模版·二叉树的实现

```
class Node{
    int val;
    Node left;
    Node right;

    public Node(int val){
        this.val=val;
        this.left=null;
        this.right=null;
    }
}

class BinaryTree{
    private Node root;

    public BinaryTree(){
        this.root=null;
    }

    //插入操作：插入一个新节点
    public void insert(int val){
        Node newNode=new Node(val);
        if(root==null){
            root=newNode;
            return;
        }

        Node curr=root;
        while(true){
            if(val<curr.val){
```

```
                if(curr.left==null){
                    curr.left=newNode;
                    return;
                }
            curr=curr.left;
            }else{
                if(curr.right==null){
                    curr.right=newNode;
                    return;
                }
                curr=curr.right;
            }
        }
    }

    public Node find(int val){
        Node curr=root;
        while(curr!=null){
            if(val==curr.val){
                return curr;
            }else if(val<curr.val){
                curr=curr.left;
            }else{
                curr=curr.right;
            }
        }
        return null;
    }

    //前序遍历：先访问根节点，再遍历左子树，再遍历右子树
    public void preOrder(Node node){
        if(node==null){
            return;
```

```
        }
        System.out.print(node.val+"");
        preOrder(node.left);
        preOrder(node.right);
    }

    //中序遍历：先遍历左子树，再访问根节点，再遍历右子树
    public void inOrder(Node node){
        if(node==null){
            return;
        }
        inOrder(node.left);
        System.out.print(node.val+"");
        inOrder(node.right);
    }

    //后序遍历：先遍历左子树，再遍历右子树，再访问根节点
    public void postOrder(Node node){
        if(node==null){
            return;
        }
        postOrder(node.left);
        postOrder(node.right);
        System.out.print(node.val+"");
    }
}
```