

## Applicazione Visual C# e Visual Basic .Net multiform - Passaggio di dati tra i form

I progetti e gli esercizi Visual C# e Visual Basic .Net proposti nell'Area Download del sito [www.maurodeberardis.it](http://www.maurodeberardis.it), riguardano quasi esclusivamente programmi con un solo Windows Form. Lavorare con un solo Form è infatti più semplice e consente di illustrare le tecniche di programmazione di base ed avanzate dei due linguaggi di programmazione. Bisogna però tener conto che spesso in un progetto è utile avere più form, ad esempio per realizzare un'attività separata dal form corrente, e che nei casi applicativi molte applicazioni Windows presentano più form: un form principale caricato all'avvio e altri form secondari accessibili da quello principale. Per questo motivo, il tutorial spiega come creare applicazioni multiform e come passare i dati da un form all'altro.

L'applicazione di esempio che ho scelto è molto semplice ed è costituita da un form principale Form1 e da un form secondario Form2.

- Quando si clicca sul bottone "Analizza" il Form1 richiama il Form2, **passandogli il testo da analizzare**
- Nel Form2 viene eseguito il conteggio dei caratteri, delle maiuscole e delle minuscole contenute nel testo inserito nel Form1. Cliccando sul bottone "Chiudi", il Form2 viene chiuso e il controllo passa di nuovo al Form1

Il passaggio di dati da un form ad un altro viene realizzato attraverso il costruttore del form che riceve i dati: **Form1 invia i dati come argomenti del costruttore di Form2**. Ci sono diversi altri metodi per passare i dati da un form all'altro ma l'approccio dei costruttori credo sia il più immediato

Questo tutorial e l'applicazione, sia nell'implementazione Visual C# che nell'implementazione Visual Basic .Net, si possono scaricare dall'Area Download del sito [www.maurodeberardis.it](http://www.maurodeberardis.it)

Facendo riferimento alle due immagini sopra, vediamo come realizzare l'applicazione, prima in Visual C# e poi in Visual Basic .Net.

## Progettazione visuale

E' la stessa sia per la soluzione Visual C# che per la soluzione Visual Basic .Net

### Creazione Form1

1. Creiamo un nuovo progetto (a cui diamo per esempio il nome MultiformVCS per il Visual C# e il nome MultiformVBN per il Visual Basic .Net) e scegliamo di realizzare un'applicazione Windows Form. Automaticamente verrà creato un modulo, ovvero un form, predefinito come *Form1*
2. Lavorando in modalità progettazione, inseriamo in *Form1* i controlli:
  - Label *label1 label2*
  - TextBox *tTestoForm1* con la proprietà Multiline=True
  - Button *bSalva bReset bAnalizza bChiudi*

### Creazione Form2

3. Aggiungiamo un secondo form selezionando dal menù "Progetto" l'opzione "Aggiungi Windows Form". Viene proposto, e noi lo accettiamo, il nome *Form2*
4. Lavorando in modalità progettazione, inseriamo in *Form2* i controlli:
  - Label *label1 label2 IOutput1 IOutput2 IOutput3*
  - TextBox *tTestoForm2* con le proprietà Multiline=True e ReadOnly=True
  - Button *bChiudi*

## Soluzione Visual C#

### Codice Form1

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace MultiformVCS
{
    public partial class Form1 : Form
```

```

{
    public Form1()
    {
        InitializeComponent();
    }
    private void bReset_Click(object sender, EventArgs e)
    {
        tTestoForm1.Clear();
        tTestoForm1.Focus();
    }
    private void bChiudi_Click(object sender, EventArgs e)
    {
        this.Close();
    }
    private void bAnalizza_Click(object sender, EventArgs e)
    {
        if (tTestoForm1.Text == "")
        {
            MessageBox.Show("Inserire il testo da analizzare", "Attenzione");
            return;
        }
        Form2 frm = new Form2(tTestoForm1.Text);
        frm.Show();
    }
}
}

```

Premesso che per semplicità ho preferito non programmare l'evento `bSalva_Click()` (segnalo, però, che nell'Area Download viene proposto uno specifico esercizio Visual C#: "Files di testo in Visual C# .Net"), ci soffermiamo sull'evento `bAnalizza_Click()`

```

private void bAnalizza_Click(object sender, EventArgs e)
{
    if (tTestoForm1.Text == "")
    {
        MessageBox.Show("Inserire il testo da analizzare", "Attenzione");
        return;
    }
    Form2 frm = new Form2(tTestoForm1.Text);
    frm.Show();
}

```

Le due righe evidenziate mostrano come si può passare da un form ad un altro. In questo caso il *Form1* richiama il *Form2* istanziando la classe *Form2* nell' oggetto frm e passandogli come parametro *tTestoForm1.Text*, ovvero il testo inserito in *Form1* e che *Form2* deve analizzare. Anticipo a questo punto che il costruttore della classe *Form2* non è vuoto ma contiene un parametro di tipo string (l'ho chiamato *TestoForm1* ad indicare che è il testo proveniente dal *Form1*) che viene assegnato alla proprietà text della TextBox *tTestoForm2*.

```
public Form2(string TestoForm1)
{
    InitializeComponent();
    tTestoForm2.Text = TestoForm1;
}
```

### Codice Form2

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MultiformVCS
{
    public partial class Form2 : Form
    {
        public Form2(string TestoForm1)
        {
            InitializeComponent();
            tTestoForm2.Text = TestoForm1;
        }

        private void bChiudi_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

```
private void Form2_Load(object sender, EventArgs e)
{
    int lunghezza = tTestoForm2.TextLength;
    int maiuscole = 0, minuscole = 0;
    for (int i = 0; i < lunghezza; i++)
    {
        if (tTestoForm2.Text[i] >= 'a' && tTestoForm2.Text[i] <= 'z')
        {
            minuscole++;
        }
        if (tTestoForm2.Text[i] >= 'A' && tTestoForm2.Text[i] <= 'Z')
        {
            maiuscole++;
        }
    }
    lOutput1.Text = "Numero caratteri: " + lunghezza.ToString();
    lOutput2.Text = "Numero maiuscole: " + maiuscole.ToString();
    lOutput3.Text = "Numero minuscole: " + minuscole.ToString();
}
}
```

[Scarica la soluzione Visual C# con l'approccio "costruttori"](#)

## Soluzione Visual C# alternativa

Come già detto, i metodi per passare dei dati da un form ad un altro sono diversi e l'approccio con i costruttori è forse quello più immediato. Una seconda soluzione per la nostra applicazione che mi viene in mente è quella di **utilizzare una variabile pubblica statica**. L'analisi del codice, in cui sono evidenziate le righe differenti rispetto alla soluzione con i costruttori sopra descritta, ci fa capire subito come funziona.

### Codice Form1

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MultiformVCS
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public static String testoForm1; //uso una variabile pubblica statica che sarà visibile anche nel Form2

        private void bReset_Click(object sender, EventArgs e)
        {
            tTestoForm1.Clear();
            tTestoForm1.Focus();
        }

        private void bChiudi_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

```

private void bAnalizza_Click(object sender, EventArgs e)
{
    if (tTestoForm1.Text == "")
    {
        MessageBox.Show("Inserire il testo da analizzare", "Attenzione");
        return;
    }
    testoForm1 = tTestoForm1.Text; //assegno alla variabile pubblica statica il testo inserito nel Form1
    Form2 frm = new Form2(); // istanzio Form2 con il costruttore vuoto
    frm.Show();
}
}
}

```

## Codice Form2

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MultiformVCS
{
    public partial class Form2 : Form
    {
        public Form2() // costruttore vuoto
        {
            InitializeComponent();
        }

        private void bChiudi_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

```

```
private void Form2_Load(object sender, EventArgs e)
{
    tTestoForm2.Text = Form1.testoForm1;
    //assegno alla TextBox di Form2 il valore della variabile testoForm1 proveniente da Form1
    int lunghezza = tTestoForm2.Text.Length;
    int maiuscole = 0, minuscole = 0;
    for (int i = 0; i < lunghezza; i++)
    {
        if (tTestoForm2.Text[i] >= 'a' && tTestoForm2.Text[i] <= 'z')
        {
            minuscole++;
        }
        if (tTestoForm2.Text[i] >= 'A' && tTestoForm2.Text[i] <= 'Z')
        {
            maiuscole++;
        }
    }
    lOutput1.Text = "Numero caratteri: " + lunghezza.ToString();
    lOutput2.Text = "Numero maiuscole: " + maiuscole.ToString();
    lOutput3.Text = "Numero minuscole: " + minuscole.ToString();
}
}
```

## Soluzione Visual Basic .Net

### Codice Form1

```

Public Class Form1
    Private Sub bReset_Click(sender As System.Object, e As System.EventArgs) Handles bReset.Click
        tTestoForm1.Clear()
        tTestoForm1.Focus()
    End Sub

    Private Sub bChiudi_Click(sender As System.Object, e As System.EventArgs) Handles bChiudi.Click
        Me.Close()
    End Sub

    Private Sub bAnalizza_Click(sender As System.Object, e As System.EventArgs) Handles bAnalizza.Click
        If tTestoForm1.Text = "" Then
            MessageBox.Show("Inserire il testo da analizzare", "Attenzione")
            Return
        End If
        Dim frm As Form2 = New Form2(tTestoForm1.Text)
        frm.Show()
    End Sub
End Class

```

Premesso che per semplicità ho preferito non programmare l'evento bSalva\_Click() (segnalo, però, che nell'Area Download viene proposto uno specifico esercizio Visual Basic .Net " Files di testo in Visual Basic .Net"), ci soffermiamo sull'evento bAnalizza\_Click()

```

Private Sub bAnalizza_Click(sender As System.Object, e As System.EventArgs) Handles bAnalizza.Click
    If tTestoForm1.Text = "" Then
        MessageBox.Show("Inserire il testo da analizzare", "Attenzione")
        Return
    End If
    Dim frm As Form2 = New Form2(tTestoForm1.Text)
    frm.Show()
End Sub

```

Le due righe evidenziate passano il testo inserito nel Form1 al costruttore del Form2. Mentre In Visual C #, il costruttore viene denominato utilizzando il nome della classe, in Visual Basic .Net il costruttore viene definito utilizzando una subroutine denominata **New**. In entrambi i casi, il parametro definisce il testo che viene passato da Form1 a Form2

```
Public Sub New(ByVal value As String)
    InitializeComponent()
    tTestoForm2.Text = value
End Sub
```

## Codice Form2

```
Public Class Form2
    Public Sub New(ByVal value As String)
        InitializeComponent()
        tTestoForm2.Text = value
    End Sub
    Private Sub Form2_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
        Dim lunghezza As Integer = tTestoForm2.TextLength
        Dim maiuscole As Integer = 0
        Dim minuscole As Integer = 0
        Dim s As String = tTestoForm2.Text
        For i As Integer = 0 To lunghezza - 1 Step 1
            If s.Substring(i, 1) >= "a" And s.Substring(i, 1) <= "z" Then
                minuscole += 1
            End If
            If s.Substring(i, 1) >= "A" And s.Substring(i, 1) <= "Z" Then
                maiuscole += 1
            End If
        Next
        lOutput1.Text = "Numero caratteri: " + lunghezza.ToString()
        lOutput2.Text = "Numero maiuscole: " + maiuscole.ToString()
        lOutput3.Text = "Numero minuscole: " + minuscole.ToString()
    End Sub
    Private Sub bChiudi_Click(sender As System.Object, e As System.EventArgs) Handles bChiudi.Click
        Me.Close()
    End Sub
End Class
```

[Scarica la soluzione Visual Basic .Net con l'approccio "costruttori"](#)

## Soluzione Visual Basic .Net alternativa

Una seconda soluzione per la nostra applicazione Visual Basic, in alternativa all'approccio con un costruttore, **utilizza una variabile pubblica**. In Visual Basic .Net una variabile dichiarata Public è accessibile in qualsiasi punto dello stesso progetto. Anche in questo caso la comprensione del funzionamento è immediata

### Codice Form1

Public Class Form1

```
Public testoForm1 As String 'dichiaro una variabile pubblica
```

```
Private Sub bReset_Click(sender As System.Object, e As System.EventArgs) Handles bReset.Click
    tTestoForm1.Clear()
    tTestoForm1.Focus()
End Sub
```

```
Private Sub bChiudi_Click(sender As System.Object, e As System.EventArgs) Handles bChiudi.Click
    Me.Close()
End Sub
```

```
Private Sub bAnalizza_Click(sender As System.Object, e As System.EventArgs) Handles bAnalizza.Click
```

```
    If tTestoForm1.Text = "" Then
        MessageBox.Show("Inserire il testo da analizzare", "Attenzione")
        Return
    End If
```

```
    testoForm1 = tTestoForm1.Text 'assegno alla variabile pubblica il testo inserito nel Form1
    Dim frm As Form2 = New Form2() 'richiamo il Form2 con costruttore vuoto
    frm.Show()
End Sub
```

End Class

## Codice Form2

### Public Class Form2

```
Private Sub Form2_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
    tTestoForm2.Text = Form1.testoForm1
    ' assegno alla TextBox di Form2 il valore della variabile pubblica testoForm1 proveniente da Form1
    Dim lunghezza As Integer = tTestoForm2.Text.Length
    Dim maiuscole As Integer = 0
    Dim minuscole As Integer = 0
    Dim s As String = tTestoForm2.Text
    For i As Integer = 0 To lunghezza - 1 Step 1
        If s.Substring(i, 1) >= "a" And s.Substring(i, 1) <= "z" Then
            minuscole += 1
        End If
        If s.Substring(i, 1) >= "A" And s.Substring(i, 1) <= "Z" Then
            maiuscole += 1
        End If
    Next

    lOutput1.Text = "Numero caratteri: " + lunghezza.ToString()
    lOutput2.Text = "Numero maiuscole: " + maiuscole.ToString()
    lOutput3.Text = "Numero minuscole: " + minuscole.ToString()
End Sub

Private Sub bChiudi_Click(sender As System.Object, e As System.EventArgs) Handles bChiudi.Click
    Me.Close()
End Sub
```

### End Class

Teramo 18.11.2020