Java program to find the total number of possible Binary Search Trees with N keys

1. **public** **class** BinarySearchTree {
3. //Represent the node of binary tree
4. **public** **static** **class** Node{
5. **int** data;
6. Node left;
7. Node right;
9. **public** Node(**int** data){
10. //Assign data to the new node, set left and right children to null
11. **this**.data = data;
12. **this**.left = **null**;
13. **this**.right = **null**;
14. }
15. }
17. //Represent the root of binary tree
18. **public** Node root;
20. **public** BinarySearchTree(){
21. root = **null**;
22. }
24. //factorial() will calculate the factorial of given number
25. **public** **int** factorial(**int** num) {
26. **int** fact = 1;
27. **if**(num == 0)
28. **return** 1;
29. **else** {
30. **while**(num > 1) {
31. fact = fact \* num;
32. num--;
33. }
34. **return** fact;
35. }
36. }
38. //numOfBST() will calculate the total number of possible BST by calculating Catalan Number for given key
39. **public** **int** numOfBST(**int** key) {
40. **int** catalanNumber = factorial(2 \* key)/(factorial(key + 1) \* factorial(key));
41. **return** catalanNumber;
42. }
44. **public** **static** **void** main(String[] args) {
46. BinarySearchTree bt = **new** BinarySearchTree();
48. //Display total number of possible binary search tree with key 5
49. System.out.println("Total number of possible Binary Search Trees with given key: " + bt.numOfBST(5));
50. }
51. }