```
package Iterators;

import java.util.Stack;
/*
class TreeNode {
    int val;
    TreeNode left, right;
    public TreeNode(int v) {
        val = v;
    }
}
*/
public class PostorderIterator {
    Stack<TreeNode> stack;
    public PostorderIterator(TreeNode root) {
        stack = new Stack<>();
        findNextLeaf(root);
    }

    public void findNextLeaf(TreeNode cur) {
        while (cur != null) {
            stack.push(cur);
            if (cur.left != null) {
                cur = cur.left;
            } else {
                cur = cur.right;
            }
        }
    }

    public boolean hasNext() {
        return !stack.isEmpty();
    }

    public int next() {
        if (!hasNext()) return -1;
        TreeNode res = stack.pop();

        if (!stack.isEmpty()) {
            TreeNode top = stack.peek();
            if (res == top.left) {
                findNextLeaf(top.right);
            }
        }

        return res.val;
    }

    public static void main(String[] args) {
        TreeNode n1 = new TreeNode(1);
        TreeNode n2 = new TreeNode(2);
        TreeNode n3 = new TreeNode(3);
        TreeNode n4 = new TreeNode(4);
        n3.left = n2;
        n2.left = n1;
        n3.right = n4;
        PostorderIterator itr = new PostorderIterator(n3);
        while (itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}
```