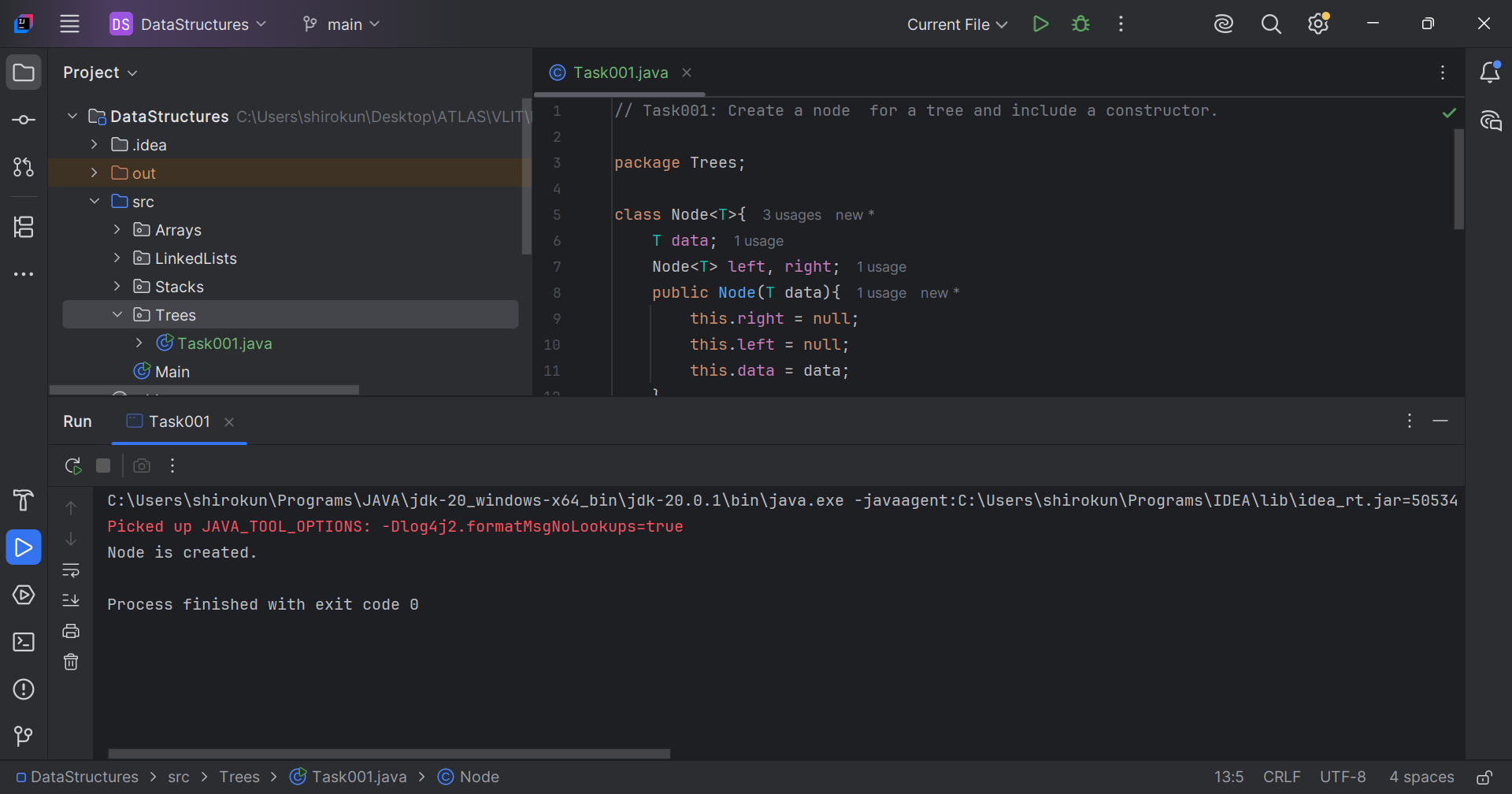
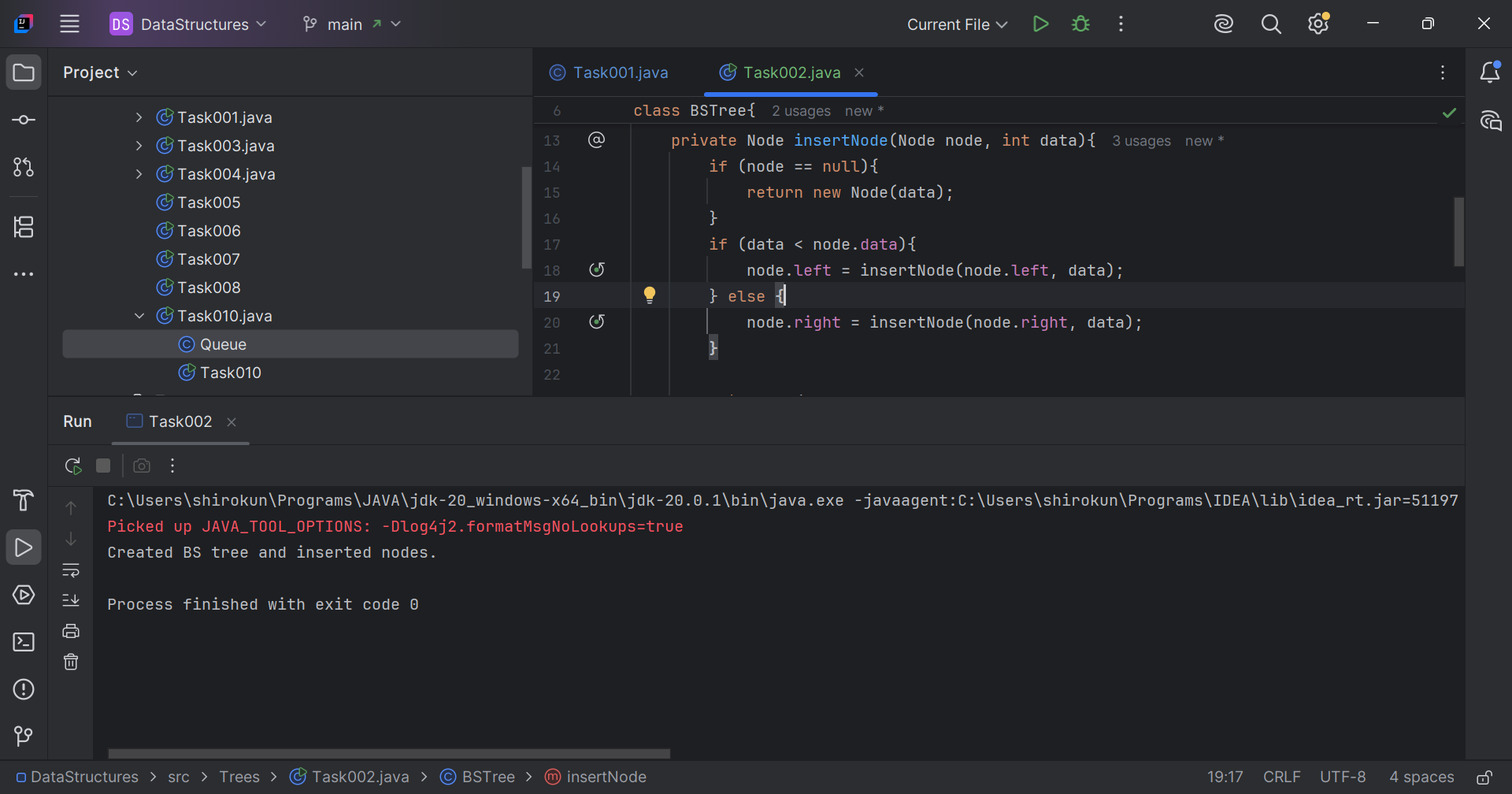
Day 15 – 05/07/2025

// Task001: Create a node for a tree and include a constructor.  
  
package Trees;  
  
class Node<T>{  
 T data;  
 Node<T> left, right;  
 public Node(T data){  
 this.right = null;  
 this.left = null;  
 this.data = data;  
 }  
  
 void display(){  
 System.*out*.println("Node is created.");  
 }  
}  
  
public class Task001 {  
 public static void main(String[] args) {  
 Node<String> node = new Node<>(" ");  
 node.display();  
 }  
}



// Task002: Create a class named Binary Search tree  
// in which you have 2 insert operations  
  
package Trees;  
  
class BSTree{  
 Node root;  
  
 public void insert(int data){  
 root = insertNode(root, data);  
 }  
  
 private Node insertNode(Node node, int data){  
 if (node == null){  
 return new Node(data);  
 }  
 if (data < node.data){  
 node.left = insertNode(node.left, data);  
 } else {  
 node.right = insertNode(node.right, data);  
 }  
 return node;  
 }  
  
}  
  
public class Task002 {  
 public static void main(String[] args) {  
 BSTree bsTree = new BSTree();  
 int[] arr = {5, 1, 6, 4, 3, 2, 8, 7};  
 for (int a:arr){ bsTree.insert(a);}  
 System.*out*.println("Created BS tree and inserted nodes.");  
 }  
}



// Task003: Inorder traversal.  
  
package Trees;  
  
class Traversal extends BSTree {  
  
 void inorderTraversal() {  
 printInorder(root);  
 }  
  
 private void printInorder(Node node) {  
 if (node != null) {  
 printInorder(node.left);  
 System.*out*.print(node.data + " ");  
 printInorder(node.right);  
 }  
 }  
}  
  
public class Task003 {  
 public static void main(String[] args) {  
 Traversal traversal = new Traversal();  
 int[] arr = {5, 1, 6, 4, 3, 2, 8, 7};  
 for (int a:arr){  
 traversal.insert(a);  
 }  
 traversal.inorderTraversal();  
 }  
}

