package BST;

public class BST {

Node root;

BST()

{

root = null;

}

Node Chen(int x , Node root)

{

Node P = new Node(x);

if(root == null)

{

root = P;

}

else

{

if(root.data == x)

{

System.***out***.println(x+"Da ton tai");

}

else

{

if(x > root.data)

{

root.r = Chen(x, root.r);

}

else

{

root.l = Chen(x,root.l);

}

}

}

return root;

}

void Taocay()

{

root = Chen(6,root);root = Chen(5,root);root = Chen(7,root);

root = Chen(3,root);root = Chen(2,root);root = Chen(9,root);

root = Chen(4,root);root = Chen(8,root);

}

void PreOrder(Node root)

{

if(root!=null)

{

System.***out***.print(root.data);

PreOrder(root.l);

PreOrder(root.r);

}

}

void InOrder(Node root)

{

if(root!=null)

{

InOrder(root.l);

System.***out***.print(root.data);

InOrder(root.r);

}

}

void PostOrder(Node root)

{

if(root!=null)

{

PostOrder(root.l);

PostOrder(root.r);

System.***out***.print(root.data);

}

}

Node Delete(Node root,int x)

{

if(root!=null)

{

if(x < root.data)

{

root.l = Delete(root.l,x);

}

if(x>root.data)

{

root.r = Delete(root.r,x);

}

else

{

if(root.l == null)

return root.r;

if(root.r == null)

return root.l;

root.data = minValue(root.r);

root.r = Delete(root.r,root.data);

}

}

return root;

}

int minValue(Node root)

{

int minv = root.data;

while(root.l!=null)

{

minv = root.l.data;

root = root.l;

}

return minv;

}

void Deletedata(int x)

{

root = Delete(root,x);

}

public Node search(Node root, int x)

{

if(root == null || root.data == x)

{

return root;

}

if(root.data < x)

{

return search(root.r,x);

}

return search(root.l,x);

}

public static void main(String[] args)

{

BST B = new BST();

B.Taocay();

System.***out***.println("Duyet Pre Order: ");

B.PreOrder(B.root);

System.***out***.println("\nDuyet In Order: ");

B.InOrder(B.root);

System.***out***.println("\nDuyet Post Order: ");

B.PostOrder(B.root);

System.***out***.println("\n Xoa: ");

B.Deletedata(8);

B.PostOrder(B.root);

}

}