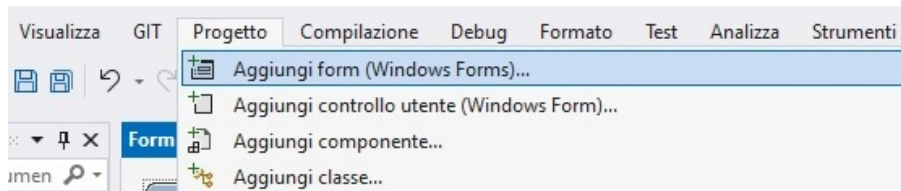


# Gli approfondimenti di

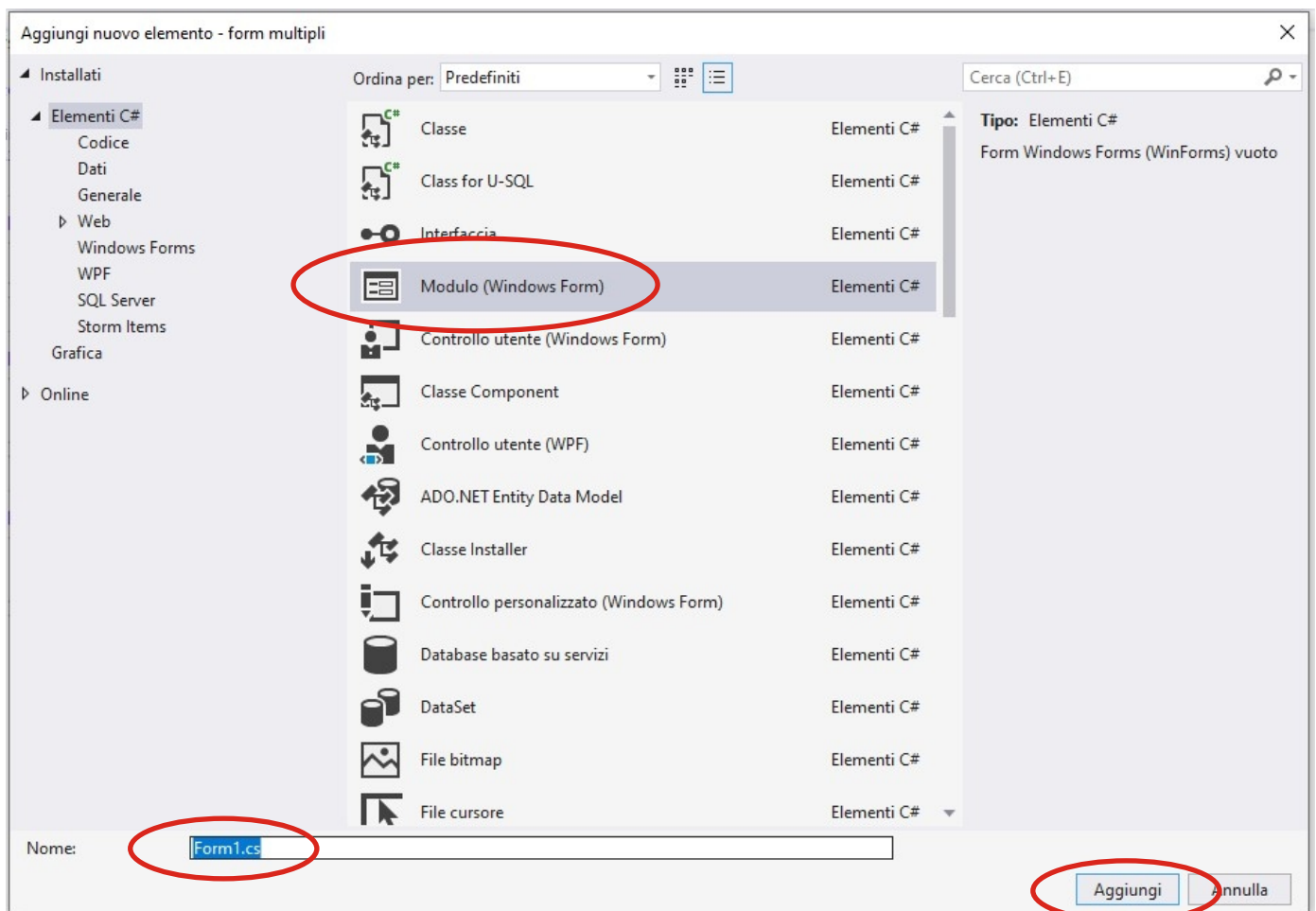
# L@BOROBOTICA

## FORM MULTIPLI CON SERIALE

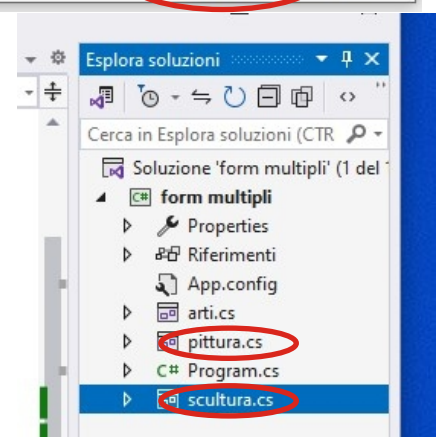
Può succedere, all'interno di un progetto, di voler aggiungere dei form con programmi differenti, per poter gestire situazioni articolate.



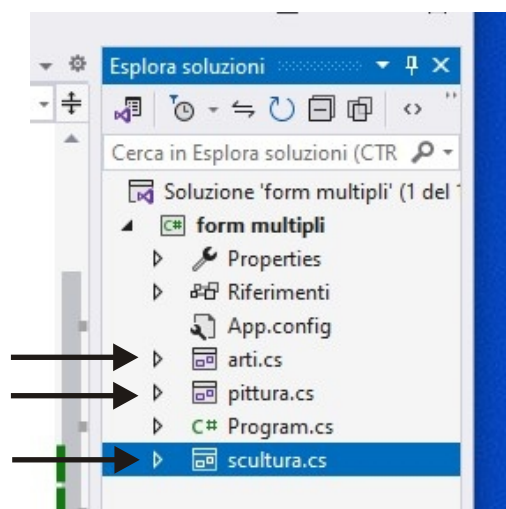
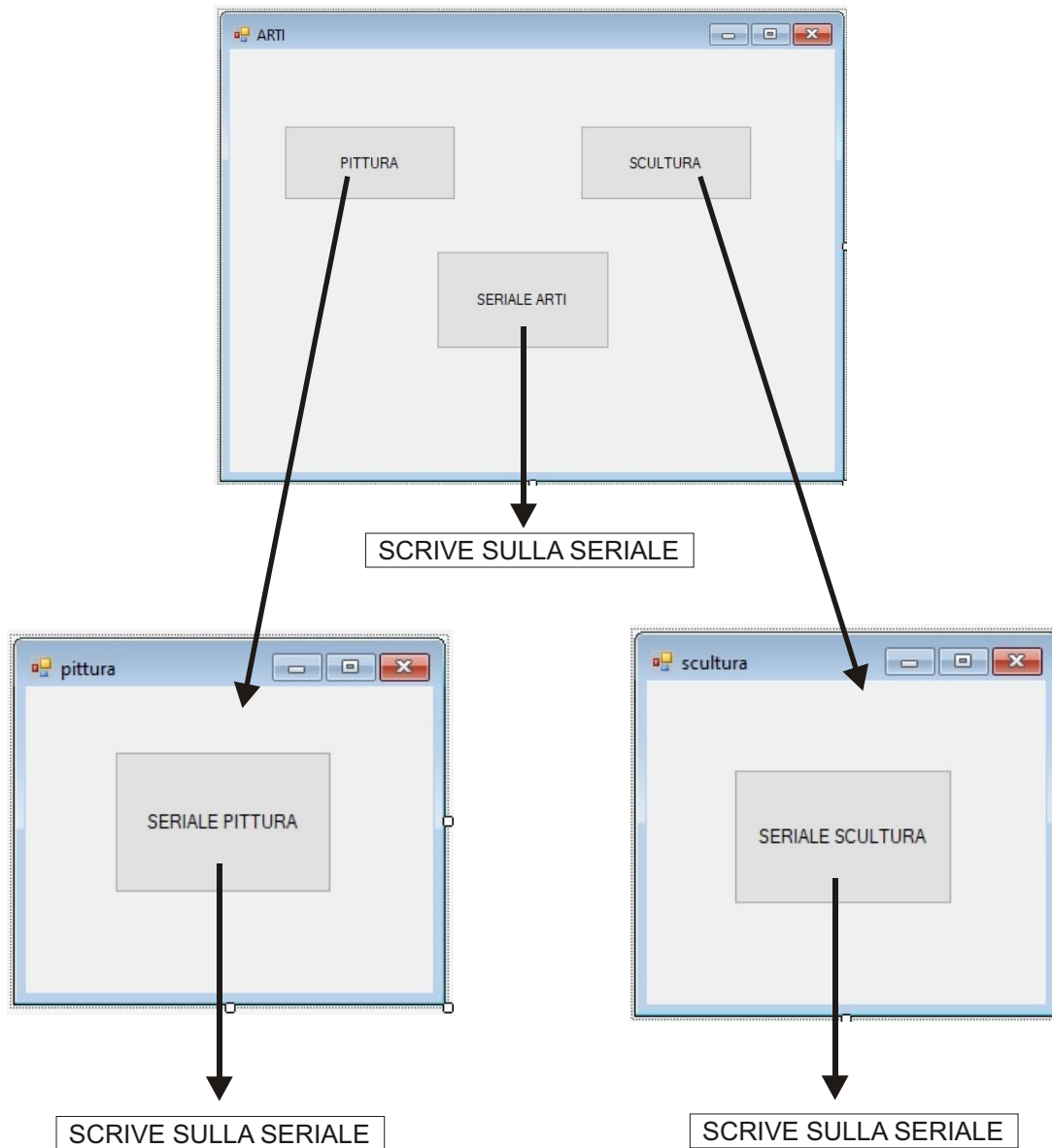
Portiamoci in Progetto, e selezioniamo Aggiungi form (Windows Form).



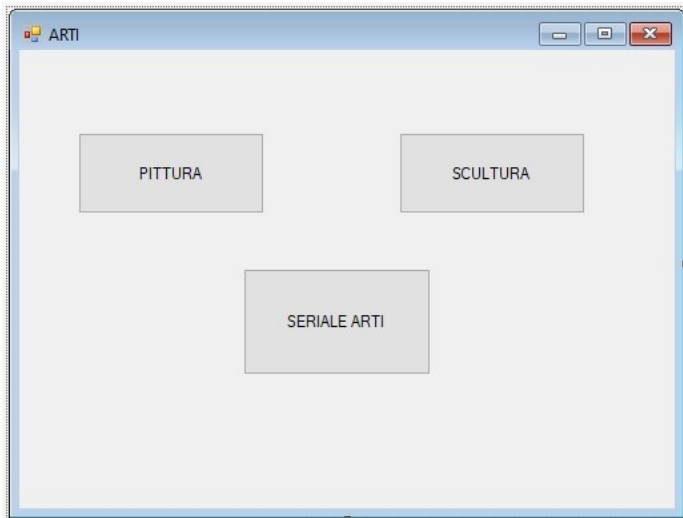
Nel campo "Nome" digitiamo il nome che vogliamo assegnare al nuovo form. Poi click su "Aggiungi". Il nuovo form si potrà visualizzare nella finestra delle soluzioni. Creiamo due elementi: pittura e scultura.



Quello che vogliamo ottenere è che premendo “PIITURA” e “SCULTURA” si aprano due form ausiliari, ognuno in grado di comunicare dati sulla seriale.



# FORM PRINCIPALE "ARTI"



Nel form "padre" inseriamo tre buttons: cmdPittura e cmdScultura richiameranno i form secondari, mentre cmdSerialeArti, andrà a scrivere sulla seriale un carattere vuoto. Non dimentichiamoci di aggiungere il componente nascosto serialPort1.

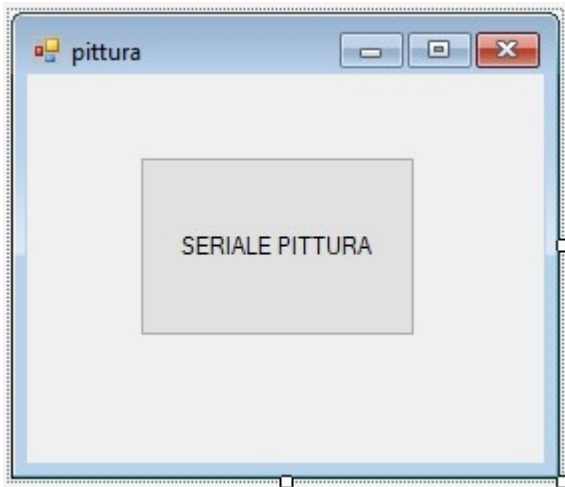
```
1  using System;
2  using System.Windows.Forms;
3  namespace form_multipli
4  {
5      3 riferimenti
6      public partial class arti : Form
7      {
8          1 riferimento
9          public arti()
10         {
11             InitializeComponent();
12         }
13
14         1 riferimento
15         private void cmdPittura_Click(object sender, EventArgs e)
16         {
17             pittura pitturaForm = new pittura();
18             pitturaForm.ShowDialog();
19         }
20
21         1 riferimento
22         private void cmdScultura_Click(object sender, EventArgs e)
23         {
24             scultura sculturaForm = new scultura();
25             sculturaForm.ShowDialog();
26         }
27
28         1 riferimento
29         private void cmdSerialeArti_Click(object sender, EventArgs e)
30         {
31             serialPort1.Open();
32             serialPort1.Write(" ");
33             serialPort1.Close();
34         }
35     }
36 }
```

Premendo il button cmdPittura, richiamiamo il form pittura.

Premendo il button cmdScultura, richiamiamo il form scultura.

Premendo il button cmdSerialeArti si apre la serial port, si scrive un carattere vuoto e si chiude la serial port.

# FORM AUSILIARIO “PITTURA”

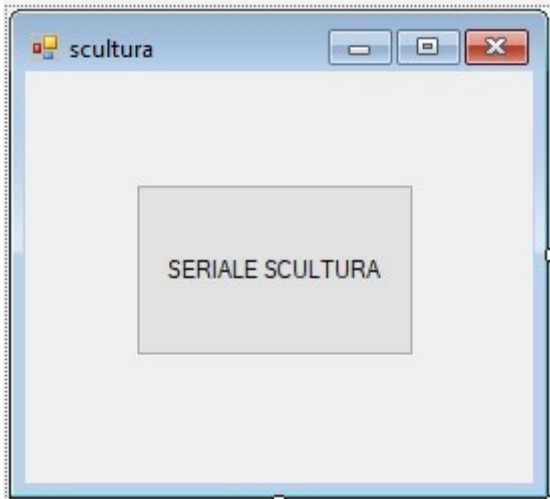


Nel form ausiliario “pittura” inseriamo un solo button (cmdSerialePittura) che stamperà sulla seriale un carattere vuoto. Non dimentichiamoci di aggiungere il componente nascosto serialPort1.

```
1  using System;
2  using System.Windows.Forms;
3
4  namespace form_multipli
5  {
6      public partial class pittura : Form
7      {
8
9          public pittura()
10         {
11             InitializeComponent();
12         }
13
14
15         private void cmdSerialePittura_Click(object sender, EventArgs e)
16         {
17             serialPort1.Open();
18             serialPort1.Write(" ");
19             serialPort1.Close();
20         }
21     }
22 }
23
```

Premendo il button cmdSerialePittura si apre la serial port, si scrive un carattere vuoto e si chiude la serial port.

# FORM AUSILIARIO “SCULTURA”



Nel form ausiliario “scultura” inseriamo un solo button (cmdSerialeScultura) che stamperà sulla seriale un carattere vuoto. Non dimentichiamoci di aggiungere il componente nascosto serialPort1.

```
1  using System;
2  using System.Windows.Forms;
3
4  namespace form_multipli
5  {
6      public partial class scultura : Form
7      {
8          public scultura()
9          {
10             InitializeComponent();
11         }
12
13
14         private void cmdSerialeScultura_Click(object sender, EventArgs e)
15         {
16             serialPort1.Open();
17             serialPort1.Write(" ");
18             serialPort1.Close();
19         }
20     }
21 }
22
```

Premendo il button cmdSerialeScultura si apre la serial port, si scrive un carattere vuoto e si chiude la serial port.