SOFTWARE TESTING

THE ROLE OF A TESTER IS TO BREAK THE SOFTWARE

Prepared For :
SIR MUKHTAIR ZAMIN

Prepared By :

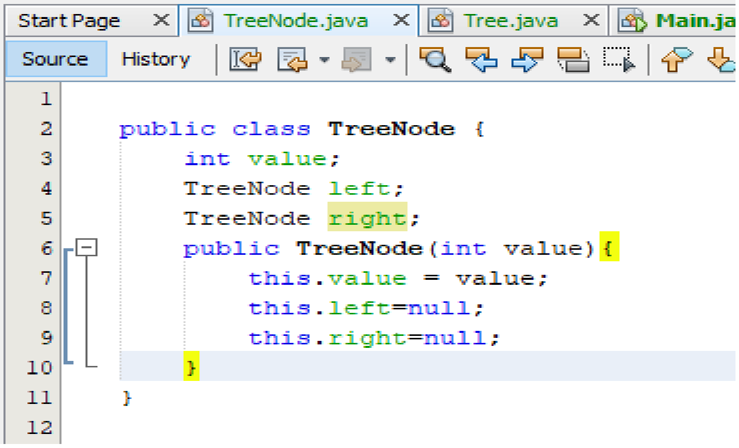
BASIT IQBAL
FA21-BSE-050

Question:

Implement the Inorder Traversal of the Tree....?

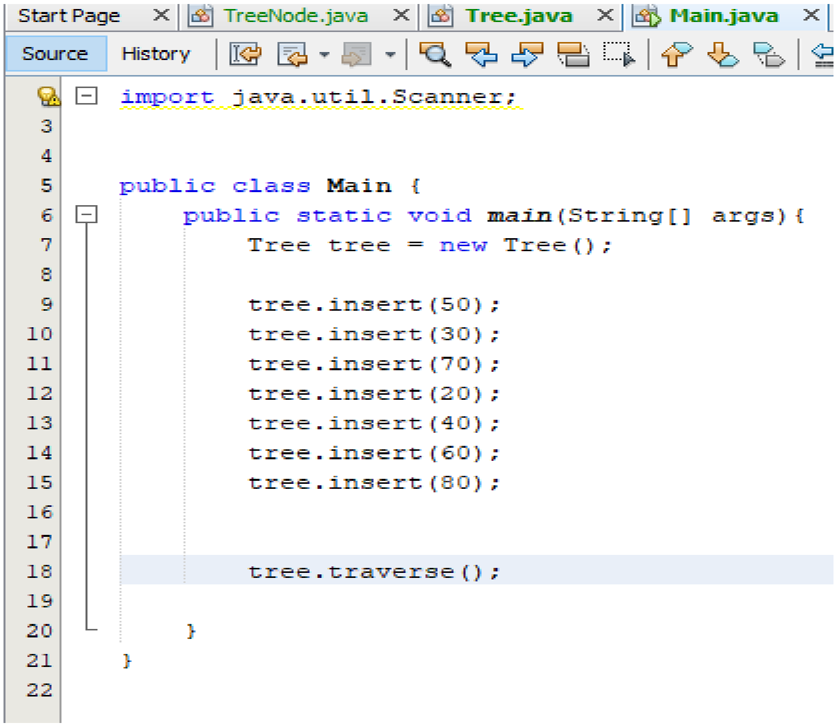
Answer:

TreeNode.java



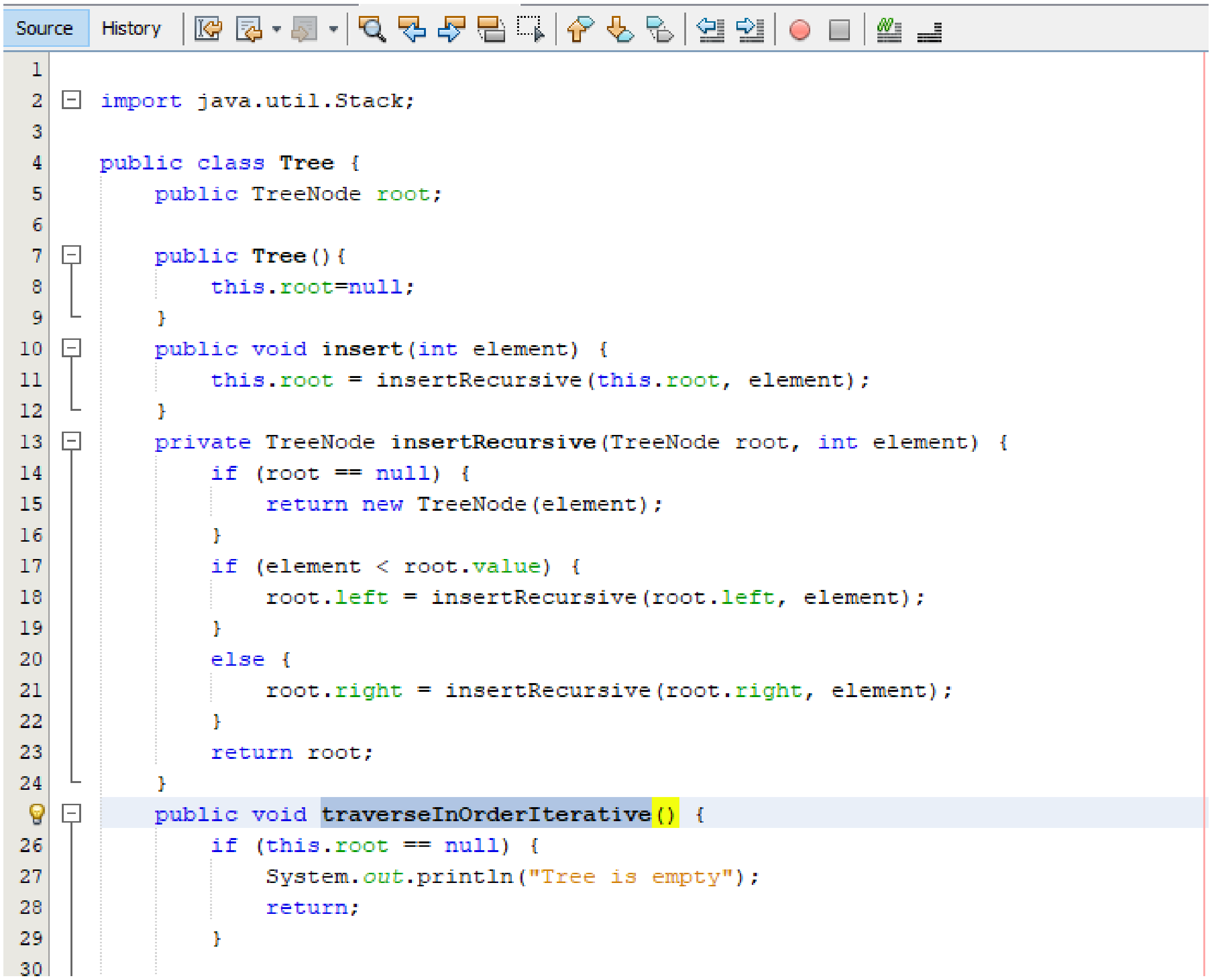
```java

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

```

Main.java

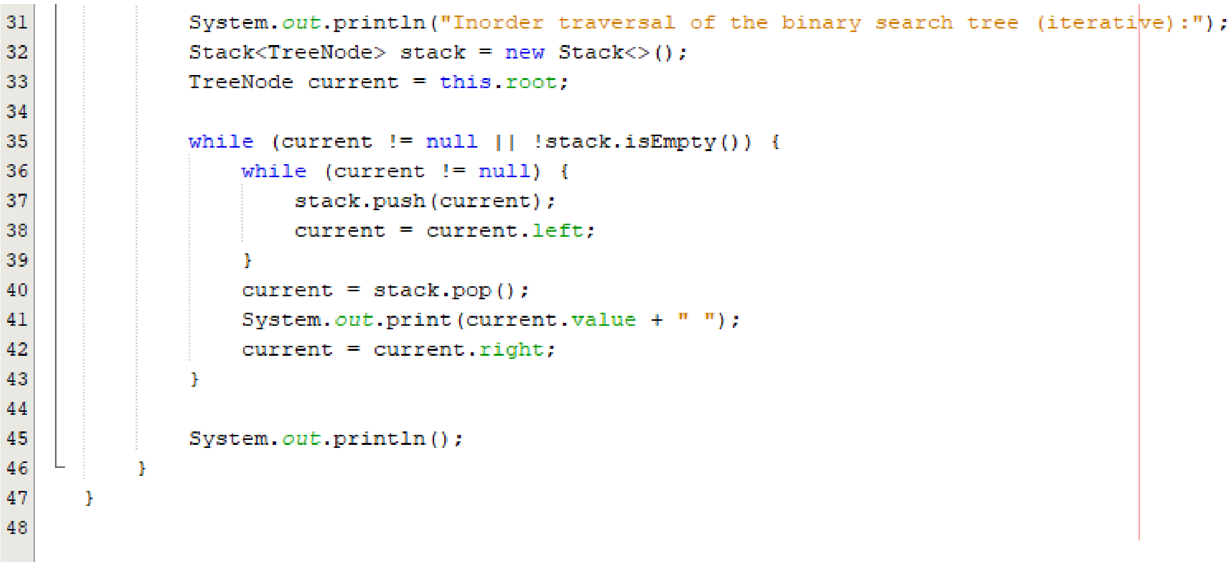


```java
import java.util.Scanner;


public class Main {
    public static void main(String[] args){
        Tree tree = new Tree();

        tree.insert(50);
        tree.insert(30);
        tree.insert(70);
        tree.insert(20);
        tree.insert(40);
        tree.insert(60);
        tree.insert(80);


        tree.traverse();

    }
}

```

Tree.java:

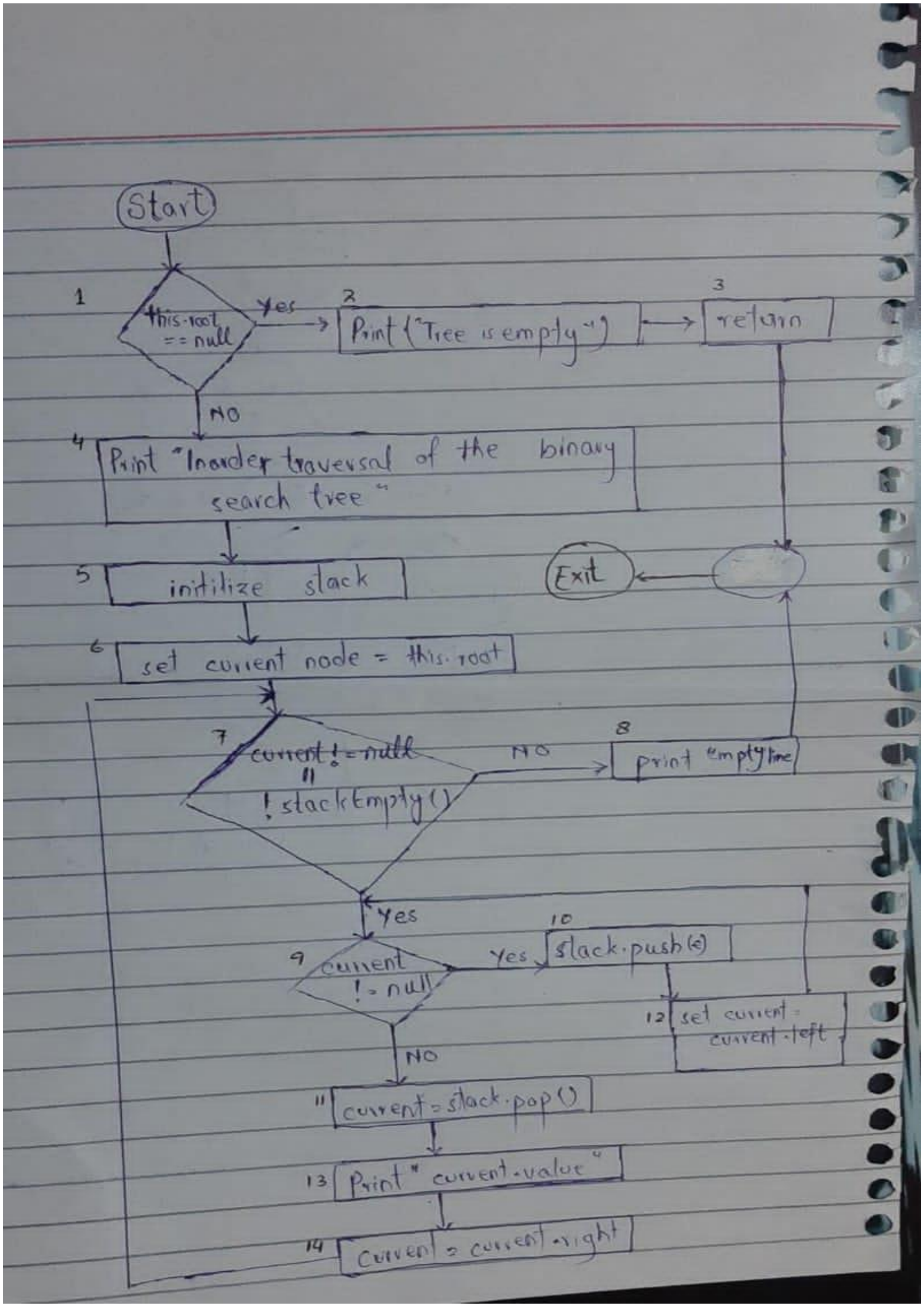


```java

import java.util.Stack;

public class Tree {
    public TreeNode root;

    public Tree(){
        this.root=null;
    }
    public void insert(int element) {
        this.root = insertRecursive(this.root, element);
    }
    private TreeNode insertRecursive(TreeNode root, int element) {
        if (root == null) {
            return new TreeNode(element);
        }
        if (element < root.value) {
            root.left = insertRecursive(root.left, element);
        }
        else {
            root.right = insertRecursive(root.right, element);
        }
        return root;
    }
    public void traverseInOrderIterative() {
        if (this.root == null) {
            System.out.println("Tree is empty");
            return;
        }

```



```java
        System.out.println("Inorder traversal of the binary search tree (iterative):");
        Stack<TreeNode> stack = new Stack<>();
        TreeNode current = this.root;

        while (current != null || !stack.isEmpty()) {
            while (current != null) {
                stack.push(current);
                current = current.left;
            }
            current = stack.pop();
            System.out.print(current.value + " ");
            current = current.right;
        }

        System.out.println();
    }
}

```

Control Flow Graph:



Paths:

Path 1 → 1, 2, 3.

Path 2 → 1, 4, 5, 6, 7, 8.

Path 3 → 1,4,5,6,7,9,10,12,9,11,13,14,7,8.

Path 4 → 1,4,5,6,7,9,11,13,14,7,8.

Test Cases:

Test Case ID	Description	Input Data	Expected Output	Actual Output	Status
TC_001	The root of the tree is “null” (i.e the tree is empty)		Display Msg: “Tree is Empty”	Display Msg: “Tree is empty”	Pass
TC_002	The stack has some element (i.e node) in it.		Display Msg: “Inorder traversal of the binary search tree (iterative):” and an empty line is printed.		
TC_003	The tree has some elements in it. Now when this method will be called the elements will be printed.		Display Msg: “Inorder traversal of the binary search tree (iterative):” and all the elements of the tree.	Display Msg: “Inorder traversal of the binary search tree (iterative):” and all the elements of the tree.	Pass
TC_004	Current is equal to null but the stack is not empty and has some element..		Display Msg: “Inorder traversal of the binary search tree (iterative):” and one element that	Display Msg: “Inorder traversal of the binary search tree (iterative):” and one element that	Pass

			was in stack is printed.	was in stack is printed.	