

```

1 using System;
2 using System.Collections;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading;
9 using System.Windows;
10 using RubricaModale;
11
12 namespace RubricaModale
13 {
14     /// <summary>
15     /// Michele Sprocatti 4E
16     /// Inserimento,
17     /// modifica(solo stato civile e foto dopo aver inserito nome e cognome),
18     /// elimina(effettuando doppio click sull'elemento o premendo il tasto
19     /// elimina che permette di effettuare la ricerca tramite nome e cognome),
20     /// setup(solo un aumento disponibile), ricerca(tramite nome e cognome),
21     /// cancella tutto, visualizza
22     /// </summary>
23     public partial class Rubrica : Form
24     {
25         public List<Persona> lista = new List<Persona>(); (3) dimensione lista
26         int dim = 3; utilizzare proprietà capacity
27         public Rubrica()
28         {
29             InitializeComponent();
30
31         private void cancellaArchivioToolStripMenuItem1_Click(object sender,
32             EventArgs e)
33         {
34             //cancella tutto l'archivio
35             lista.Clear();
36             listBox.Items.Clear();
37
38         private void inserisciToolStripMenuItem_Click(object sender, EventArgs
39             e)
40         {
41             //inserire nuovo elemento
42             DialogResult d;
43             if (lista.Count < dim)//controllo se si può inserire
44             {
45                 FormModale form = new FormModale();
46                 form.ShowDialog();//apertura form modale
47                 if (form.DialogResult == DialogResult.OK)
48                 {
49                     listBox.Items.Add("Cognome:" + form.p1.cognome + ",Nome:"
50                     + form.p1.nome + ",Sesso:" + form.p1.Sesso + ",Stato
51                     civile:" + form.p1.stato);
52                     lista.Add(form.p1);//aggiunta
53                     listBox.SelectedIndex = 0;
54                 }
55             }
56         }
57     }
58 }

```

```
51     }
52     else
53     {
54         if(lista.Count==3) costante!!!
55         {
56             //setup rubrica
57             d=MessageBox.Show("Massima dimensione raggiunta, aumentare
58             dimensione?", "Impossibile ampliare rubrica",
59             MessageBoxButtons.YesNo, MessageBoxIcon.Information);
60             if(d==DialogResult.Yes)
61             setupRubricaToolStripMenuItem.PerformClick();//richiamo
62             click setuprubrica
63         }
64         else
65         {
66             MessageBox.Show("Effettuato già l'unico ampliamento
67             disponibile", "Impossibile ampliare rubrica",
68             MessageBoxButtons.OK, MessageBoxIcon.Information);
69         }
70     }
71 }
72 private void setupRubricaToolStripMenuItem_Click(object sender,
73 EventArgs e)
74 {
75     setuprubrica form = new setuprubrica();//aumento dimensione
76     tramite codice(123)
77     form.ShowDialog();
78     if(form.DialogResult==DialogResult.OK)
79     {
80         dim = form.dim; Gestire dimensione della lista
81     }
82 }
83 private void modificaToolStripMenuItem_Click(object sender, EventArgs e)
84 {
85     RicercaToolStripMenuItem.PerformClick();
86     if (lista.Count > 0)//controllo se elementi presenti
87     {
88         FormModale form = new FormModale(lista
89         [listBox.SelectedIndex],false);
90         form.ShowDialog();
91         int i = listBox.SelectedIndex;
92         if (form.DialogResult == DialogResult.OK)
93         {
94             listBox.Items.RemoveAt(listBox.SelectedIndex);
95             listBox.Items.Add("Cognome:" + form.p1.cognome + ",Nome:"
96             + form.p1.nome + ",Sesso:" + form.p1.Sesso + ",Stato
97             civile:" + form.p1.stato);
98             lista[i] = form.p1;
99             listBox.SelectedIndex = 0;
100         }
101     }
102     else
103     {
104         MessageBox.Show("Non ci sono elementi
105         presenti", "Impossibile", MessageBoxButtons.OK, MessageBoxIcon.
106         Exclamation);
107     }
108 }
109 private void eliminaToolStripMenuItem_Click(object sender, EventArgs e)
```

```
93     {
94         RicercaToolStripMenuItem.PerformClick();
95         if (lista.Count > 0)
96         {
97             lista.RemoveAt(listBox.SelectedIndex); //rimozione elemento
98             listBox.Items.RemoveAt(listBox.SelectedIndex);
99             if (lista.Count > 0)
100                 listBox.SelectedIndex = 0;
101         }
102         else
103             MessageBox.Show("Non ci sono elementi presenti",
104                             "Impossibile", MessageBoxButtons.OK,
105                             MessageBoxIcon.Exclamation);
106     }
107 private void VisualizzaToolStripMenuItem1_Click(object sender, EventArgs
108     e)
109     {
110         if (lista.Count > 0)
111         {
112             FormModale form = new FormModale(lista[listBox.SelectedIndex],
113             true);
114             form.ShowDialog(); //form modale per visualizzare
115         }
116     }
117 private void RicercaToolStripMenuItem_Click(object sender, EventArgs
118     e)
119     {
120         if (lista.Count > 0) //controllo se elementi presenti
121         {
122             bool trovato = false;
123             FormModale form = new FormModale(true);
124             form.ShowDialog();
125             for (int i = 0; i < lista.Count && !trovato; i++)
126             {
127                 if (form.p1.nome == lista[i].nome && form.p1.cognome ==
128                 lista[i].cognome) //ricerca
129                 {
130                     listBox.SelectedIndex = i;
131                     trovato = true;
132                 }
133             }
134             if (!trovato) //elemento non trovato
135                 MessageBox.Show("Elemento non trovato", "Non presente",
136                 MessageBoxButtons.OK, MessageBoxIcon.Information);
137         }
138         else
139             MessageBox.Show("Non ci sono elementi presenti",
140                             "Impossibile", MessageBoxButtons.OK,
141                             MessageBoxIcon.Exclamation);
142     }
143 private void listBox_DoubleClick(object sender, EventArgs e)
144     {
145         if (lista.Count > 0)
146         {
147             lista.RemoveAt(listBox.SelectedIndex); //rimozione elemento
```

```
140         listBox.Items.RemoveAt(listBox.SelectedIndex);
141         if (lista.Count > 0)
142             listBox.SelectedIndex = 0;
143     }
144     else
145         MessageBox.Show("Non ci sono elementi presenti",
146             "Impossibile", MessageBoxButtons.OK,
147             MessageBoxIcon.Exclamation);
148 }
149 }
```