

Binary Tree Traversals in Java

Java Code: Binary Tree Traversals with Maximum Depth

```
class Node {
    int data;
    Node left, right;

    public Node(int item) {
        data = item;
        left = right = null;
    }
}

public class BinaryTree {
    Node root;

    void inorder(Node node) {
        if (node == null)
            return;
        inorder(node.left);
        System.out.print(node.data + " ");
        inorder(node.right);
    }

    void preorder(Node node) {
        if (node == null)
            return;
        System.out.print(node.data + " ");
        preorder(node.left);
        preorder(node.right);
    }

    void postorder(Node node) {
        if (node == null)
            return;
        postorder(node.left);
        postorder(node.right);
        System.out.print(node.data + " ");
    }

    int maxDepth(Node node) {
        if (node == null)
            return 0;
        int leftDepth = maxDepth(node.left);
        int rightDepth = maxDepth(node.right);
        return Math.max(leftDepth, rightDepth) + 1;
    }

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

Binary Tree Traversals in Java

```
BinaryTree tree = new BinaryTree();
tree.root = new Node(1);
tree.root.left = new Node(2);
tree.root.right = new Node(3);
tree.root.left.left = new Node(4);
tree.root.left.right = new Node(5);

System.out.print("Inorder traversal: ");
tree.inorder(tree.root);
System.out.println();

System.out.print("Preorder traversal: ");
tree.preorder(tree.root);
System.out.println();

System.out.print("Postorder traversal: ");
tree.postorder(tree.root);
System.out.println();

System.out.println("Maximum depth of the tree: " + tree.maxDepth(tree.root));
}
```