Laboratorio 1

Nombre: Juan David Jiménez Romero

Codigo Main:

package arbolgenealogico\_lab1\_u2\_juanj;

public class ArbolGenealogico\_Lab1\_U2\_JuanJ {

public static void main(String[] args) {

//se ingresaban los datos al nodo principal

Node root = new Node("a","Alfonso Romero","16/08/1947","Masculino","Vivo");

//se agregan las ramas

root.left = new Node("b","Flor Romero","02/05/1967","Femenino","Vivo");

root.right = new Node("c","Paty Romero","20/02/1968","Femenino","Vivo");

root.left = new Node("d","Flor Romero","02/05/1967","Femenino","Vivo");

root.left = new Node("e","Diana Romero","13/07/1967","Femenino","Vivo");

root.left.left = new Node("d","d","","","");

root.left.right = new Node("e","e","","","");

root.right.right = new Node("f","f","","","");

root.left.left.left = new Node("g","g","","","");

root.left.right.left = new Node("h","h","","","");

//llamada a las los metodos para el recorrido del arbol binario in Orden

System.out.println("IN-ORDER TRAVERSAL: ");

BinaryTreeTraversals.inOrdenTraversal(root);

System.out.println();

//llamada al metodo para el reccorido del arbol binario pre Orden

System.out.println("PRE-ORDER TRAVERSAL: ");

BinaryTreeTraversals.preOrdenTraversal(root);

System.out.println();

//llamada al metodo para el recorrido del arbol binario post orden

System.out.println("POST-ORDER TRAVERSAL: ");

BinaryTreeTraversals.postOrdenTraversal(root);

System.out.println();

}

}

Codigo Node:

package arbolgenealogico\_lab1\_u2\_juanj;

public class Node {//declaracion de la clase Node

//declaracion de las variables

public String value;

public Node left;

public Node right;

//constructor de la clase Node

public Node (String value, String nombre, String fechaNacimiento, String genero, String estado){//formato del contructor

this.value = (value = String.format("""

Nombre: %s

Fecha de nacimiento: %s

Genero: %s

Estado: %s

-----------------------------------

""", nombre, fechaNacimiento,genero,estado));//se le crea un diseño al String value para mostrar los datos de una forma deseada

}

}

Codigo BinaryTreeTraversals:

package arbolgenealogico\_lab1\_u2\_juanj;

//clase Binary Tree Traversals donde estan los metodos para el recorrido del arbol binario

public class BinaryTreeTraversals {

//metodo para realizar el recorrido inOrden del arbol donde se pide un nodo para recorrerlo

public static void inOrdenTraversal(Node node){

if(node !=null){//validacion del metodo para no realizarlo en caso de estar vacio

inOrdenTraversal(node.left);//la forma de recorrido dice que empieza desde la izquierda

System.out.print(node.value+" ");//imprime el valor da value y le da un espaciado

inOrdenTraversal(node.right);//toma el valor de la derecha

}}

//metodo para realizar el recorrido preOrden del arbol

public static void preOrdenTraversal(Node node){

if(node !=null){//validacion del metodo para no realizarlo en caso de estar vacio

System.out.print(node.value+" ");//imprime el valor da value y le da un espaciado

preOrdenTraversal(node.left);//toma el valor de la izquierda

preOrdenTraversal(node.right);//toma el valor de la derecha

}}

//metodo para realizar el recorrido postOrden del arbol

public static void postOrdenTraversal (Node node){

if(node !=null){//validacion del metodo para no realizarlo en caso de estar vacio

postOrdenTraversal(node.left);//la forma de recorrido dice que empieza desde la izquierda

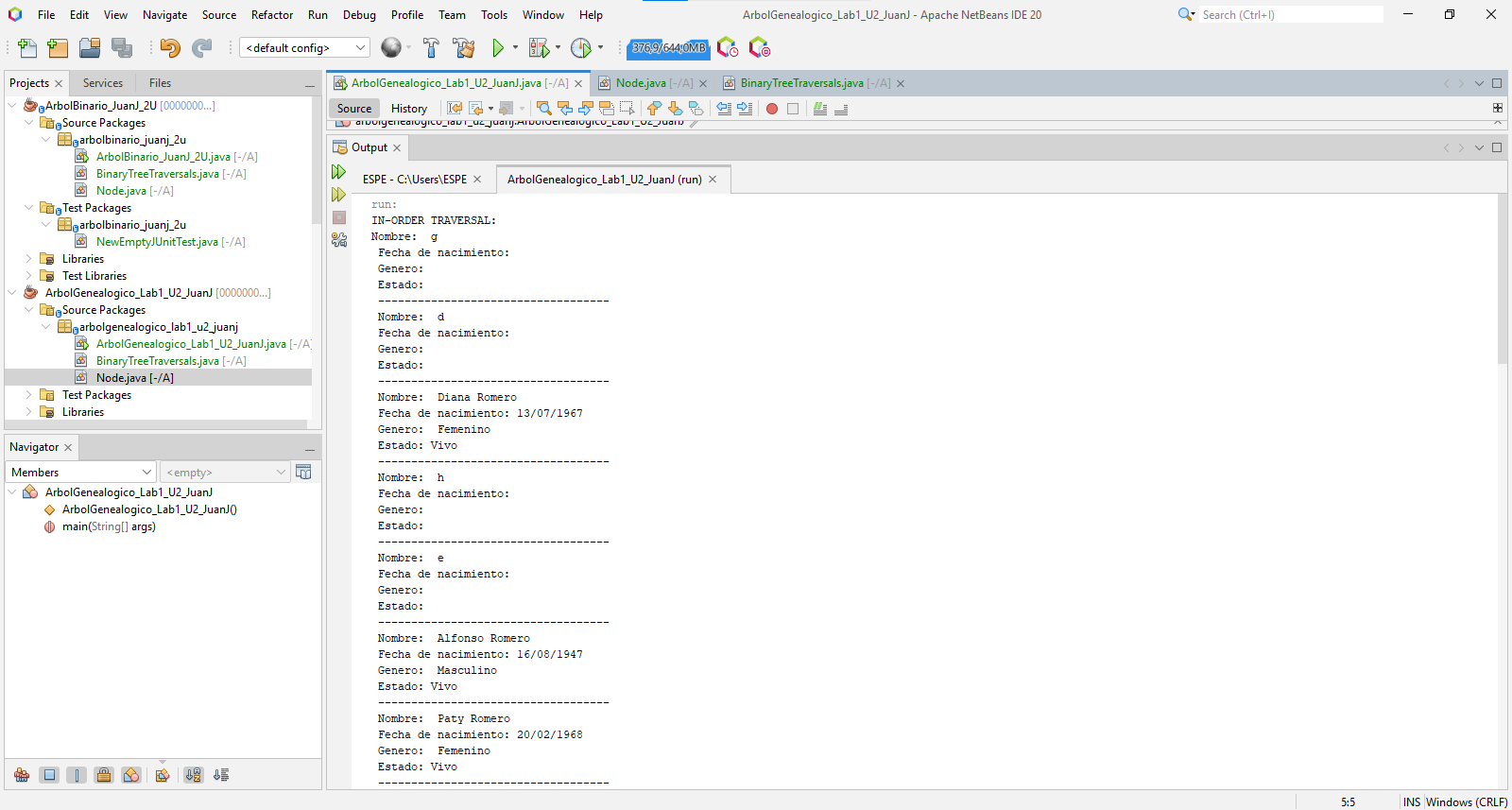
postOrdenTraversal(node.right);//toma el valor de la derecha

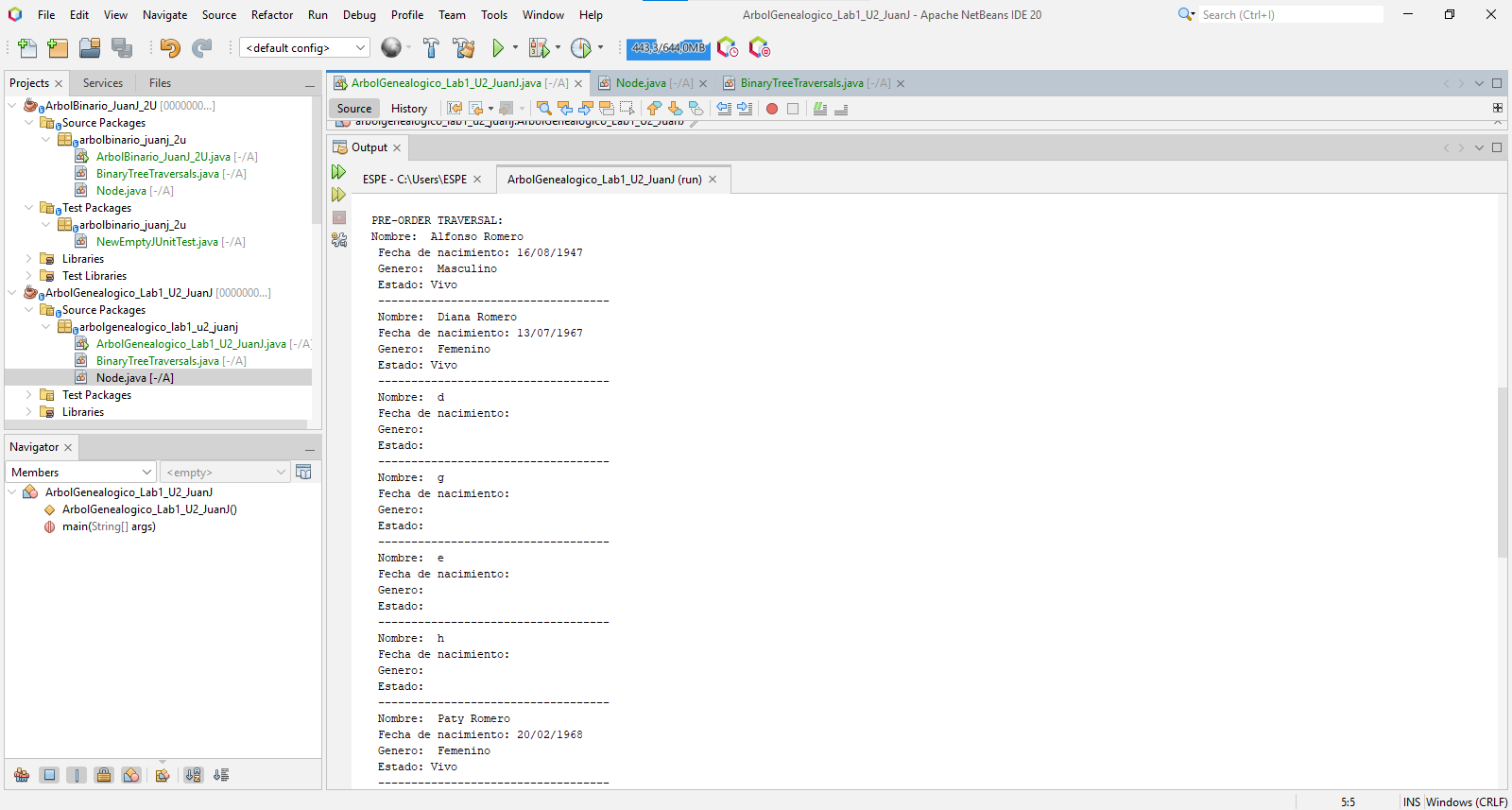
System.out.print(node.value+" ");//imprime el valor da value y le da un espaciado

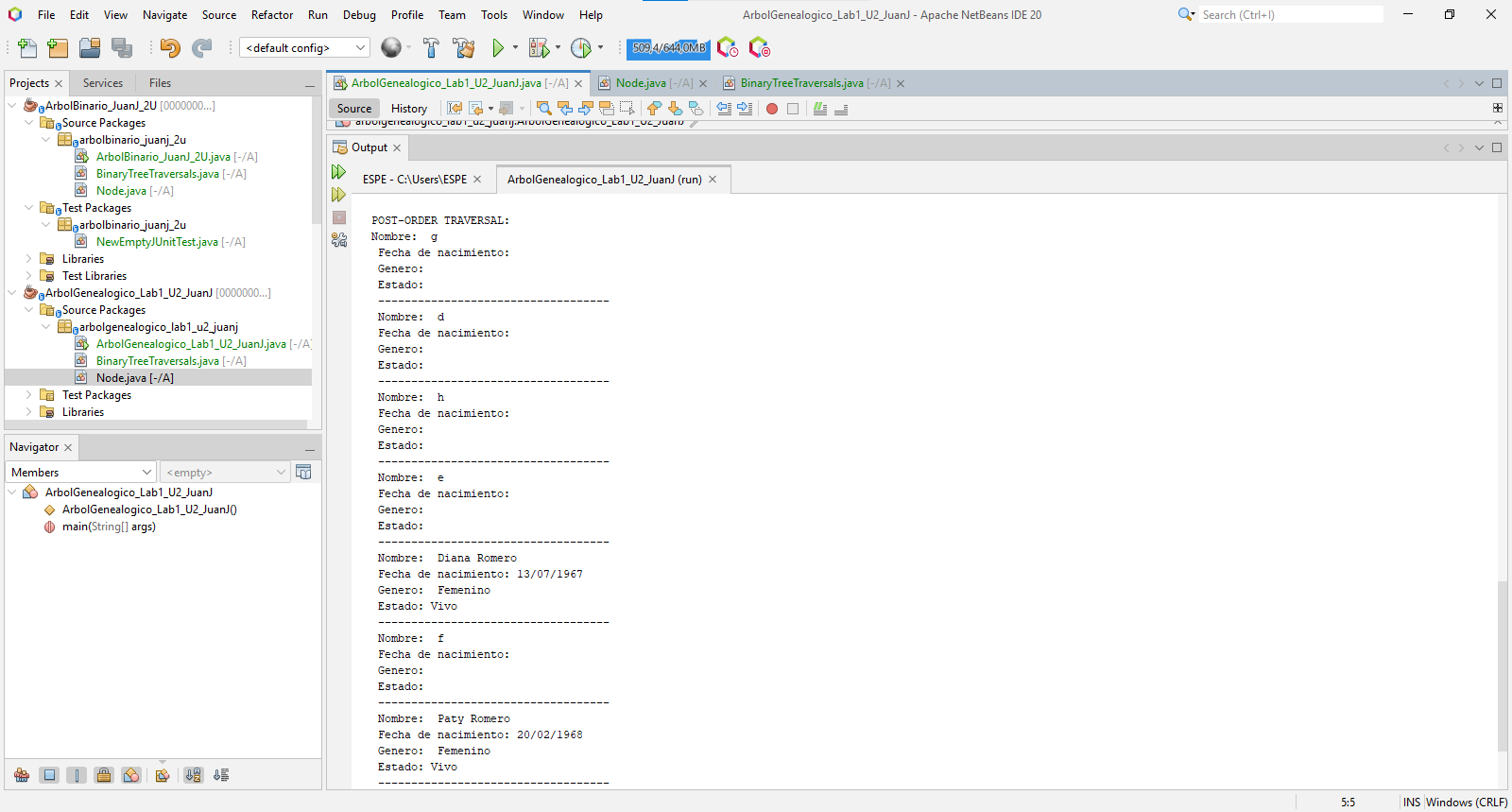
}}

}

Ejecución del programa:







El programa ingresaba de manera estática los datos de los familiares y tenía para realizar el recorrido, pero al momento de recorrerlo este no mostraba correctamente los datos, y faltaba implementarle para que muestre el árbol y un menú de opciones para realizar cada función por separado y no tener mucho texto en la consola.