Assignment 4

Objectives:

- To create a binary search tree.
- To create an iterator that will traverse a binary tree in preorder, postorder and inorder.
- To have exposure to inheritance, recursion, abstraction, encapsulation and modularity.
- To have exposure to unit testing.

Introduction

A common theme in data maintenance is the storage, access and ordering of individual records of a larger system. For instance, in the Health Care System, patient medication lists must be kept up to date to monitor potential drug interactions and side effects. Medications need to be changed or dosages adjusted and the access to an individual patient's record must be efficient and accurate.

For this assignment, you will implement a class PatientMedRecord that simulates a record of a patient's current list of medications. All the records will be stored in a data structure, a BinarySearchTree, using the unique patient id number as the key. The textbook, lectures and labs are the resources for this assignment.

Steps to completing the assignment.

1) Download `PatientMedRecord.java`, `AbstractBinaryTree.java`, `TreeNode.java`, `BinarySearchTree.java`, `BinaryTreeIterator.java`, and `DrawableBTree.java` into a single assignment folder specifically created for this assignment.
2) All the files will compile; run `javac *.java`. During the process of completely the assignment, check to make sure everything still compiles before continuing. Note that the public methods in BinarySearchTree take their comments from the BinaryTreeAbstract file, so you do not have to repeat the method headers.
3) You will be making use of several classes that are created for you. To keep track of what is available to you, create a mini-API documentation set for yourself by running `javadoc *.java`. This creates many files; double-click on the index.html and you will have all the information you need to use the created classes and their public methods.
4) One of the two files you are to complete is the BinarySearchTree.java. Some of the code has been started for you and you are to complete it. The public methods for the BinarySearchTree make calls to private recursive methods, which you are to complete.
    a) Note that it is standard protocol to let the private methods do the actual recursion.

b) Note also that the TreeNode datatype should never be publicly accessible; the actual structure is not the user's concern, and a user could gain access to the rest of the tree by accessing the TreeNode's left and right child.

5) The second file you are to complete is the BinaryTreeIterator.java file. This class implements the java.util.Iterator interface. You can find the specifications for the required public methods called `next`, `hasNext` and `remove` in the API documentation for the java.util.Iterator.

Some details to consider

All the proper commenting is provided for you in this assignment. Please put your name and V number in a header at the top of your file.

There are many lines of java code provided and there are hidden gems to learn from the professional style of the programming; please take advantage of these, by reading the code thoroughly, until you understand it. You do not need to know anything about the DrawableBTree class. It is a gift to help you visualize the tree.

You must test the functionality of your programming yourself. This is to be done in the main method (*unit tester*) of the BinarySearchTree; the testing of the BinaryTreeIterator class can also be done in the BinarySearchTree main method, by invoking the two iterator methods there. Some code has been started for you. Use this to test every one of the public methods.

You may assume that in this assignment, there will be no duplicate PatientMedRecord objects ever inserted in the BinarySearchTree.

To assist you in testing the BinarySeachTree results, we provide the DrawableBSTree class, which allows you to visualize the current tree as you progressively test your code. When you want to see what the tree looks like, uncomment the two lines of code you see in the main method BinarySearchTree. At that point, the program pauses and a frame becomes visible on your computer. You can resize the frame as desired and the tree will expand to fit the frame. When you close the frame, control goes back to the main method.

Submission

Submit your BinarySearchTree.java and BinaryTreeIterator.java using conneX. **Please be sure you submit your assignment, not just save a draft**.

A reminder that it is OK to talk about your assignment with your classmates, and you are encouraged to design solutions together, but each student must implement their own solution.

We will be using plagiarism detection software on your assignment submissions.

Grading

If you submit something that does not compile, you will receive a grade of 0 for the assignment. It is your responsibility to make sure you submit the correct files.

Requirement	Marks
You submit something that compiles	expected
Proper style as per coding conventions	expected
1 mark for each test case you pass	9
Evidence of thorough unit testing	1