**package** dal;

**public** **class** TreeNode{

**int** data;

TreeNode leftNode;

TreeNode rightNode;

**public** TreeNode()

{

}

**public** TreeNode(**int** d)

{

data = d;

}

**public** TreeNode(TreeNode left,TreeNode right,**int** d)

{

leftNode=left;

rightNode=right;

data=d;

}

}

**package** dal;

**import** java.util.LinkedList;

**import** java.util.Queue;

**import** java.util.Stack;

**public** **class** dal2 {

**public** **static** **void** main(String[] args) {

TreeNode head = **new** TreeNode(1);

TreeNode second = **new** TreeNode(2);

TreeNode three = **new** TreeNode(3);

TreeNode four = **new** TreeNode(4);

TreeNode five = **new** TreeNode(5);

TreeNode six = **new** TreeNode(6);

TreeNode seven = **new** TreeNode(7);

head.rightNode=three;

head.leftNode=second;

second.rightNode=five;

second.leftNode=four;

three.rightNode=seven;

three.leftNode=six;

System.***out***.print("广度优先:");

**new** dal2().BroadFirstSearch(head);

System.***out***.println();

System.***out***.print("深度优先");

**new** dal2().depthFirstSearch(head);

}

**public** **void** BroadFirstSearch(TreeNode nodeHead) {

**if**(nodeHead==**null**) {

**return**;

}

Queue<TreeNode> myQueue = **new** LinkedList<>();

myQueue.add(nodeHead);

**while**(!myQueue.isEmpty()) {

TreeNode node= myQueue.poll();

System.***out***.print(node.data+" ");

**if**(**null** != node.leftNode) {

myQueue.add(node.leftNode);

}

**if**(**null** != node.rightNode) {

myQueue.add(node.rightNode);

}

}

}

**public** **void** depthFirstSearch(TreeNode nodeHead) {

**if**(nodeHead==**null**) {

**return**;

}

Stack<TreeNode> myStack=**new** Stack<>();

myStack.add(nodeHead);

**while**(!myStack.isEmpty()) {

TreeNode node= myStack.pop();

System.***out***.print(node.data+" ");

**if**(node.rightNode!=**null**) {

myStack.push(node.rightNode);

}

**if**(node.leftNode!=**null**) {

myStack.push(node.leftNode);

}

}

}

}

