Design a java code to calculate the difference between the sum of the odd level and even level nodes of a binary tree

import java.util.LinkedList;

import java.util.Queue;

public class evenoddtree {

//Represent a node of binary tree

public static class Node{

int data;

Node left;

Node right;

public Node(int data){

//Assign data to the new node, set left and right children to null

this.data = data;

this.left = null;

this.right = null;

}

}

//Represent the root of binary tree

public Node root;

public evenoddtree(){

root = null;

}

public int difference() {

int oddLevel = 0, evenLevel = 0, diffOddEven = 0;

int nodesInLevel = 0;

int currentLevel = 0;

Queue<Node> queue = new LinkedList<Node>();

if(root == null) {

System.out.println("Tree is empty");

return 0;

}

else {

queue.add(root);

currentLevel++;

while(queue.size() != 0) {

nodesInLevel = queue.size();

while(nodesInLevel > 0) {

Node current = queue.remove();

if(currentLevel % 2 == 0)

evenLevel += current.data;

else

oddLevel += current.data;

if(current.left != null)

queue.add(current.left);

if(current.right != null)

queue.add(current.right);

nodesInLevel--;

}

currentLevel++;

}

diffOddEven = Math.abs(oddLevel - evenLevel);

}

return diffOddEven;

}

public static void main (String[] args) {

evenoddtree bt = new evenoddtree();

bt.root = new Node(1);

bt.root.left = new Node(3);

bt.root.right = new Node(5);

bt.root.left.left = new Node(4);

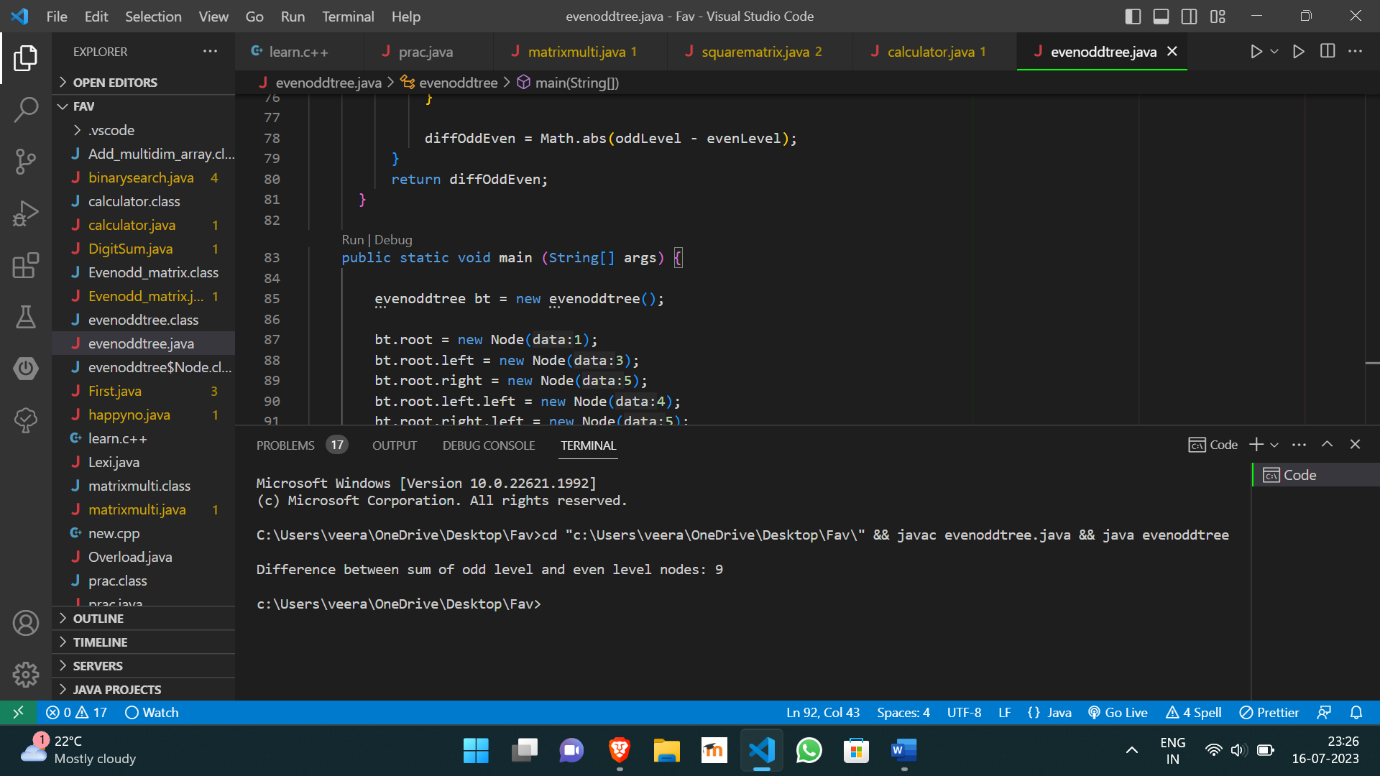
bt.root.right.left = new Node(5);

bt.root.right.right = new Node(7);

System.out.println("Difference between sum of odd level and even level nodes: " + bt.difference());

}

}



https://github.com/Veeragoutham04/Java\_Lab/blob/main/Binary%20tree%20(even%20and%20odd%20level)