be 41

Note: You will need to override the hashCode() method in the class that you will use to represent the key. All code for this hash function should go into that method.

Phase 2: Checking a file for spelling mistakes

In this phase you will now be using your Hash table to check a file for possible spelling mistakes based upon a list of known wrong words. In the SpellChecker class you will find a method called *checkFile.* The basics steps are given below:

1. Open the file using the name passed to the checkFile method
2. **Note:** for all test input files you will never have to deal with any kind of punctuation or brackets. Every line will only have words separated by spaces.
3. Loop through all the words in the file
    a. Check to see if the word is in your wrong word hash table.
        i. If the word is in your table create a new WrongWordFound object and store all necessary information and add it to your result list
4. Return the array list of results from the *checkFile* method.
5. If you have done everything proper then the main driver should print out all the found spelling mistakes.

For example given the following text:

This is a test acheive
another ecstacy test

The resulting output should be:

WrongWordFound [line=1, wordLocation=5, originalWord=acheive, correctWord=achieve]
WrongWordFound [line=2, wordLocation=2, originalWord=ecstacy, correctWord=ecstasy]