CS 635 Advanced OO Design and Programming
Fall Semester, 2021
Assignment 2

San Diego State University -- This page last updated 9/15/21

Assignment 2
B-Tree Revisited

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

Due Oct 3

1. Refactor your B-Tree code to use standard names for methods, remove helper methods on the B-Tree that deal with B-Tree nodes, and any other clean up you feel is needed in your code. You might find the refactorings rename and move useful here.

2. The B-Tree 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.

3. Make the parent class of your B-Tree the parent determined in problem 3. 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 B-Tree.

4. Use the strategy pattern to allow clients to how your B-Tree orders the elements. You need to implement two different strategies. One strategy will have the B-Tree order Student objects lexicographically by name. The other strategy will have the B-Tree order Student objects by GPA.

5. Implement an external iterator for your B-Tree. Using an in-order traversal. (The iterator needs to keep a stack of nodes from the current node back to the root. The stack also needs to store for each node how many times each node has been visited.) Don't covert your B-Tree to an array or other collection to implement your iterator.

6. Implement an internal iterator traversing in reverse order for your B-Tree.

7. Use the null object pattern to remove null checks to see if you are at the end of the B-Tree. Can you remove other null checks?

8. Write unit tests for your code.

Grading

Item	Points
Working Code	10
Unit Tests	10
Proper implementation of Patterns	20 points per pattern
Quality of Code	10
Proper Parent Class & Method names	10

Turning in your Assignment

Turn in zip file of your source code in the course web portal.