```
import java.util.Scanner;

class Node {
	int key;
	Node left, right;

	Node(int e) {
		key = e;
		left = right = null;
	}
}

class Tree {
	Node root;

	Tree() {
		root = null;
	}

	public void inSert(Node r, Node n) {
		if (root == null)
		{
			root = n;
		}
		else
		{
			if (n.key < r.key)
			{
				if (r.left == null)
```

```
                {
                    r.left = n;
                }
                else
                {
                    inSert(r.left, n);
                }
            }
            else
            {
                if (r.right == null)
                {
                    r.right = n;
                }
                else
                {
                    inSert(r.right, n);
                }
            }
        }
}

public void inorder(Node r) {
        if (r != null) {
            inorder(r.left);
            System.out.println(r.key);
            inorder(r.right);
        }
}
```

```
	public void preorder(Node r) {
		if (r != null) {
			System.out.println(r.key);
			inorder(r.left);
			inorder(r.right);
		}
	}

	public void postorder(Node r) {
		if (r != null) {


			inorder(r.left);
			inorder(r.right);
			System.out.println(r.key);
		}
	}
}

public class DSTree {
	public static void main(String args[]) {


		int ch;
		Tree t = new Tree();
		Scanner scanner = new Scanner(System.in);

		for(int i=0;i<=5;i++)
		{
			System.out.println("Insert Tree");
```

```
    int in = scanner.nextInt();
    Node n = new Node(in);
    t.inSert(t.root, n);

}
do {
         System.out.println(
         "\n1.Inorder \n2.Preorder \n3.Postorder \n Enter choice :");
        ch = scanner.nextInt();
        switch (ch) {

                case 1:
                System.out.println("Inorder");
                t.inorder(t.root);
                break;
                case 2:
                System.out.println("Preorder");
                t.preorder(t.root);
                break;
          case 3:
                System.out.println("Postorder");
                t.postorder(t.root);
                break;

        }
} while (ch != 0);

}
```

```
}
```