Learning Activity: SimpleSet

Back

Educational Objectives

- Gain familiarity with basic binary search tress.

Instructions

Download a copy of SimpleSet.java. Implement this interface within the package csc143.data_strctures. Name the implementation class **SimpleTreeSet**. Use a binary search tree as the underlying data structure.

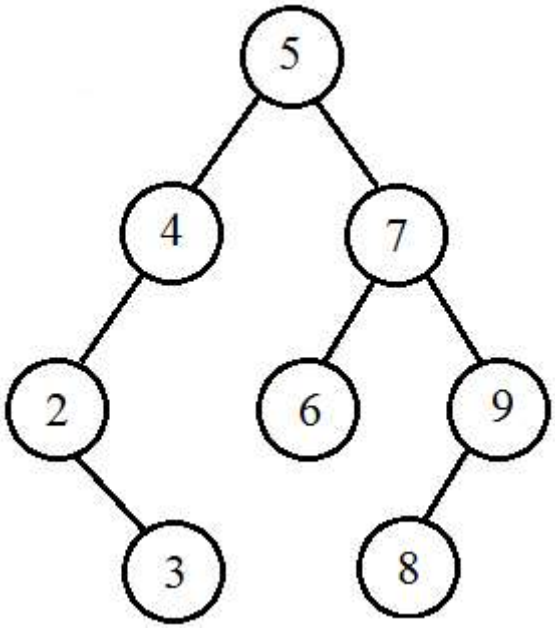
Implementation notes:

- Binary search trees require some mechanism to order the values. In keeping with the tree-based collection in java.util, this implementation of SimpleSet will use the *natural order* of the generic type parameter, E. This natural order is supplied by the java.lang.Comparable interface. So, your implementation can cast the values it is working with to java.lang.Comparable as required. If the object does not implement this interface, a ClassCastException will be thrown, an unchecked exception. That is to be expected. This cast is an unchecked conversion. Use the @SuppressWarnings annotation to compile without warning messages. Minimize the scope of the @SuppressWarnings annotation. My implementation uses only one, on a constructor which only contains assignments for the fields of the class.
- **add**
  The algorithm in the notes uses an expensive exception to indicate a duplicate value. For the Check Level, you may implement add using the exception mechanism. For the learning activity, do not be concerned about efficiency, that is, there is no requirement to implement any form of balancing for the underlying BST.
- **size**
  The method can be implemented using either a special field or recursive calculation.
- **toString**
  Facilitates debugging and checking. More information about toString follows this list.
- Recursion is your friend. Make sure that you structure it "appropriately". That is, the recursive methods are not necessarily the methods inherited from SimpleSet.
- **Important**:
  Do NOT use any resources from the java.util package. You must create the underlying binary search tree "from scratch."

The toString Method

The toString method should show the structure of the tree, not only its contents. These shall use a *pre-order* traversal of the tree. Use nested parenthetical expressions, in which each tree or subtree is indicated by enclosing it in parentheses. Empty trees are indicated by ( ).

For example, given the following BST:



The nested parenthetical toString would give:

```
(5 (4 (2 () (3 () ())) ()) (7 (6 () ()) (9 (8 () ()) ())))
```

Note: There is no space immediately after the opening paren or immediately before the closing paren. However, there is one space between the value for a given node and the left subtree, and one space between the two subtrees.

Plus Level

For the Plus Level, implement the add method without using an exception to indicate a duplicate. Yes, this means that your recursive add function can return a boolean value. So, the add method of the set can return the boolean value from the recursive method.

Submission

1. One (1) Java archive file (.jar) containing SimpleTreeSet,
   both source code and bytecode, with the proper folder structure in the archive.

Evaluation Criteria

Check – implementation compiles with no errors *or warnings* and behaves correctly.
Minus – implementation is mostly correct.
Plus – add method does not use exceptions to signal a duplicate value.