ASSIGNMENT 5  Data Structures
CSI310
Spring 2011

EARLYBIRD EXTRA: Wed., Feb 9 at 11:59PM. DUE: Monday, Feb 14 at 11:59PM. Upload all your .java files for this project as attachments. Optionally, you may put them all in a .zip or .tar archive and upload one archive file.

You will need to have installed a Java programming system like Dr. Java (and the JDK except on a MAC) and have downloaded and added the Book Classes directory to your CLASSPATH. See the CSI310 Website and/or the textbook for further information. An critical part of the assignment is discussed below in the “Data Structures Requirement” section below. Sections 6.3, 7.1 and 7.2 of chapter of the “Problem Solving with Data Structures” textbook, Assignment 3, correspondence on Blackboard discussion groups, the lectures and possibly another UAlbany Supplement are the references for this assignment.

The next project (Assignment 6) will build on your work for this assignment. Stay tuned. It will require you to add the capacity for cutting, copying or pasting at any of the positions in the list, then redisplaying the combinations, and repeating the edit cycle any number of times.

The task: Make a program that does almost the same thing as what the Assignment 3 program should do, except that it MUST use a linked list data structure to store the references to the Pictures containing the imported images.

The one difference: Instead of asking the user how many images to import at first, it asks whether or not the user wants to import ANOTHER image.

When the user indicates “yes”, the File Chooser must pop up, one more image must get imported, and then the program repeats the first question.

When the user indicates “no”, the program goes on to create the three Pictures each displaying all the imported images.

Data Structure Requirement

You must define three or more classes including:

1. The application container class with the main() method.

2. One class for a linked list node fashioned after Guzdial and Ericson’s PositionSceneElement class of Chapter 7 or their SongNode class of Chapter 6.

3. Suppose you chose to name your linked list node class “PositionedSceneElement” Then one class for an entire list of Pictures is required, to be fashioned as follows:

```java
import java.util.Scanner; //For asking the user yes or no.
class PictureList
{
    private PostionSceneElement refToFirstElement;
    private PostionSceneElement refToLastElement;
    public PictureList()
    {
        this.refToFirstElement = null;
        this.refToLastElement = null;
    }

```
public void showAllTheImagesSeparately()  
//Implement this method for TESTING purposes!
{ /* You write! */ }

public void getImagesFromUser()  
{ /* You write! */ }

public void insertOnePictureAtEnd(Picture x)  
//Public for testing purposes. This method would be used by  
//getImagesFromUser().
{  
  PositionSceneElement myNewNode = new PositionSceneElement(/* ?? */);
  if(refToFirstElement == null)  
  { /* You write! The empty list case is special. */
  }
  else  
  { /* You write! */
  }
}

public int totalWidth()  
//It's public so you can test it from main().
{ /* You write! */ }
public int maxHeight()  
//It’s public so you can test it from main().
{ /* You write! */ }

public Picture makeTopJustifiedCombo()  
{ /* You write! */ }
public Picture makeCenteredCombo()  
{ /* You write! */ }
public Picture makeBottomJustifiedCombo()  
{ /* You write! */ }

Note: You may use a GUI interface to ask the user yes or no, if perhaps you know how to program a  
GUI facility better than using the Scanner class. (In that case, omit my import statement but you might  
need to put in a different one.) No extra credit for GUIs though, but your friends might be more impressed!