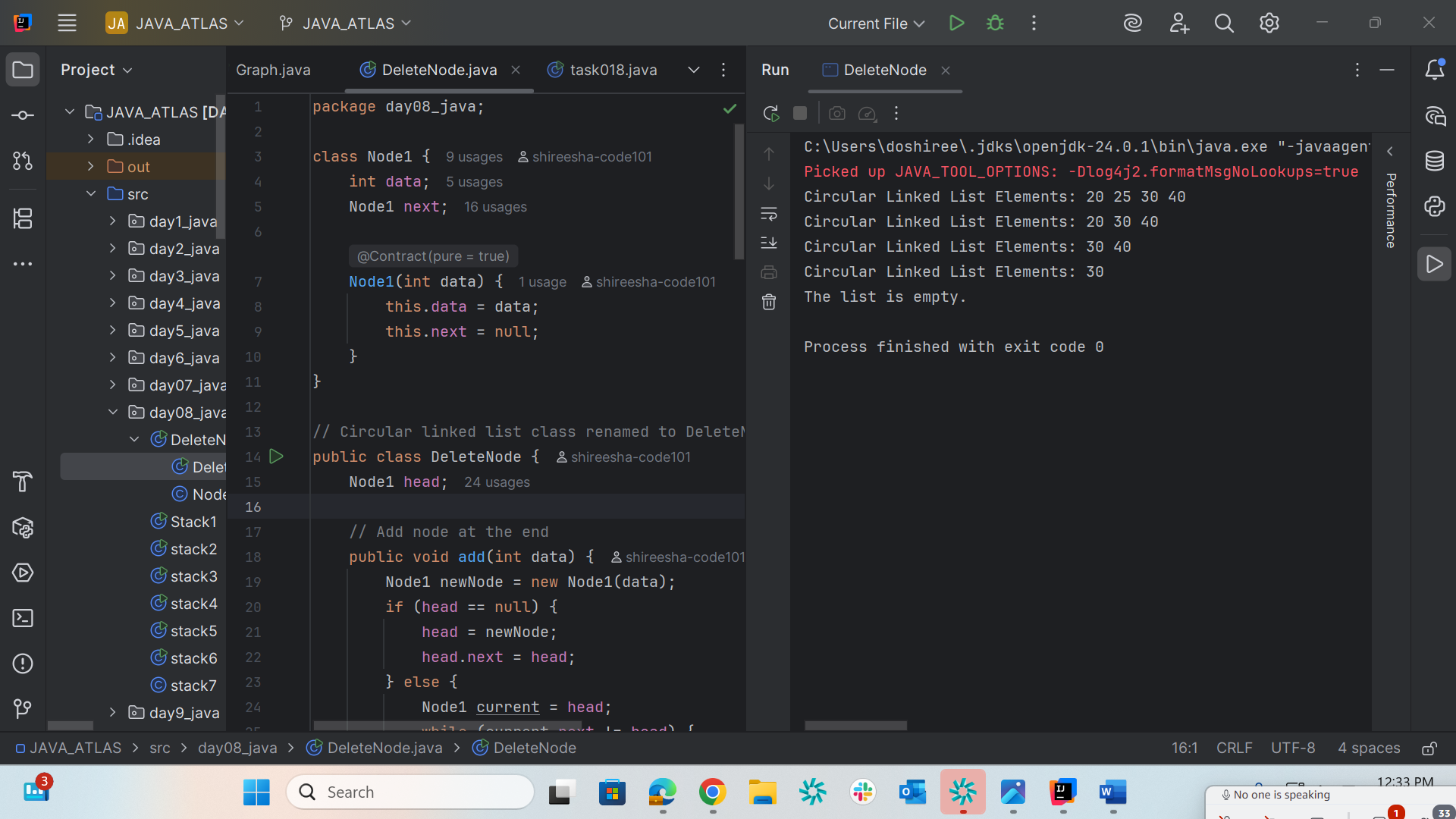
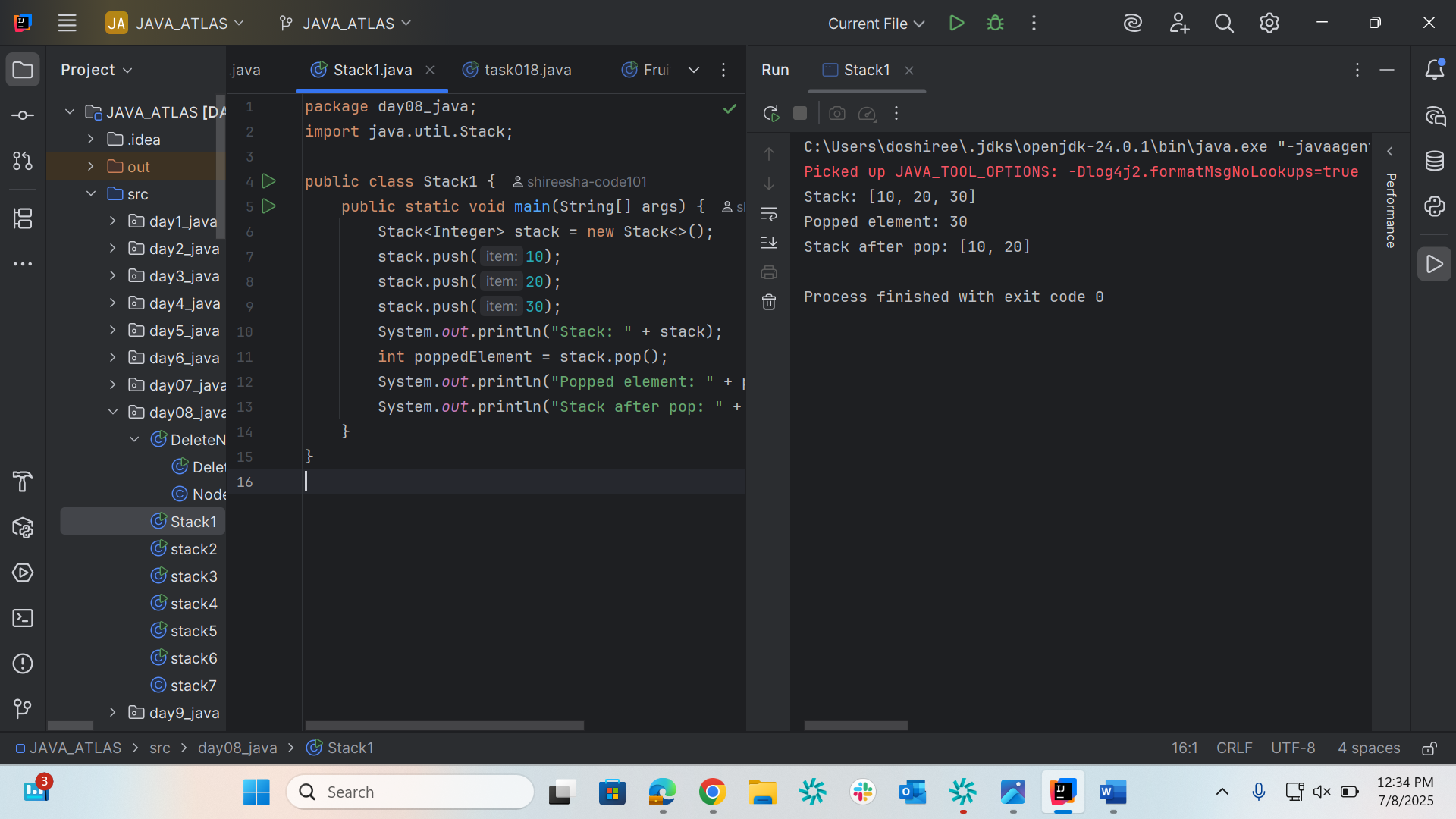
Task001

package day08\_java;  
  
class Node1 {  
 int data;  
 Node1 next;  
  
 Node1(int data) {  
 this.data = data;  
 this.next = null;  
 }  
}  
  
// Circular linked list class renamed to DeleteNode  
public class DeleteNode {  
 Node1 head;  
  
 // Add node at the end  
 public void add(int data) {  
 Node1 newNode = new Node1(data);  
 if (head == null) {  
 head = newNode;  
 head.next = head;  
 } else {  
 Node1 current = head;  
 while (current.next != head) {  
 current = current.next;  
 }  
 current.next = newNode;  
 newNode.next = head;  
 }  
 }  
  
 // Display list elements  
 public void display() {  
 if (head == null) {  
 System.*out*.println("The list is empty.");  
 return;  
 }  
 Node1 current = head;  
 System.*out*.print("Circular Linked List Elements: ");  
 do {  
 System.*out*.print(current.data + " ");  
 current = current.next;  
 } while (current != head);  
 System.*out*.println();  
 }  
  
 // Delete first occurrence of key  
 public void delete(int key) {  
 if (head == null) return;  
  
 // Only one node  
 if (head.next == head && head.data == key) {  
 head = null;  
 return;  
 }  
  
 // If head node needs deletion  
 if (head.data == key) {  
 Node1 last = head;  
 while (last.next != head) {  
 last = last.next;  
 }  
 head = head.next;  
 last.next = head;  
 return;  
 }  
  
 // Deleting other nodes  
 Node1 prev = head;  
 Node1 curr = head.next;  
 while (curr != head) {  
 if (curr.data == key) {  
 prev.next = curr.next;  
 return;  
 }  
 prev = curr;  
 curr = curr.next;  
 }  
 }  
  
 // Test the implementation  
 public static void main(String[] args) {  
 DeleteNode list = new DeleteNode();  
  
 list.add(20);  
 list.add(25);  
 list.add(30);  
 list.add(40);  
  
 list.display(); // Output: 20 25 30 40  
  
 list.delete(25);  
 list.display(); // Output: 20 30 40  
  
 list.delete(20); // Deleting head  
 list.display(); // Output: 30 40  
  
 list.delete(40); // Deleting last  
 list.display(); // Output: 30  
  
 list.delete(30); // Deleting sole node  
 list.display(); // Output: The list is empty.  
 }  
}



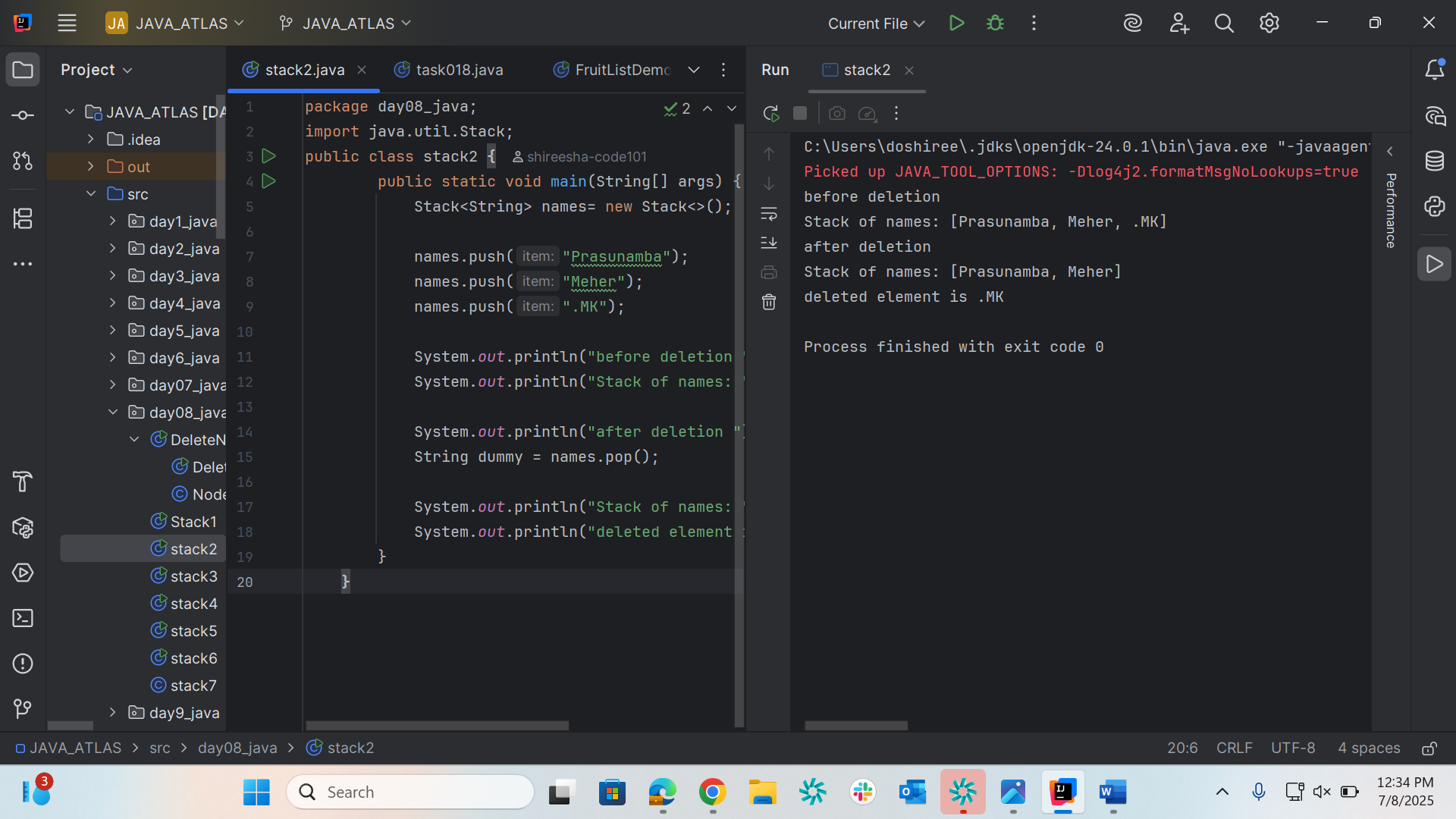
Task002

package day08\_java;  
import java.util.Stack;  
  
public class Stack1 {  
 public static void main(String[] args) {  
 Stack<Integer> stack = new Stack<>();  
 stack.push(10);  
 stack.push(20);  
 stack.push(30);  
 System.*out*.println("Stack: " + stack);  
 int poppedElement = stack.pop();  
 System.*out*.println("Popped element: " + poppedElement);  
 System.*out*.println("Stack after pop: " + stack);  
 }  
}



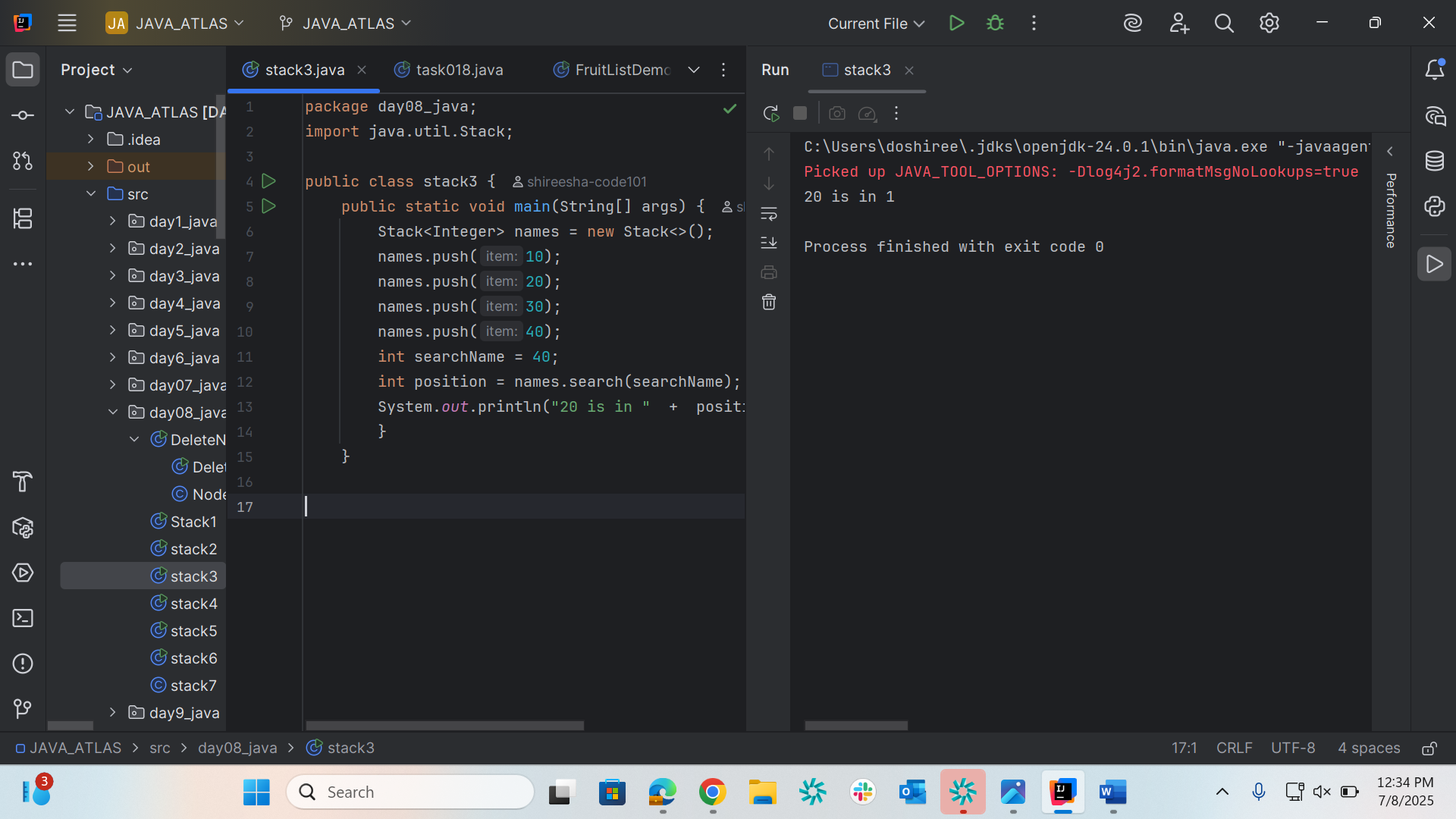
Task003

package day08\_java;  
import java.util.Stack;  
public class stack2 {  
 public static void main(String[] args) {  
 Stack<String> names= new Stack<>();  
  
 names.push("Prasunamba");  
 names.push("Meher");  
 names.push(".MK");  
  
 System.*out*.println("before deletion ");  
 System.*out*.println("Stack of names: " + names);  
  
 System.*out*.println("after deletion ");  
 String dummy = names.pop();  
  
 System.*out*.println("Stack of names: " + names);  
 System.*out*.println("deleted element is "+ dummy);  
 }  
 }



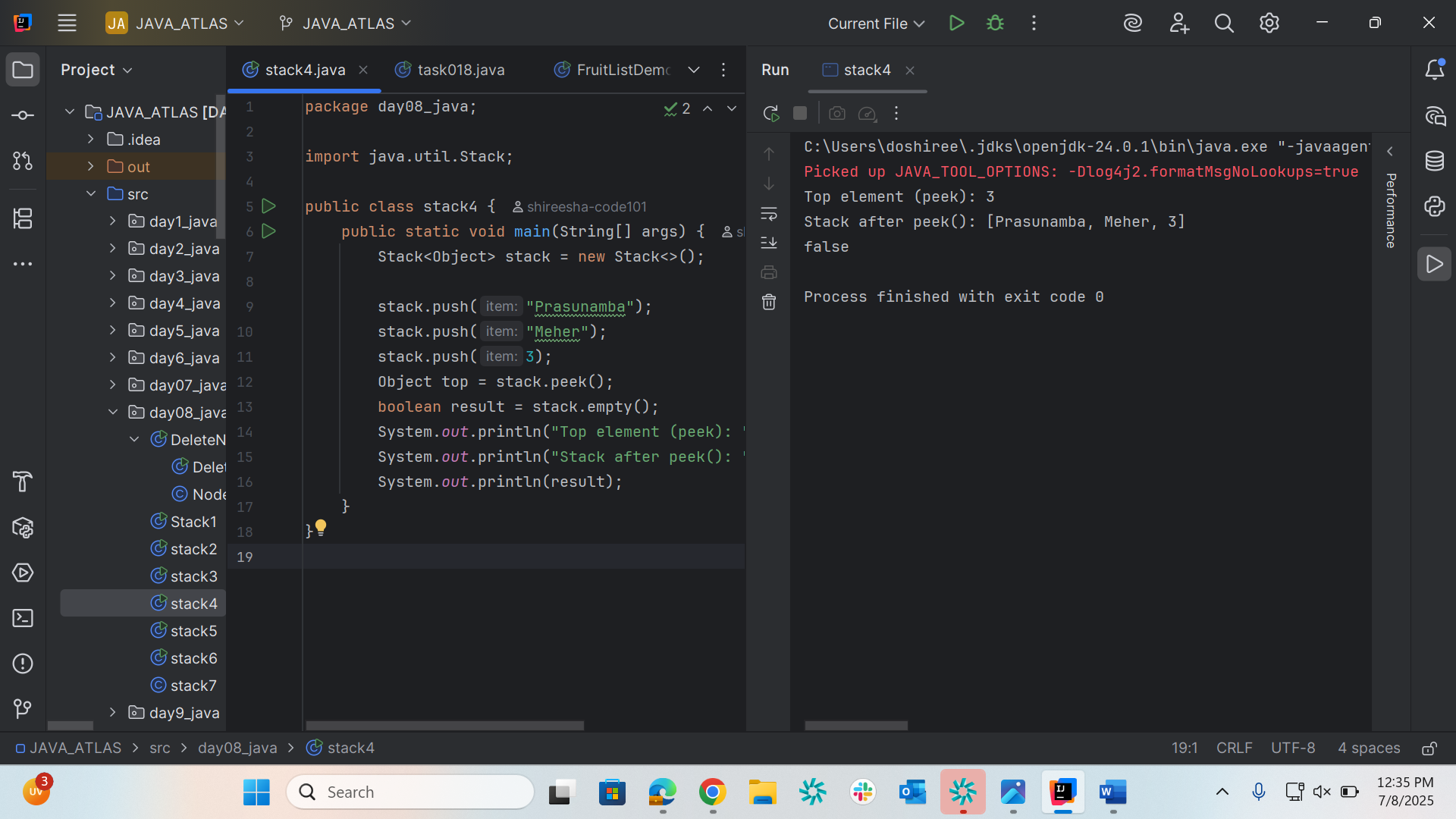
Task004

package day08\_java;  
import java.util.Stack;  
  
public class stack3 {  
 public static void main(String[] args) {  
 Stack<Integer> names = new Stack<>();  
 names.push(10);  
 names.push(20);  
 names.push(30);  
 names.push(40);  
 int searchName = 40;  
 int position = names.search(searchName);  
 System.*out*.println("20 is in " + position);  
 }  
 }



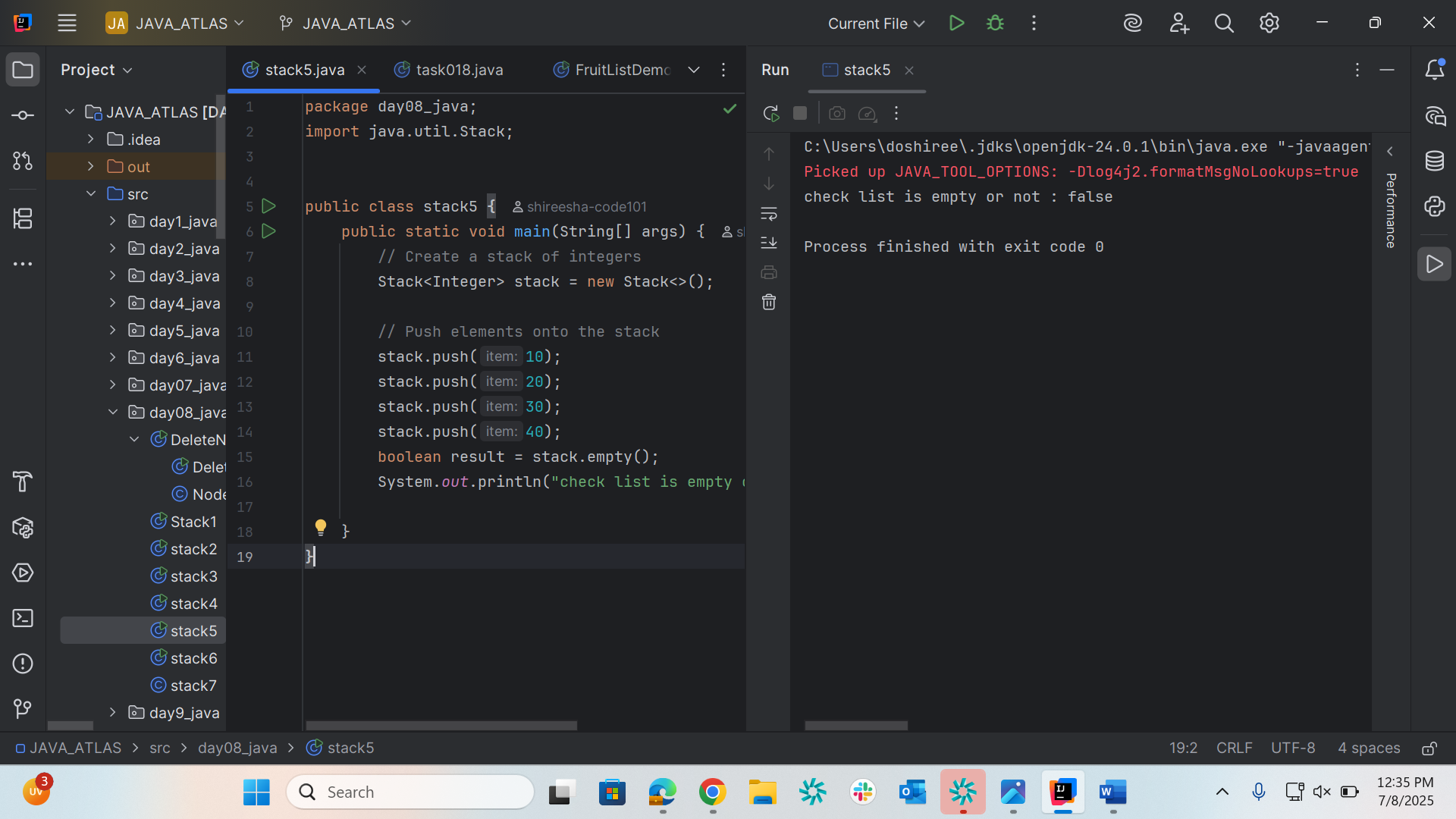
Task005

package day08\_java;  
  
import java.util.Stack;  
  
public class stack4 {  
 public static void main(String[] args) {  
 Stack<Object> stack = new Stack<>();  
  
 stack.push("Prasunamba");  
 stack.push("Meher");  
 stack.push(3);  
 Object top = stack.peek();  
 boolean result = stack.empty();  
 System.*out*.println("Top element (peek): " + top);  
 System.*out*.println("Stack after peek(): " + stack);  
 System.*out*.println(result);  
 }  
}



Task006

package day08\_java;  
import java.util.Stack;  
  
  
public class stack5 {  
 public static void main(String[] args) {  
 // Create a stack of integers  
 Stack<Integer> stack = new Stack<>();  
  
 // Push elements onto the stack  
 stack.push(10);  
 stack.push(20);  
 stack.push(30);  
 stack.push(40);  
 boolean result = stack.empty();  
 System.*out*.println("check list is empty or not : " + result);  
  
 }  
}



Task007

package day08\_java;  
  
public class stack6 {  
 int queueLength = 4;  
 int items[] = new int[queueLength];  
 int front = -1;  
 int back = -1;  
  
 void deQue(){  
 if(isEmpty()){  
 System.*out*.println("Queue is empty. Nothing to dequeue");  
 } else if (front == back){  
 front = back = -1;  
 } else {  
 front++;  
 }  
 }  
  
 void enQue(int itemValue) {  
 if(isFull()){  
 System.*out*.println("Queue is full");  
 } else if(front == -1 && back == -1){  
 front = back = 0;  
 items[back] = itemValue;  
 } else{  
 back++;  
 items[back] = itemValue;  
 }  
 }  
  
 boolean isFull(){  
 if(back == queueLength - 1){  
 return true;  
 } else {  
 return false;  
 }  
 }  
  
 boolean isEmpty(){  
 if(front == -1 && back == -1){  
 return true;  
 } else {  
 return false;  
 }  
 }  
  
 void display(){  
 int i;  
  
 if(isEmpty()){  
 System.*out*.println("Queue is empty");  
 } else {  
 for(i = front; i <= back; i++){  
 System.*out*.println(items[i]);  
 }  
 }  
 }  
  
 void peek(){  
 System.*out*.println("Front value is: " + items[front]);  
 }  
 public static void main(String[] args) {  
 stack6 myQueue = new stack6();  
 myQueue.enQue(100);  
 myQueue.enQue(200);  
 myQueue.enQue(300);  
 myQueue.enQue(400);  
  
 myQueue.display();  
  
 myQueue.peek();  
 }  
  
}

