3주차 실습과제

20135151 이갑성

**public** **class** TreeNode {

**int** data; //데이터 값

TreeNode left; //왼쪽 노드

TreeNode right; //오른쪽 노드

**public** TreeNode(**int** data){ //생성자

**this**.data = data;

**this**.left=**null**;

**this**.right=**null**;

}

}

**public** **class** BinaryTree {

**int** count;// 노드 개수

**int** level;// 레벨

**public** BinaryTree() { // 생성자 생성

**this**.count = 0; // count 초기화

**this**.level = 0; // 레벨 초기화

}

**public** TreeNode makenode(TreeNode left, **int** data, TreeNode right) { // 노드를

// 만드는

// 메소드

TreeNode root = **new** TreeNode(data); // root객체생성

root.left = left;

root.right = right;

count++; // 생성할때마다 count의 개수를 증가시켜 총 노드의 개수를 알고싶을때 알 수 있음

**return** root;

}

**public** **void** preorder(TreeNode root) { // 전위순회 메소드

**if** (root != **null**) {

System.***out***.print(root.data + ", "); // 찍기

preorder(root.left); // 왼쪽

preorder(root.right); // 오른쪽

}

}

**public** **void** postorder(TreeNode root) { // 후위순회 메소드

**if** (root != **null**) {

postorder(root.left); // 왼쪽

postorder(root.right); // 오른쪽

System.***out***.print(root.data + ", "); // 찍기

}

}

**public** **void** inorder(TreeNode root) {

**if** (root != **null**) {

inorder(root.left); // 왼쪽

System.***out***.print(root.data + ", "); // 찍고

inorder(root.right); // 오른쪽

}

}

**public** **int** count() { // 트리의 노드 총 개수를 알 수 있는 메소드

**return** count;

}

**public** **int** level(TreeNode root) { // 트리의 레벨을 알 수 있는 메소드

**int** a = 0, b = 0;

**if** (root == **null**) {

**return** 0;

} **else** {

a = level(root.left) + 1;

b = level(root.right) + 1;

**if** (a > b)

**return** a;

**else**

**return** b;

}

/\*

\* if (level(root.left) > level(root.right)) {

\*

\* return level(root.left) + 1; } else { return level(root.right) + 1; }

\*/

}

}

**public** **class** Main {

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

BinaryTree A = **new** BinaryTree();

TreeNode node1 = A.makenode(**null**,1,**null**);

TreeNode node2 = A.makenode(**null**,7,**null**);

TreeNode node3 = A.makenode(node1,3,node2);

TreeNode node4 = A.makenode(**null**,14,**null**);

TreeNode node5 = A.makenode(node3,9,node4);

TreeNode node6 = A.makenode(**null**,44,**null**);

TreeNode node7 = A.makenode(**null**,36,node6);

TreeNode node8 = A.makenode(**null**,97,**null**);

TreeNode node9 = A.makenode(**null**,94,node8);

TreeNode node10 = A.makenode(node7,67,node9);

TreeNode node11 = A.makenode(node5,22,node10);

System.***out***.println("Preorder");

A.preorder(node11);

System.***out***.println("\n");

System.***out***.println("Postorder");

A.postorder(node11);

System.***out***.println("\n");

System.***out***.println("Inorder");

A.inorder(node11);

System.***out***.println("\n");

System.***out***.println("노드의 개수: " + A.count() + "개 입니다.");

System.***out***.println("Level: " + A.level(node11));

}

}

