CS 474: Object Oriented Programming Languages and Environments

Fall 2012

First Smallalk project

Due time: 7:00 pm on Friday 9/28/2012

You are required to create a simple *character parser* program in Cincom Smalltalk while taking advantage of Smalltalk's automatic GUI builder. The parser allows an interactive user to enter a sequence of words separated by blank characters. Next, the parser will scan all non-blank characters alphabetic and numerical characters from the input word sequence and arrange them in a binary search tree, which is explained below. The binary search tree will be sorted by lexicographic ordering of the characters stored in the tree. Each node will hold two items of information, namely a character and a positive integer indicating how many times the character occurs in the input word sequence.

Recall that BSTs are binary trees subject to the following properties. First, each node stores at least one key value (a character in this case). Second, each node can have at most two children, a left child and a right child. Third, given a node x, the key 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 key values (i.e., characters in our case) values will be allowed in your trees. Of course, our tree nodes will have an additional item, the integer number denoting the number of occurrences of the corresponding character in the input sequence.

The GUI of your program should minimally contain widgets to support the following functionality. Please choose an appropriate widget or widgets from the GUI builder's palette for each piece of functionality.

1. *Entering input word sequence*—You may assume that this input sequence would fit in a single line of text.
2. *Perform tree construction*—Once the input word sequence is complete, an interactive user would call this functionality to create a new BST. The previous BST, if any, will magically disappear.
3. *List the current tree nodes in pre-order*—When a user calls this functionality, the values of nodes currently stored in the tree are listed in pre-order and displayed in an appropriate widget.
4. *List the current tree nodes in in-order*—When a user calls this functionality, the values of nodes currently stored in the tree are listed in pre-order and displayed in an appropriate widget, different from the pre-order listing.

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. Submit the file using the submit link in the assignment page of the Blackboard course web site. No late submissions will be accepted.