Jack Cobb

Project 2 PDF

1). My overall design was to have the main handle input, then pass to the database which tells the tree what to do. The database gets back information from the tree and outputs the commands for user interaction. The tree just held a reference to the root node, which was of type Node. Three other nodes extended the abstract Node class. They are ENode, INode, and LNode which represent all the differing types of nodes in this project. The nodes handled the different methods individually based on how they needed to. This was based on a node centric design instead of tree centric implementation. This was implemented alongside the functionality of the bsts that indexed the cities by name and by population. This allows cities to be indexed in three different ways and search for them in multiple ways.

However, I did not include my old bst implementation because it did not work correctly. I instead added a placeholder class that explains how it would function with other classes in the overall design, but does not actually implement methods because they did not function correctly.

2) In order for find to search efficiently, I used the bounds of the quadrants that were passed down recursively to figure out which quadrant the internal node should call to continue the search for the location. This was done with the current x and current y that gets passed down recursively and compares the desired location versus the current location. The rfind works similarly, except that all quadrants that intersect the rectangles bounds also get called to find elements within the rectangle and add them to an arraylist that later gets printed out.