

Assignment 2  
Binary Search Tree Revisited

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

Due March 1

1. Make sure that you have a copy of your unit tests from assignment 1. When you are done with this assignment determine how good the tests were. That is after making the changes required in this assignment how confident are you that your code works after running the tests. Did you have to write new tests as you refactored your code?
2. Refactor your tree code to use standard names for methods, remove helper methods on the Tree that deal with nodes, and any other clean up you feel is needed in your code. You might find the refactorings rename and move useful here. In Eclipse these refactorings can be found in the Refactoring menu.
3. A Binary Search Tree is a collection. Determine the correct location in your language's collection class hierarchy. Find **all** methods that you need to implement in-order to add your class in the language's collection class hierarchy.
4. Make the parent class of your tree class the parent determined in problem 3. Rename your existing methods to conform to the collection classes standards. You may need to stub some methods to satisfy the parent class's constraints. Note we will only be interested in implementing a few of these methods. You do not have to implement all the methods in the parent class. We will need at least the add method, toArray and the toString method. As in assignment 1 don't use arrays or other collection classes to implement your tree.
5. Use the strategy pattern to allow you determine how the tree will be ordered when a tree object is created.
6. Implement an iterator for your Tree class. Don't covert your tree to an array or other collection to implement your iterator. You can implement either an internal or external Iterator.
7. Use the null object pattern to represent head and leaf nodes in the tree. Can you remove any null checks?
8. Implement two separate and independent decorators for Java's Iterator. The first decorator only return Strings that start with a vowel. The constructor for the decorator needs one argument of type Iterator<String>. The second decorator also only works on Strings. The second iterator capitalizes all the strings it returns in its next() method.
9. Write unit tests for the code you write for this assignment.

## Grading

Item	Points
Working Code	10
Unit Tests	10
Proper implementation of Patterns	10 per Pattern (40 points total)
Quality of Code	10
Proper Parent Class & Method names	10
Written Answers to questions 1,7	5

### Turning in your Assignment

The assignment is given as a series of steps. Turn in the code as it is when you have finished all steps. Do not turn in multiple copies of the same class.