Q- WAP in java to define treeNode & construct a tree and perform inorder, preorder, postorder operation in that tree

Java code

```
class TreeNode {
   int data;
   TreeNode left;
   TreeNode right;

   TreeNode(int value) {
      data = value;
      left = right = null;
   }
}
public class BinaryTree {
   TreeNode root;
   void inorder(TreeNode node) {
      if (node == null) return;
      inorder(node.left);
      System.out.print(node.data + " ");
      inorder(node.right);
   }
   void preorder(TreeNode node) {
      if (node == null) return;
      System.out.print(node.data + " ");
      preorder(node.left);
      preorder(node.right);
   }
   void postorder(TreeNode node) {
      if (node == null) return;
      postorder(node.left);
```

```
        postorder(node.right);
        System.out.print(node.data + " ");
    }
    public static void main(String[] args) {
        BinaryTree tree = new BinaryTree();
        tree.root = new TreeNode(1);
        tree.root.left = new TreeNode(2);
        tree.root.right = new TreeNode(3);
        tree.root.left.left = new TreeNode(4);
        tree.root.left.right = new TreeNode(5);
        tree.root.right.right = new TreeNode(6);

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

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

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

Q2-Maximum depth of Binary tree.

Java code

```
class TreeNode {
    int data;
    TreeNode left;
```

```
    TreeNode right;

    TreeNode(int value) {
        this.data = value;
        this.left = null;
        this.right = null;
    }
}

public class BinaryTreeDepth {
    TreeNode root;

    int maxDepth(TreeNode 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) {
        BinaryTreeDepth tree = new BinaryTreeDepth();
        tree.root = new TreeNode(1);
        tree.root.left = new TreeNode(2);
        tree.root.right = new TreeNode(3);
        tree.root.left.left = new TreeNode(4);
        tree.root.left.right = new TreeNode(5);

        int depth = tree.maxDepth(tree.root);
        System.out.println("Maximum Depth of the Binary Tree: " + depth);
```

```
    }
}
```