CS 635 Advanced OO Design and Programming
Spring Semester, 2014
Assignment 2

San Diego State University -- This page last updated 2/18/14

Assignment 2
Min-Heap Revisited

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

Due Mar 6

1. Review your units tests for adding elements to the Min Heap from assignment 1. Make sure that the tests adequately test adding elements to the heap. Record those tests. When you are done with the assignment determine how good the tests were. That after making the changes required in this assignment you were confident that worked after running the tests. Did you have add to or modify your tests?

2. Refactor your tree code to use standard names for methods, remove helper methods on the heap that deal with heap 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. The heap class 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. (C++ people get a pass on this problem as STL is painful to subclass.)

4. Make the parent class of your heap the parent determined in problem 1. Rename your existing methods to conform to the collection classes standards. One 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 heap.

5. Use the strategy pattern to allow your heap to be either a min heap or a max heap.

6. Implement an iterator for your heap. Using an in-order traversal will be easier than preorder or post-order. Don't covert your heap to an array or other collection to implement your iterator.

7. Use the null object pattern to remove null checks when finding the height of the heap. Can you remove other null checks?

8. Implement what we will for now will call IngFilter. Using Java syntax the class will have the methods given below. People using other languages may need to implement different methods to conform to their language's conventions.

IngFilter(Iterator input) - constructor

boolean hasNext() - returns true if the iteration has more elements that end in "ing".

next() - returns the next element in the iteration that ends in "ing".

9. Create a decorator class IngHeapDecorator that decorates your new Heap class. The decorator modifies the toString, toArray and iterator methods to return just elements ending in "ing. Does this make sense as a decorator?

Grading

Item	Points
Working Code	10
Unit Tests	10
Proper implementation of Patterns	14 per Pattern (70 points total)
Quality of Code	10
Proper Parent Class & Method names	10

Turning in your Assignment

Turn in a hard copy of your assignment.