

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Arbol_Binario
8 {
9     class NodoArbol
10    {
11        int dato;
12        NodoArbol izq;
13        NodoArbol der;
14
15        public NodoArbol()
16        {
17            this.dato = 0;
18            this.izq = null;
19            this.der = null;
20        }
21
22        public int Dato { get => dato; set => dato = value; }
23        public NodoArbol Izq { get => izq; set => izq = value; }
24        public NodoArbol Der { get => der; set => der = value; }
25    }
26
27 }
28
29
```

```
1 namespace Arbol_Binario
2 {
3     class arbolito
4     {
5         NodoArbol raiz, nuevo, aux;
6         bool xx;
7         public int suma;
8         public int pares;
9         public int impares;
10        public arbolito()
11        {
12            this.raiz = null;
13            this.nuevo = null;
14            this.aux = raiz;
15            this.xx = false;
16            this.suma = 0;
17            this.pares = 0;
18            this.impares = 0;
19        }
20
21        public void crearNodo(int d)
22        {
23            this.nuevo = new NodoArbol();
24            nuevo.Dato = d;
25            nuevo.Izq = null;
26            nuevo.Der = null;
27        }
28
29        public void crear_arbol(int d)
30        {
31            raiz = crear_arbol(d, raiz);
32        }
33
34        public NodoArbol crear_arbol(int d, NodoArbol x)
35        {
36            crearNodo(d);
37            // aux = x;
38            if (x == null) {even(d); suma += d; return nuevo; }
39            else
40            {
41                if (d < x.Dato) { x.Izq = crear_arbol(d, x.Izq);}
42                else if (d > x.Dato) { x.Der = crear_arbol(d, x.Der); }
43                else { return x; }
44                return x;
45            }
46        }
47
48        public bool buscar(int dato, NodoArbol x)
49        {
```

```
50         if (x == null) { return false; }
51         if (x.Dato == dato) { xx= true; }
52         else if (dato > x.Dato) { xx = buscar(dato, x.Der); }
53         else if (dato < x.Dato) { xx = buscar(dato, x.Izq); }
54         else xx = false;
55
56         return xx;
57     }
58
59     public NodoArbol get_raiz()
60     {
61         return this.raiz;
62     }
63
64     void even(int dato)
65     {
66         if (dato % 2 == 0) {
67             pares++;
68         }
69         else
70         {
71             impares++;
72         }
73     }
74 }
75 }
76
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Arbol_Binario
12 {
13     public partial class principal : Form
14     {
15
16         public principal()
17         {
18             InitializeComponent();
19         }
20
21
22
23         private void BtnAgregar_Click(object sender, EventArgs e)
24         {
25             StaticClass.a.crear_arbol(int.Parse(txtDato.Text));
26             txtDato.Clear();
27         }
28         private void preOrder(NodoArbol punt,int acc)
29
30             if(punt != null)
31             {
32                 acc += punt.Dato;
33                 lstpreOrden.Items.Add(punt.Dato);
34                 preOrder(punt.Izq, acc);
35                 preOrder(punt.Der, acc);
36
37             }
38         }
39
40         private void inOrder(NodoArbol punt)
41         {
42             if(punt != null)
43             {
44
45                 inOrder(punt.Izq);
46                 lstInOrdne.Items.Add(punt.Dato);
47
48                 inOrder(punt.Der);
49             }
50         }
51     }
52 }
```

```
50     }
51     private void postOrder(NodoArbol punt)
52     {
53         if(punt != null)
54         {
55             postOrder(punt.Izq);
56             postOrder(punt.Der);
57             lstPosrOrden.Items.Add(punt.Dato);
58         }
59     }
60
61     private void BtnPreordne_Click(object sender, EventArgs e)
62     {
63
64         NodoArbol punt = new NodoArbol();
65         punt = StaticClass.a.get_raiz();
66         if(punt != null)
67         {
68             preOrder(punt, 0);
69         }
70         else
71         {
72             MessageBox.Show("The tree is empty");
73         }
74     }
75
76     private void BtnInordern_Click(object sender, EventArgs e)
77     {
78         int acc = 0;
79         NodoArbol punt = new NodoArbol();
80         punt = StaticClass.a.get_raiz();
81         if (punt == null)
82         {
83             MessageBox.Show("Arbol vacio");
84         }
85         else
86         {
87             inOrder(punt);
88         }
89     }
90
91     private void BtnPostOrden_Click(object sender, EventArgs e)
92     {
93         NodoArbol punt = StaticClass.a.get_raiz();
94         if (punt == null) {
95             MessageBox.Show("Arbon vacio");
96         }
97         else
98         {
```

```
99         postOrder(punt);
100     }
101 }
102
103 private void BtnLimpiar_Click(object sender, EventArgs e)
104 {
105     lstInOrdne.Items.Clear();
106     lstPosrOrden.Items.Clear();
107     lstpreOrden.Items.Clear();
108 }
109
110
111
112 private void BtnBuscar_Click(object sender, EventArgs e)
113 {
114     if (StaticClass.a.buscar(int.Parse(txtDato.Text),           ↗
115         StaticClass.a.get_raiz()))
116     {
117         MessageBox.Show("El Numero se encuentra registrado");
118     }
119     else
120     {
121         MessageBox.Show("No se encuentra registrado");
122     }
123 }
124
125 private void button1_Click(object sender, EventArgs e)
126 {
127     MessageBox.Show("La suma de los datos del arbol es: " +   ↗
128         StaticClass.a.suma);
129 }
130
131 private void BtnEliminar_Click(object sender, EventArgs e)
132 {
133     MessageBox.Show("los numeros pares son : " +             ↗
134         StaticClass.a.pares + "\n" + "los numeros impares son: " + ↗
135         StaticClass.a.impares);
136 }
137 }
```