**Write a Java program to find maximum width of a binary tree.**

**class** Node

{

**int** data;

Node left, right;

Node(**int** item)

{

data = item;

left = right = **null**;

}

}

**class** BinaryTree

{

Node root;

**int** getMaxWidth(Node node)

{

**int** maxWidth = 0;

**int** width;

**int** h = height(node);

**int** i;

**for** (i = 1; i <= h; i++)

{

width = getWidth(node, i);

**if** (width > maxWidth)

maxWidth = width;

}

**return** maxWidth;

}

**int** getWidth(Node node, **int** level)

{

**if** (node == **null**)

**return** 0;

**if** (level == 1)

**return** 1;

**else** **if** (level > 1)

**return** getWidth(node.left, level - 1)

+ getWidth(node.right, level - 1);

**return** 0;

}

**int** height(Node node)

{

**if** (node == **null**)

**return** 0;

**else**

{

**int** lHeight = height(node.left);

**int** rHeight = height(node.right);

**return** (lHeight > rHeight) ? (lHeight + 1) : (rHeight + 1);

}

}

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

{

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);

tree.root.right.right = **new** Node(18);

tree.root.right.right.left = **new** Node(9);

tree.root.right.right.right = **new** Node(7);

System.***out***.println("Maximum width is: " + tree.getMaxWidth(tree.root));

}

}

**Output:**

