CSCI 2235
Programming Assignment 04 - SpellChecker

Assigned: November 14, 2019
Due: December 06, 2019 @ 23:00h

Purpose

- Explore the implementation of Tries
- Explore the use of Tries in the construction of a Spell Checker
- Learn about Unit testing.

Problem Statement

Spellchecking is an extremely useful component of any editing software. In this assignment, you will implement a Trie and use this to spell check a series of documents. In addition to this, you will need to implement a series of tests for both your spellchecker and for your trie.

In addition to this document you have been provided a dictionary containing a large selection of English words. You will need to utilize this, a Trie, and your own personal ingenuity to implement a spellchecker which meets the requirements below. **Take Note: the dictionary contains 152,512 words, so you will need to adjust your JVM memory settings appropriately.** When testing your program, I would suggest creating a data directory in your project where the dictionary will be stored and the creation of a smaller dictionary for testing. This smaller dictionary can be created by selecting a much smaller number of words randomly sampled from the main dictionary. I would then store it in the data folder with a name like "test.dic".

Assignment

0. Initialize a new Gradle project, using the gradle command at the command line, with the following characteristics:
    - Create a directory named `pa04` in your normal project directory
    - Open a command prompt/terminal and change directories to the newly created `pa04`
    - From within the `pa04` directory run the following command: `gradle init` and provide the following information
        - Type of project: 4 // Java application
        - Build script DSL: 1 // groovy
        - Test framework: 1 // junit
        - Project name: [enter] // default should be `pa04`

    - Source package: edu.isu.cs2235
  - This will create several files in your directory. Open the build.gradle file note the last line of the file says: mainClassName = 'edu.isu.cs2235.App'
  - You will need to change this to the fully qualified name of the main class for your project, in previous projects this was Driver. You can keep app but you will need to modify it to complete the assignment. Also note that there is a test class called AppTest which if you delete App, you will need to delete AppTest as well.

1. Implement a Trie Data Structure. I would suggest that you follow similar programming constructs as we did in prior assignments. Programming By Contract, Abstraction, Encapsulation, etc.

- Implement your Trie to validate that it works correctly
  - Create tests to evaluate your Trie Structure, based on the lecture on Unit Testing. Your tests will be graded for effectiveness.

2. Implement a SpellChecker using your Trie
   - Read in a provided dictionary of English words into your Trie
   - Create an interface which allows a person to enter either a string of text to spell check, or -1 to quit.
   - Then for each word in the text that is misspelled, show the word in context, specify which is misspelled, and provide 3-5 potential spellings to correct the text or allow the user to skip.
   - Then once all misspelled words are handled, print out the resulting string to the console.
   - Finally allow the user to start the process over again.
   - Create a set of tests to test your spellchecker functionality. Your tests will be graded for effectiveness.

Example Program Operation

```
Loading dictionary...
Dictionary loaded.

Enter a string to spellcheck (−1 to quit): Hello World! I m finaly hear!

Checking Spelling...

Misspelling found: "m" in ... I m finaly ...
Replace With:
1. me
2. my
3. man
4. Manual Entry
5. Skip
Choice: 4

Manual Replacement for "m": am
Replaced "... I m finaly ..." with "... I am finaly ..."

Misspelling found: "finaly" in ... am finaly hear!
```

```
Replace With:
1. finally
2. finalize
3. finalized
4. Manual Entry
5. Skip
Choice: 5

Replaced "... am finaly hear!" with "... am finally hear!"

Spelling Check Complete:

Results: Hello World! I am finally hear!

Enter a string to spellcheck (-1 to quit): -1

Good bye.
```

Submission

Submit a zip file containing your project directory to moodle by the deadline.

Grading

10 Points - Implementation of Trie 10 Points - Effective Testing of the Trie 20 Points - Implementation of the SpellChecker 10 Points - Effective Testing of the SpellChecker