EE 602
Java Programming Assignment #3
Hualiang Li

Red-Black Tree Junit tes plan

Insertion:

Case 1: The current node N is at the root of the tree.
Case 2: The current node's parent P is black, so property 4 (both children of every red node are black) is not invalidated.
Case 3: If both the parent P and the uncle U are red, then both of them can be repainted black and the grandparent G becomes red (to maintain property 5 (all paths from any given node to its leaf nodes contain the same number of black nodes)).
Case 4: The parent P is red but the uncle U is black; also, the current node N is the right child of P, and P in turn is the left child of its parent G.RedBlackTree tree = new RedBlackTree();
Case 5: The parent P is red but the uncle U is black, the current node N is the left child of P, and P is the left child of its parent G.

Deletion:

Case 1: N is the new root. In this case, we are done. We removed one black node from every path, and the new root is black, so the properties are preserved.
Case 2: sibling is red.
Case 3: P, sibling, and sibling's children are black.
Case 4: sibling and sibling's children are black, but P is red.
Case 5: sibling is black, sibling's left child is red, sibling's right child is black, and N is the left child of its parent.
Case 6: sibling is black, sibling's right child is red, and N is the left child of its parent P.