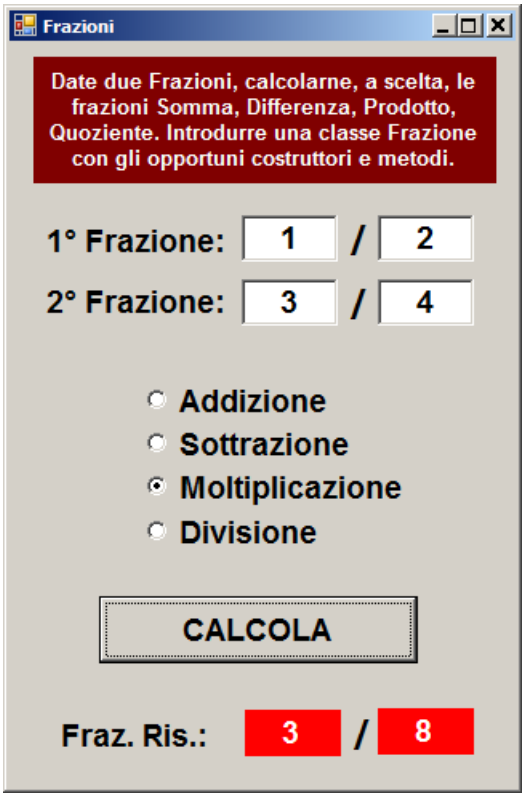
Esempio Pilota:

ClasseFrazione_ProprietaElementiPrivati

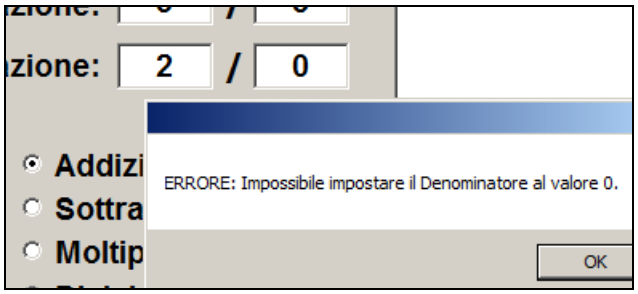
Definizione della Form




```
TextBox txtNumA - TextBox txtDenA
TextBox txtNumB - TextBox txtDenB

Button plsCalcola
RadioButton radDivisione
RadioButton radMoltiplicazione
RadioButton radSottrazione
RadioButton radAddizione

Label lblNum - Label lblDen
```



Codice della Form

```
using System;
using System.Windows.Forms;

namespace ClasseFrazioneConCostruttoreInCSharp
{
   public partial class frmOperazioniSulleFrazioni : Form
   {
      public frmOperazioniSulleFrazioni()
      {
         InitializeComponent();
      }

      private void plsCalcola_Click(object sender, EventArgs e)
      {

// Creiamo, tramite il COSTRUTTORE VUOTO, gli oggetti FA FB e usiamo
// le PROPRIETA' per impostarne i valori di Numeratore e Denominatore.

         Frazione FA = new Frazione();
         FA.Numeratore = Convert.ToInt16(txtNumA.Text);
         FA.Denominatore = Convert.ToInt16(txtDenA.Text);

         Frazione FB = new Frazione();
         FB.Numeratore = Convert.ToInt16(txtNumB.Text);
         FB.Denominatore = Convert.ToInt16(txtDenB.Text);

// N.B.: Assegnando dei valori alle proprietà ne viene eseguita la zona SET.
//       Il valore assegnato viene reso disponibile nella variabile "value".



         // creiamo l'oggetto FR, cioè la frazione risulatato
         // usando il COSTRUTTORE "vuoto", ossia senza parametri ...

         Frazione FR = new Frazione();


         // calcoliamo FR in base alla richiesta dell'utente ...

         if (radAddizione.Checked)

            // il metodo Somma pone in FR la frazione FA+FB ...
            FR.Somma(FA, FB);

         else if (radSottrazione.Checked)

            // il metodo Differenza pone in FR la frazione FA-FB ...
            FR.Differenza(FA, FB);

         else if (radMoltiplicazione.Checked)

            // il metodo Prodotto pone in FR la frazione FA * FB ...
            FR.Prodotto(FA, FB);

         else if (radDivisione.Checked)

            // il metodo Quoziente pone in FR la frazione FA / FB ...
            FR.Quoziente(FA, FB);


         // Per visualizzare il risultato, leggiamo le proprietà Numeratore e
         // Denomin. di FR e le assegniamo alle due labal lblNum e lblDen

         lblNum.Text = Convert.ToString( FR.Numeratore );
         lblDen.Text = Convert.ToString( FR.Denominatore );

// N.B.: Leggendo i valori delle proprietà ne viene eseguita la zona GET.
      }
   }
}
```

Codice della Classe Frazione

```
using System;
using System.Windows.Forms;

namespace ClasseFrazioneConCostruttoreInCSharp
{
   class Frazione
   {

      // Area Dati interna ...

      private int _Num;   // ... memorizza il Numeratore della frazione
      private int _Den;   // ... memorizza il Denominatore della frazione


      // COSTRUTTORI ...

      // Il Costruttore "vuoto" (senza parametri) inizializza l'oggetto
      // alla frazione nulla (Numeratore 0 e Denominatore 1 ... 0/1) ...

      public Frazione()
      {
         _Num = 0;
         _Den = 1;
      }

      // Il Costruttore inizializza l'oggetto impostando Numeratore e
      // Denominatore in base ai valori che riceve nei parametri ...

      public Frazione (int NuovoNum, int NuovoDen)
      {
         _Num = NuovoNum;
         _Den = NuovoDen;
      }


      // PROPRIETA' ...

// La zona GET viene eseguita quando
// la proprietà è usata in una espressione (lettura).

// Esempio:    lblNum.Text = Converto.ToString( FA.Numeratore )

// La zona SET viene eseguita quando
// alla proprietà viene assegnato un valore (scrittura).

// Esempio:    FA.Numeratore = Convert.ToInt16 ( txtNumA.Text )

// Nella variabile "value" è disponibile il valore
// che viene assegnato alla proprietà.


      // La proprietà Numeratore permette di leggere/impostare
      // dall'esterno il valore del numeratore della frazione ...

      public int Numeratore
      {
            // ... restituisci _Num che contiene il numeratore
            get { return _Num; }

              // ... memorizza in _Num il valore assegnato (value)
            set { _Num = value; }
      }

      // La proprietà Denominatore permette di leggere/impostare
      // dall'esterno il valore del Denominatore della frazione.
      // Il codice della proprietà IMPEDISCE di assegnarvi il valore 0.

      public int Denominatore
      {
            // ... restituisci _Den che contiene il numeratore
            get { return _Den; }

            set
            {

            // Impedisci di memorizzare un denominatore nullo ...
            if (value == 0)
                  MessageBox.Show("ERRORE: Impossibile impostare il
                                   Denominatore al valore 0.");
            else
                  // ... memorizza in _Den il valore assegnato (value)
                  _Den = value;
      }
   }


      // METODI ...

      // Il metodo Somma riceve, come paramtri, due frazione F1 ed F2 e
      // imposta l'oggetto frazione in modo che sia la frazione Somma F1+F2
      // (ossia calcola numeratore e denominatore di F1+F2 e
                                                li pone in _Num e _Den).

      public void Somma(Frazione F1, Frazione F2)
      {
         _Num = F2._Den * F1._Num + F1._Den * F2._Num;
         _Den = F1._Den * F2._Den;

         Semplifica();
      }

      // I metodi Differenza, Prodotto e Quoziente agiscono come Somma ...

      public void Differenza(Frazione F1, Frazione F2)
      {
         _Num = F2._Den * F1._Num - F1._Den * F2._Num;
         _Den = F1._Den * F2._Den;

         Semplifica();
      }

      public void Prodotto (Frazione F1, Frazione F2)
      {
         _Num = F1._Num * F2._Num;
         _Den = F1._Den * F2._Den;

         Semplifica();
      }

      public void Quoziente(Frazione F1, Frazione F2)
      {
         _Num = F1._Num * F2._Den;
         _Den = F1._Den * F2._Num;

         Semplifica();
      }
```

```
    // Il metodo Visualizza riceve come parametri, due Label e, in esse,
    // pone il numeratore e il denominatore dell'oggetto frazione.

    public void Visualizza(Label lblNum, Label lblDen)
    {
       lblNum.Text = _Num.ToString();
       lblDen.Text = _Den.ToString();
    }


    // il metodo Semplifica richiama il sottoprogramma privato MCD ...

    public void Semplifica()
    {
       int M = MCD(_Num, _Den);

       _Num = _Num / M;
       _Den = _Den / M;
    }



    // ELEMENTI PRIVATI ...

    // il sottoprogramma privato MCD utilizza il metodo di Euclide
    // nella versione cosiddetta delle "sottrazioni successive" ...

    private int MCD(int N1, int N2)
    {
       if ((N1 == 0) || (N2 == 0))
          return 1;
       else
       {
          while (N1 != N2)

             if (N1 > N2)
                N1 = N1 - N2;
             else
                N2 = N2 - N1;

          return N1;
       }
    }

   }
}
```