

Programmazione Orientata agli Oggetti in Linguaggio Java

Programmazione Grafica: C#

versione 1.0

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons
(vedi ultima pagina)

G. Mecca – Università della Basilicata – mecca@unibas.it



Programmazione Grafica: C# >> Sommario



Sommario

- Panoramica
- Componenti
- Eventi
- Un Esempio



Panoramica

○ Panoramica

- ⇒ il namespace di .NET per la programmazione grafica sono due
- ⇒ System.Windows.Forms
- ⇒ System.Drawing
- ⇒ approccio sostanzialmente simile a quello di Java
- ⇒ con alcune differenze di impostazione dovute alla storia dei due toolkit



Panoramica

○ Analogia

- ⇒ una collezione di componenti
- ⇒ un modello di gestione degli eventi di tipo publish&subscribe con sorgenti disaccoppiate dai gestori

○ Principali differenze

- ⇒ la gestione degli eventi è basata sui delegati
- ⇒ non sono fornite alcune funzionalità di Swing (gestori di layout, gestione del look and feel)



Componenti

○ Principali componenti

- ⇒ `System.Windows.Forms.Form`
- ⇒ `System.Windows.Forms.Button`
- ⇒ `System.Windows.Forms.TextBox`
- ⇒ `System.Windows.Forms.Label`
- ⇒ e molti altri

○ Nota

- ⇒ esistono i pannelli ma sono meno importanti (`System.Windows.Forms.Panel`)



Componenti

○ I componenti

- ⇒ sono contenitori; hanno una collezione `Controls` a cui con il metodo `Add()` è possibile aggiungere i componenti contenuti

○ Il posizionamento: due strategie

- ⇒ I strategia: posizionamento assoluto
- ⇒ Il strategia: ancoraggio (un punto di un componente viene ancorato ad un altro)



Componenti

○ La classe Application

- ⇒ `System.Windows.Forms.Application`
- ⇒ è la classe che viene utilizzata per eseguire l'applicazione avviando la coda degli eventi
- ⇒ attraverso l'esecuzione del metodo statico `Application.Run()`
- ⇒ es: `Application.Run(new MainForm())`



Eventi

○ La gestione degli eventi

- ⇒ è basata sul `publish&subscribe`
- ⇒ ma viene implementata attraverso l'uso di delegati

○ Evento

- ⇒ funzionalità di primo livello del linguaggio
- ⇒ proprietà di una classe il cui tipo corrisponde ad un delegato multicast



Eventi

○ In sintesi

- ⇒ ogni componente ha una serie di proprietà pubbliche di tipo evento
- ⇒ che corrispondono a delegati multicast con un prototipo definito
- ⇒ per sottoscrivere l'evento, è possibile aggiungere nuovi delegati al delegato multicast
- ⇒ in corrispondenza dell'evento verrà eseguito il codice di tutti i delegati



Eventi

○ Sintassi della dichiarazione di un evento

⇒ [`<modificatori>`] event `<tipo>` `<nome>`;

○ dove

- ⇒ `<tipo>` è il nome di un delegato multicast
- ⇒ `<nome>` è il nome della proprietà

○ Esempio

⇒ `public event EventHandler Click;`



Eventi

○ Semantica

- ⇒ viene aggiunto un delegato multicast a ciascun oggetto della classe
- ⇒ il delegato è di tipo System.EventHandler

○ System.EventHandler

- ⇒ il delegato utilizzato per la maggior parte degli eventi
- ⇒ `public delegate void EventHandler(object sender, EventArgs e);`



Eventi

○ Sottoscrizione di un delegato

- ⇒ operatore +=

○ Rimozione di un delegato

- ⇒ operatore -=

○ Esempio

- ⇒ `using System.Windows.Forms;`
- ⇒ `Button bottone = new Button();`
- ⇒ `bottone.Click += new EventHandler(this.Gestisci1);`
- ⇒ `bottone.Click += new EventHandler(this.Gestisci2);`



Un Esempio

>> convertitore

- Un esempio

- ⇒ il convertitore con C#
- ⇒ sviluppato utilizzando l'editor grafico di SharpDevelop
- ⇒ impostazione molto simile a quella di NetBeans



Riassumendo

- Windows Forms
- Componenti
- Eventi
- Un Esempio



Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.