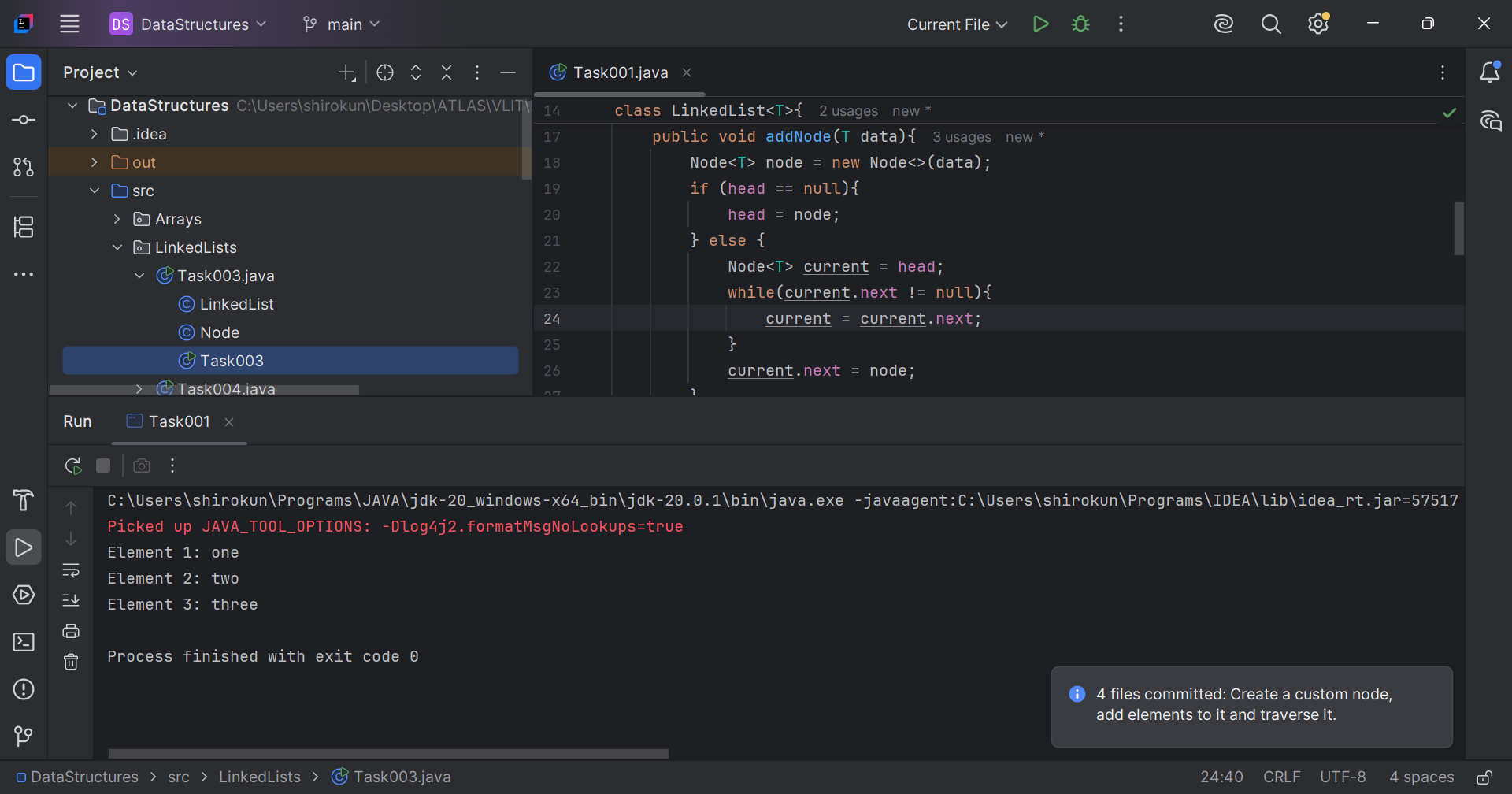
Day 14 – 04/07/2025

// Task001: Create a custom node, add elements to it and traverse it.  
  
package Stacks;  
  
class Node<T>{  
 T data;  
 Node<T> next;  
 public Node(T data){  
 this.data = data;  
 this.next = null;  
 }  
}  
  
class LinkedList<T>{  
 Node<T> head;  
  
 public void addNode(T data){  
 Node<T> node = new Node<>(data);  
 if (head == null){  
 head = node;  
 } else {  
 Node<T> current = head;  
 while(current.next != null){  
 current = current.next;  
 }  
 current.next = node;  
 }  
 }  
  
 public void traverse(){  
 int index = 1;  
 Node<T> current = head;  
 while (current != null){  
 System.*out*.println("Element "+index+": "+current.data);  
 current = current.next;  
 index++;  
 }  
 }  
}  
  
public class Task001 {  
 public static void main(String[] args) {  
 LinkedList<String> list = new LinkedList<>();  
 list.addNode("one");  
 list.addNode("two");  
 list.addNode("three");  
 list.traverse();  
 }  
}



Q2. What do you understand by traversing a linked list?  
Ans. Traverse means travel. So, in a linked list traverse means travelling across the linked list element by element. Traversing helps us to search, delete or print elements.