

Assignment 2
Due Feb 14

The goal of this assignment is to improve on your assignment one and implement Iterator, visitor and strategy patterns.

1. Modify your Trie classes to use standard collection method names in your language.
2. Implement an iterator on your Trie class to iterate through all the words in the trie.
3. Use the iterator to select all words in the Trie that contains the letters "ck" per assignment 1.
4. Implement a toString() method on your Trie class that returns a string representation of the Trie object. If you are not using Java use the name appropriate to your language.
5. Implement the Visitor pattern on your Trie. The visitor pattern uses different types of objects. While a bit unnatural use three types of node in your Trie. First add a null node at the end path in the Trie. Second use a word node and a non-word node. A word node indicates that the character sequence to that node in the trie is a word. A non-word node indicates that the character sequence through that node is not a word. Be careful with the Visitor pattern. Students often take shortcuts in the pattern which results in them not implementing the Visitor pattern.
6. Implement two different visitors for your Trie. The first one collects all the words in the Trie. The second visitor collects all the words in the Trie that contain the letters "ck".
7. Implement a third visitor for your Trie. This visitor accepts a strategy object that the the visitor uses to determine which words in the the trie to collect. Implement two strategy objects. One that you can use to find all words that contain more than two vowels. Another one that you can use to find all words that contain the letters "th" without any other letters between them.
8. Write unit test for the functionality above. You might find it instructive to write tests for your current classes and refactor and modify your existing code to meet the above requirements.
9. If you are not using a source control system you might consider doing so now. If you use Eclipse try mercurial. The Eclipse plugin can be found at <https://bitbucket.org/mercurialeclipse/main/wiki/Home>.

Grading

	Percent of Grade
Working Code	15%
Unit Tests	15%
Quality of Code	20%
Proper implementation of Patterns	50%

What to Turn in

Turn in hard copy of your code.

Late Policy

An assignment turned in 1-7 days late, will lose 3% of the total value of the assignment per day late. The eighth day late the penalty will be 40% of the assignment, the ninth day late the penalty will be 60%, after the ninth day late the penalty will be 90%. Once a solution to an assignment has been posted or discussed in class, the assignment will no longer be accepted. Late penalties are always rounded up to the next integer value.

Comments

Please no ascii based menu systems to run the code. Use Xunit tests instead. There are lot of patterns in this assignment. More than one should use in normal code. Also some of the patterns are excessive for the situation. The goal is to give you some experience implementing patterns, so we are using more than one would normally use.