CS 3410 - Homework 03

Due date: see course Schedule and Blackboard.

Overview

You will do several problems from the text. I have provided you with code that implements the BinaryTree data structure discussed in class and a driver to test your solutions.

Requirements

Obtain the download, *hw3.zip*. If you use JGrasp there is a project file, *hw3.gpj*. Open the driver which is in the file *HW3.java*. There, you will find 5 method stubs for you to write the code for these 5 problems from the text, p.683: Problem 18.9 a,b,c & Problem 18.10 a,b. For Extra Credit: Problem 18.10e (15 points)

Note that for Problem 18.10, it says to use a binary tree with integers. The code you have been provided uses a binary tree of Person objects. The Person class has a method, *getSales()* that returns an integer. So, you will use that.

Suggestions

Familiarize yourself with the BinaryTree API in the notes and the downloaded code. You'll need to know a few methods in the BinaryTree class and the BinaryNode class.

There are 5 testing methods whose code is written: p18_9aNumLeavesTest(), p18_9bOneChildTest(), p18_9cTwoChildrenTest(), p18_10aNumEvenTest(), p18_10bSumTest(). Comment out all the ones you are not working on. Further, inside each method are up to 7 tests. Comment some of those out as well, until you think you have your code working properly.

The testing methods utilize a number of hard-coded methods to create binary trees for testing: createTestTree1, createTestTree2, *etc*. You should not modify these. However, you can write your own.

Each test in the testing methods calls the *printPreOrderMethod* which prints the tree with indentation to show the levels. The output of this should be used to support your testing results.

Deliverables

1. All Java files.
2. A Word document in this format:
   a. Title page:
      CS 3410 – HW 02
      Name
      Date
   b. A summary of your progress on this homework.
   c. An annotated sequence of testing results for each part of each problem. In other words, copy the results from running your program and annotate as necessary to show that your methods work.
3. Zip the Java files and Word document and submit via Blackboard.