CS 635 Advanced OO Design and Programming
Spring Semester, 2012
Assignment 2
© 2012, All Rights Reserved, SDSU & Roger Whitney
San Diego State University -- This page last updated 1/31/12

Assignment 2
Due Feb 21, 2012

The goal of this assignment is to implement the null object, iterator, strategy and visitor patterns.

1. Implement a doubly linked with null objects at the beginning and end of the list. So an empty list will contain two null objects. The linked list class is not restricted in the type it can contain. However you can use templates or generics, which require an instance of the linked list class to contain the same type of objects.
2. Implement method(s) to add new items in the list. Items are kept in order in the list. Use the strategy pattern so client code can indicate the order of the list.
3. Implement a method to determine if an item is in the linked list. This method is to return true if the item is in the list, and false if the item is not in the list. This method is to do its work by sending a message to null node in the front of the list. The nodes will pass the message on to the next node if needed. (Some people call this a recursive implementation.)
4. Implement an iterator to iterate over the linked list from front to back. The goal is to implement an iterator not just use an existing iterator, so don't convert your list to an existing collection to use its iterator.
5. Implement an even size string filter that implements the iterator interface in your language. The filter reads data from an iterator. The next operator (using Java's iterator interface method names) only returns strings that have an even length. That is it filters out all strings of odd length. Be careful with the hasNext() method. It only returns true when there are remaining strings that are of even length.
6. Implement a visitor that constructs a string representation of the elements in the linked list. The representation starts with and "(" is followed by the elements in the linked which are separated by a " - " and ends with ")". So an empty list results in "( )". A list with one integer 42 results in "(42)", a list with the strings (in order from front to back of the list) "cat", "dog", "rat" results in "(cat - dog - rat)". The quotes in the above are just there to denote in the text the start and end of special tokens and are not intended to be part of the programs output. Be careful here. The structure here is so simple it is easy to simplify one's code so that it is not an implementation of the visitor pattern.
7. Write unit test for the functionality above. You might find it instructive to write tests for your current linked list and refactor and modify your existing code to meet the above requirements.

You are to write your own linked-list code.

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 eight 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 unit 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.