CS 474: Object Oriented Programming Languages and Environments

Spring 2014

First Smallalk project

Due time: 7:00 pm on Friday 2/21/2014

You are required to create a simple *Set calculator* program in Cincom Smalltalk while taking advantage of Smalltalk's automatic GUI builder. The calculator allows an interactive user to enter and edit information about two sets containing integer numbers. The calculator supports basic set operations such as union and intersection between the two sets. *For efficiency reasons, you are required to implement each set as a Binary Search Tree (BST).*

At any point in time, the user will edit and modify one the two sets, that is, S_1. When the editing is complete, the user can save S_1 as the second set S_2. This operation will deep copy the first set (S_1) into the second set; however, S_1 is not modified. This set could be edited further. Set operations take place between S_1 and S_2; operation results are always stored in S_1.

Recall that BSTs are binary trees subject to the following properties. First, a numeric value is associated with each node. Second, each node can have at most two children, a left child and a right child. Third, given a node *x*, the values of all nodes in the left subtree of *x* are less than the value of *x*, and the values of all nodes in the right subtree of *x* are greater than the value of *x*. No duplicate values will be allowed in your trees.

The GUI of your program should support the following functionality; you should choose an appropriate Smalltalk widget to implement each piece of functionality.

1. *Clear set.* This function allows interactive users to delete the current S_1 set. The previous value stored in S_1 is lost.

2. *Switch sets.* The sets associated with S_1 and S_2 are swapped, meaning that S_1 will receive the previous S_2 set and vice versa.

3. *Save set.* The S_1 is copied into S_2. The previous content of S_2 is lost. The content of S_1 is not affected. The two sets must not share any data structures, that is, they can be modified independently of each other.

4. *Display set contents.* The numeric values stored in the two sets are displayed as two ordered lists (non-decreasing order) in an appropriate widget or widgets. The two sets are not modified.

5. *Add element.* This function allows a user to add a new integer to S_1. The value is read from an appropriate *line input* widget. No action is taken if the number in question is already in the set. The insertion should preserve the BST properties of S_1.

6. *Remove element.* This function allows a user to remove an element from S_1. The value is entered from an appropriate *line input* widget. No action is taken if the number in question is not in the set. Otherwise, the nodes should be rearranged in such a way as to preserve the properties of a BST in the resulting tree.

7. *Union.* This element takes the set union of S_1 and stores the resulting value in S_1. The previous content of S_1 is lost. S_2 is not modified by this operation.

8. *Intersection.* This element takes the set intersection of S_1 and stores the resulting value in S_1. The previous content of S_1 is lost. S_2 is not modified by this operation.

9. *Transform.* The user specifies a one-argument block in proper block syntax in an appropriate widget in your interface. The block argument is assumed to be an integer number. The block is applied to all elements in set S_1. S_1 is destructively modified by replacing each S_1 element of with the value returned by executing the block on that element. (Hint: use the class method readFromString: to create an executable block object.)

You must work alone on this project. Your project code should be in a special package called CS474. Save all your code by filing out that package in the file xxx.st, where xxx denotes your last name. **You are required to implement all set operations on the binary search tree representation of sets. You are not allowed to use alternative representations of sets, such as the Smalltalk *List* class or *Set* class.** Submit the file using the submit link in the assignment page of the Blackboard course web site. No late submissions will be accepted.