

**Programmazione Orientata
agli Oggetti**

**Il Framework ping
Binding
Parte a**

versione 2.1

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



Il Framework ping: Binding >> Sommario

Sommario

- Terminologia
- Osservatori
- Modello Osservabile

G. Mecca - Programmazione Orientata agli Oggetti 2

Il Framework ping: Binding >> Terminologia

Terminologia

- Il problema del binding
 - ⇒ è un problema centrale della programmazione grafica
 - ⇒ un problema di collegamento (“binding”)
- Binding
 - ⇒ operazione con cui un valore mantenuto nel modello dell’applicazione (nei bean) viene collegato ad un componente grafico per la visualizzazione

G. Mecca - Programmazione Orientata agli Oggetti 3

Il Framework ping: Binding >> Terminologia

Terminologia

- In effetti
 - ⇒ a ben guardare la maggior parte del codice della vista è dedicato alle operazioni di binding
- Esempi
 - ⇒ le JLabel/JTextField che visualizzano stringhe dei bean
 - ⇒ le JList/JTable che visualizzano collezioni del modello

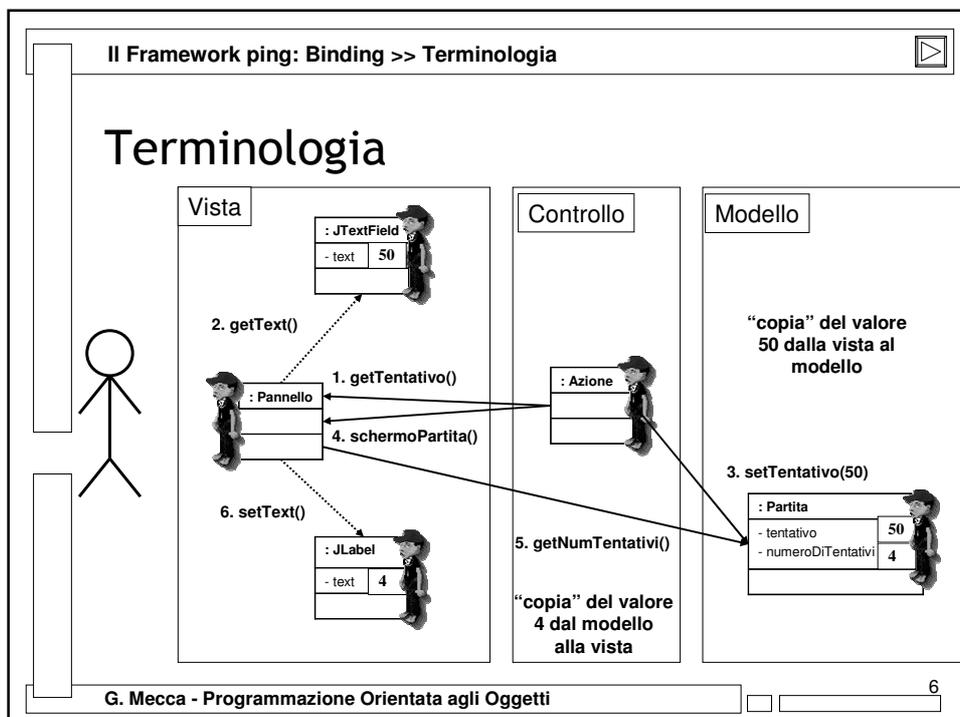
G. Mecca - Programmazione Orientata agli Oggetti 4

Il Framework ping: Binding >> Terminologia

Terminologia

- L'approccio che abbiamo usato finora
 - ⇒ copia esplicita dei valori
- Binding per copia esplicita
 - ⇒ l'azione preleva dati dal componente (es: tentativo dell'utente)
 - ⇒ aggiorna il modello
 - ⇒ al termine la vista aggiorna i componenti (es: numero di tentativi)

G. Mecca - Programmazione Orientata agli Oggetti 5



Il Framework ping: Binding >> Terminologia

Terminologia

ATTENZIONE
al problema del binding

- Vantaggio di questo approccio
 - ⇒ semplice concettualmente
- Svantaggio di questo approccio
 - ⇒ il codice della vista si allunga notevolmente per tenere “sincronizzati” i componenti
 - ⇒ molto scomodo se più componenti visualizzano le stesse proprietà (es: barra di scorrimento per i tentativi) – copie multiple
 - ⇒ nelle applicazioni complesse è un problema

G. Mecca - Programmazione Orientata agli Oggetti 7

Il Framework ping: Binding >> Terminologia

Terminologia

- Binding automatico
 - ⇒ operazione con cui un valore mantenuto nel modello dell'applicazione (nei bean) viene collegato ad un componente grafico per la visualizzazione
- Ne esistono due forme
 - ⇒ binding monodirezionale
 - ⇒ binding bidirezionale

G. Mecca - Programmazione Orientata agli Oggetti 8

Il Framework ping: Binding >> Terminologia

Terminologia

- Binding automatico monodirezionale
 - ⇒ il componente della vista riflette automaticamente le variazioni di stato del componente del modello
 - ⇒ è sempre sicuro (i valori provenienti dal modello sono sempre corretti)
 - ⇒ richiede di costruire un'infrastruttura per segnalare al componente della vista le variazioni di valore nel modello

G. Mecca - Programmazione Orientata agli Oggetti 9

Il Framework ping: Binding >> Terminologia

Terminologia

- L'altro tipo di binding
 - ⇒ riguarda gli schermi che contengono form o wizard
- Maschera ("Form")
 - ⇒ collezione di componenti contenuti in uno schermo per l'acquisizione di dati dell'utente
 - ⇒ tipicamente dotata di un bottone di invio ("bottone di commit")
 - ⇒ e di uno per annullare ("bottone di rollback")

G. Mecca - Programmazione Orientata agli Oggetti 10

Il Framework ping: Binding >> Terminologia

Terminologia

- Procedura Guidata (“Wizard”)
 - ⇒ gruppo di maschere organizzate in sequenza per guidare l’utente in una operazione di immissione che richiede vari passi
 - ⇒ le maschere che partecipano ad un wizard tipicamente hanno un bottone per muoversi al passo successivo (“bottone avanti”)
 - ⇒ e un bottone per tornare al passo precedente (“bottone indietro”)

G. Mecca - Programmazione Orientata agli Oggetti 11

Il Framework ping: Binding >> Terminologia

Terminologia

- Binding automatico bidirezionale
 - ⇒ riguarda i componenti delle form (campi di immissione)
 - ⇒ ciascun componente riflette automaticamente le variazioni di stato dell’altro (vista->modello e modello->vista)
 - ⇒ è utilizzabile solo assieme ad un meccanismo di convalida automatica: non sempre i dati della vista devono essere salvati nel modello

G. Mecca - Programmazione Orientata agli Oggetti 12

Il Framework ping: Binding >> Terminologia

Terminologia

- In queste lezioni
 - ⇒ adatteremo la seguente terminologia
- Osservatore
 - ⇒ componente capace di realizzare binding monodirezionale (dal modello alla vista)
 - ⇒ “osserva” le variazioni di una proprietà del modello e le traduce nel valore di un componente della vista

G. Mecca - Programmazione Orientata agli Oggetti 13

Il Framework ping: Binding >> Terminologia

Terminologia

- Nota
 - ⇒ il termine “osservatore” viene usato per similarità con il termine sensore/ascoltatore (“listener”) usato per i gestori di eventi
 - ⇒ e assonanza con il termine “vista”
 - ⇒ i bean del modello si dicono oggetti “osservabili” perchè le loro proprietà vengono osservate dagli osservatori

G. Mecca - Programmazione Orientata agli Oggetti 14

Il Framework ping: Binding >> Terminologia

Terminologia

- Collegatore
 - ⇒ componente capace di realizzare binding dalla vista al modello
 - ⇒ è associabile solo a campi di immissione di una form
 - ⇒ “copia” il valore fornito dall’utente in una proprietà di un bean del modello
 - ⇒ se è associato ad un osservatore realizza un binding bidirezionale (da cui il nome)

G. Mecca - Programmazione Orientata agli Oggetti 15

Il Framework ping: Binding >> Terminologia

Terminologia

- Convalidatore
 - ⇒ componente associato ad un campo di immissione di una form
 - ⇒ verifica che il valore sia corretto prima che questo venga copiato nel modello dal collegatore
 - ⇒ normalmente prima della verifica deve effettuare una conversione del valore prelevato dal componente (es: String -> int)

G. Mecca - Programmazione Orientata agli Oggetti 16

Il Framework ping: Binding >> Osservatori

Osservatori

ATTENZIONE
all'uso degli
eventi nel modello

- L'idea alla base degli osservatori
 - ⇒ utilizzare lo stesso meccanismo di publish&subscribe utilizzato per la gestione degli eventi scatenati dai gesti dell'utente
- Differenza
 - ⇒ gli eventi che verranno scatenati e gestiti, però, saranno di tipo diverso
 - ⇒ si tratterà di eventi originati dal modello

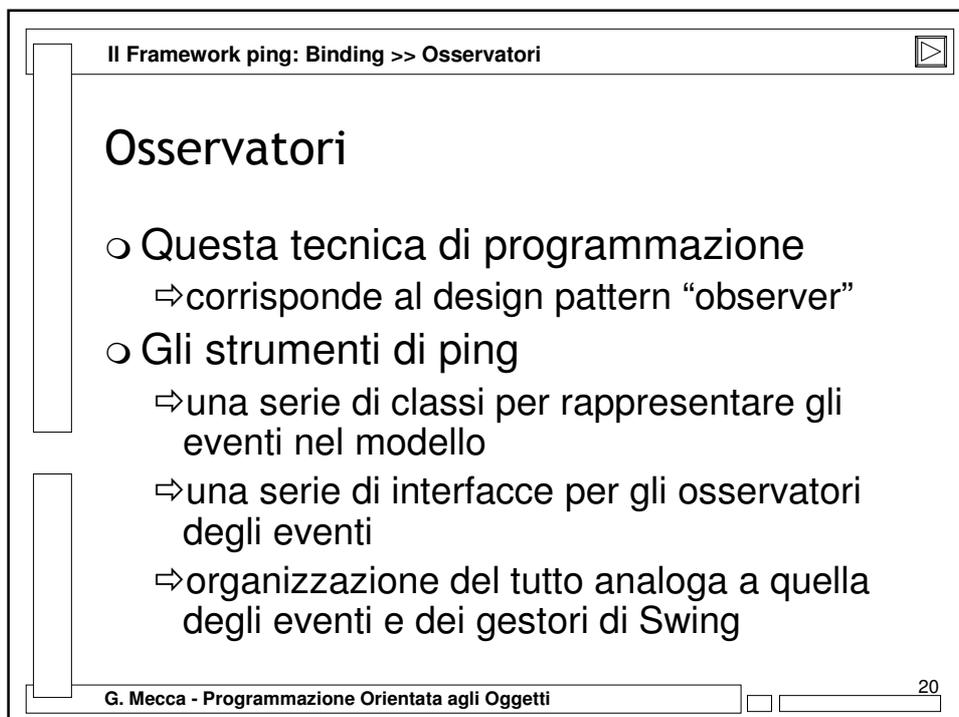
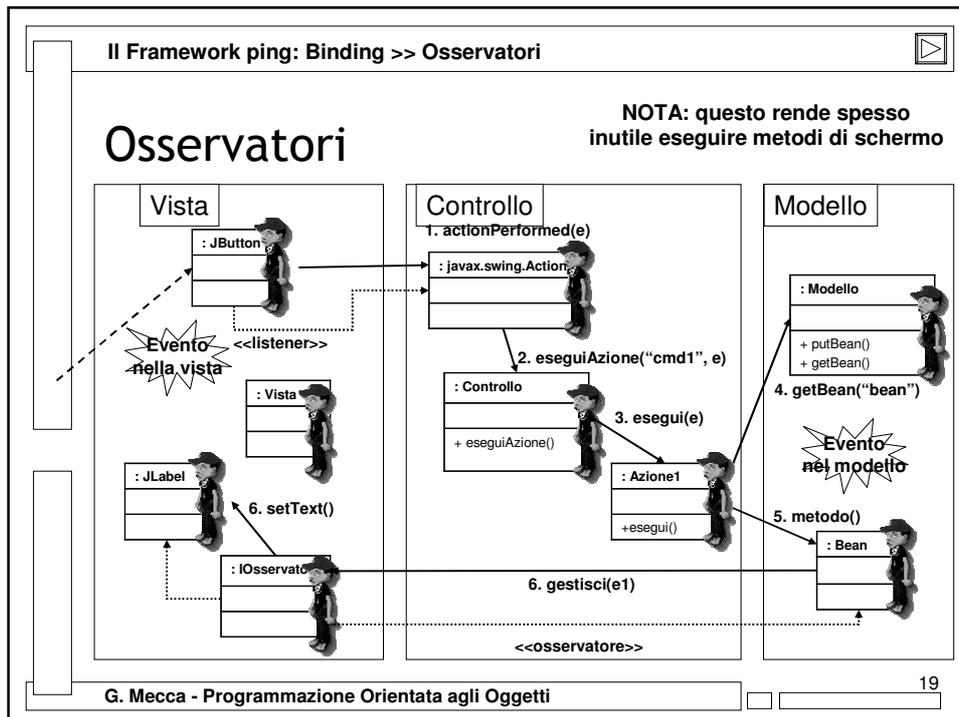
G. Mecca - Programmazione Orientata agli Oggetti 17

Il Framework ping: Binding >> Osservatori

Osservatori

- Funzionamento degli osservatori
 - ⇒ gli osservatori sono gestori di eventi lanciati dai bean del modello
 - ⇒ ogni volta che un proprietà del bean viene aggiornata, il bean notifica ai propri osservatori un evento di modifica
 - ⇒ gli osservatori gestiscono l'evento aggiornando il valore del componente nella vista

G. Mecca - Programmazione Orientata agli Oggetti 18



Il Framework ping: Binding >> Osservatori

Osservatori

ATTENZIONE
al publish&subscribe
nei JavaBeans

- Attenzione
 - ⇒ la piattaforma Java fornisce una infrastruttura già pronta per la gestione di eventi nel modello, collegata allo standard JavaBeans
 - ⇒ infatti, ciascun JavaBean può essere sorgente di eventi di tipo `java.beans.PropertyChangeEvent`
 - ⇒ i gestori devono implementare l'interfaccia `java.beans.PropertyChangeListener`

G. Mecca - Programmazione Orientata agli Oggetti 21

Il Framework ping: Binding >> Osservatori

Osservatori

- Nonostante questo
 - ⇒ nel framework l'infrastruttura è stata completamente rivisitata per varie ragioni
 - ⇒ mantenere l'API del framework completamente in italiano
 - ⇒ consentire agli studenti di consultare facilmente, studiare ed eventualmente modificare il codice sorgente
 - ⇒ implementare alcune caratteristiche originali

G. Mecca - Programmazione Orientata agli Oggetti 22

Il Framework ping: Binding >> Osservatori

Osservatori

- La gerarchia degli eventi

tipi di evento per le collezioni:
 Modello.AGGIUNTA
 Modello.ELIMINAZIONE
 Modello.MODIFICA

```

    graph BT
        EO["java.util.EventObject  
Object getSource()"]
        ENM["it.unibas.ping.eventi.EventoNelModello  
String getSorgente()"]
        EMP["it.unibas.ping.eventi.EventoModificaProprietà  
String getNomeProprietà()"]
        EMC["it.unibas.ping.eventi.EventoModificaCollezione  
int getTipo()  
int getIndiceIniziale()  
int getIndiceFinale()"]
        EMO["it.unibas.ping.eventi.EventoModificaCella  
int getRiga()  
int getColonna()"]

        ENM --> EO
        EMP --> ENM
        EMC --> EMP
        EMO --> EMP
    
```

G. Mecca - Programmazione Orientata agli Oggetti 23

Il Framework ping: Binding >> Osservatori

Osservatori

- La gerarchia dei listener

<<interface>> IOsservatore
 Object getComponente()
 void setComponente(Object componente)
 String getNomeBean()
 void setNomeBean(String nomeBean)
 String getNomeProprietà()
 void setNomeProprietà(String nomeProprietà)
 void registra()
 void ripulisci()

```

    graph BT
        EL["java.util.EventListener"]
        IO["it.unibas.ping.eventi.IOsservatore"]
        OPC["it.unibas.ping.eventi.OsservatoreProprietà  
void gestisci(EventoModificaProprietà e)"]
        OCC["it.unibas.ping.eventi.OsservatoreCollezione  
void gestisci(EventoModificaCollezione e)"]
        OMC["it.unibas.ping.eventi.OsservatoreCella  
void gestisci(EventoModificaCella e)"]

        IO --> EL
        OPC --> IO
        OCC --> IO
        OMC --> IO
    
```

G. Mecca - Programmazione Orientata agli Oggetti 24

Il Framework ping: Binding >> Osservatori

Osservatori

- Le implementazioni
 - ⇒ il framework fornisce osservatori di proprietà per la maggior parte dei componenti grafici
 - ⇒ OsservatoreLabel
 - ⇒ OsservatoreTextField
 - ⇒ OsservatoreIcona
 - ⇒ OsservatoreCheckBox
 - ⇒ OsservatoreComboBox
 - ⇒ OsservatoreButtonGroup

G. Mecca - Programmazione Orientata agli Oggetti 25

Il Framework ping: Binding >> Osservatori

Osservatori

- Inoltre
 - ⇒ fornisce due osservatori di collezione
 - ⇒ OsservatoreLista
 - ⇒ OsservatoreTabellaMono
- Infine
 - ⇒ un osservatore di celle
 - ⇒ OsservatoreTabellaBidim

G. Mecca - Programmazione Orientata agli Oggetti 26

Il Framework ping: Binding >> Osservatori

Osservatori

- L'ultimo componente necessario
 - ⇒ una classe che fornisca tutte le funzionalità per la sottoscrizione e la segnalazione degli eventi
 - ⇒ liste di sottoscrittori
 - ⇒ metodi per notificare gli eventi
- Nel framework
 - ⇒ questo ruolo viene svolto dalla classe Osservabile

G. Mecca - Programmazione Orientata agli Oggetti 27

Il Framework ping: Binding >> Osservatori

Osservatori

- L'interfaccia di Osservabile


```
void addOsservatoreProprieta(IOsservatoreProprieta osservatore)
void addOsservatoreCollezione(IOsservatoreCollezione osservatore)
void addOsservatoreCella(IOsservatoreCella osservatore)

void removeOsservatoreProprieta(IOsservatoreProprieta osservatore)
void removeOsservatoreCollezione(IOsservatoreCollezione osservatore)
void removeOsservatoreCella(IOsservatoreCella osservatore)

void notificaModificaProprieta(Object bean, String proprieta)
void notificaModificaCollezione(Object bean, String proprieta, int indice, int tipo)
void notificaModificaCollezione(Object bean, String prop, int inizio, int fine, int tipo)
void notificaModificaCella(Object bean, String proprieta, int riga, int colonna)
```

G. Mecca - Programmazione Orientata agli Oggetti 28

Il Framework ping: Binding >> Osservatori

Osservatori

- Per utilizzare un osservatore
 - ⇒ è necessario creare l'osservatore
 - ⇒ specificare il componente grafico
 - ⇒ specificare il bean e la proprietà osservata
- Successivamente
 - ⇒ nei metodi "setter" del bean, sarà necessario lanciare eventi per segnalare all'osservatore le modifiche

G. Mecca - Programmazione Orientata agli Oggetti 29

Il Framework ping: Binding >> Osservatori

Osservatori

- Tipicamente (es: JavaBeans)
 - ⇒ l'osservatore lavora direttamente in associazione con il bean (ne possiede il riferimento)
 - ⇒ in altri termini il bean estende Osservabile oppure lavora in associazione con un oggetto di tipo Osservabile e ne riproduce l'interfaccia
 - ⇒ per sottoscrivere gli eventi, gli osservatori si registrano nelle liste del bean

G. Mecca - Programmazione Orientata agli Oggetti 30

Il Framework ping: Binding >> Osservatori

Osservatori

- Questo fatto ha degli svantaggi
 - ⇒ in particolare, non consente di gestire facilmente il caso in cui una istanza del bean viene sostituita con un'altra (è necessario registrare tutti gli osservatori del primo bean nel secondo)
 - ⇒ più in generale, non consente di gestire tutte le fasi del ciclo di vita del bean, in quanto l'osservatore può esistere solo se il bean esiste

G. Mecca - Programmazione Orientata agli Oggetti 31

Il Framework ping: Binding >> Modello Osservabile

Modello Osservabile

- Nel framework
 - ⇒ viene utilizzato un altro approccio
 - ⇒ basato su un pattern che potremmo definire "modello osservabile"
- Idea
 - ⇒ l'osservatore è completamente disaccoppiato dal bean
 - ⇒ è il Modello a svolgere il ruolo di intermediazione tra l'uno e l'altro

G. Mecca - Programmazione Orientata agli Oggetti 32

Il Framework ping: Binding >> Modello Osservabile

Modello Osservabile

- Regole del framework per gli osservatori
 - ⇒ l'oggetto modello è l'unico osservabile dell'applicazione
 - ⇒ gli osservatori vengono registrati come sottoscrittori nel modello
 - ⇒ il riferimento al bean a cui sono interessati viene specificato attraverso una stringa, che corrisponde alla chiave con cui il bean viene salvato nella mappa del modello

G. Mecca - Programmazione Orientata agli Oggetti 33

Il Framework ping: Binding >> Modello Osservabile

Modello Osservabile

- Regole del framework (continua)
 - ⇒ ogni volta che viene modificata una proprietà osservata di un bean, è necessario effettuare due operazioni
 - ⇒ creare un oggetto di tipo EventoNelModello che contiene i dati rilevanti
 - ⇒ chiedere al modello di notificare l'evento agli osservatori

G. Mecca - Programmazione Orientata agli Oggetti 34

Il Framework ping: Binding >> Modello Osservabile

Modello Osservabile

- Regole del framework (continua)
 - ⇒ il modello segnala l'evento a tutti gli osservatori del bean che ha generato l'evento
 - ⇒ ciascun osservatore riceve la notifica e verifica la proprietà modificata
 - ⇒ se la proprietà è rilevante, l'osservatore acquisisce dal modello il riferimento al bean
 - ⇒ preleva il nuovo valore della proprietà eseguendo il metodo get del bean
 - ⇒ e modifica il valore del componente grafico

G. Mecca - Programmazione Orientata agli Oggetti 35

Il Framework ping: Binding >> Modello Osservabile

Modello Osservabile

- Vantaggi del disaccoppiamento
 - ⇒ è possibile gestire una serie di casi particolari del ciclo di vita bean/osservatore
 - ⇒ questi casi particolari si riferiscono alla circostanza in cui esistono nell'applicazione osservatori per bean che non sono stati ancora creati e aggiunti al modello
 - ⇒ oppure esistevano nel modello e sono stati rimossi

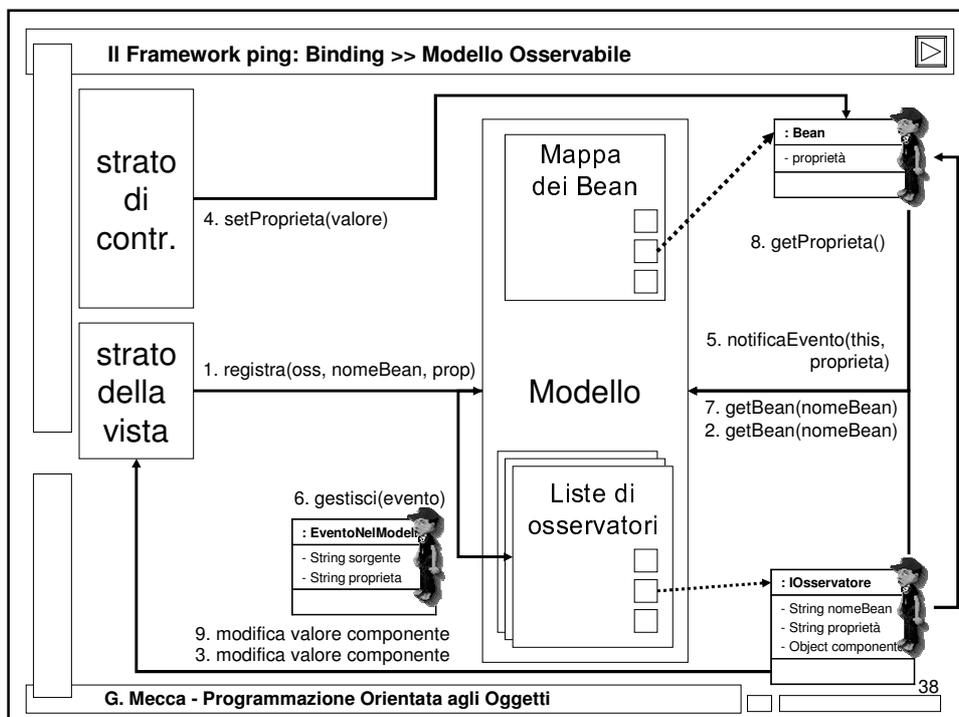
G. Mecca - Programmazione Orientata agli Oggetti 36

Il Framework ping: Binding >> Modello Osservabile

Modello Osservabile

- Caso ordinario
 - ⇒ quando l'osservatore viene registrato, il bean esiste nel modello
 - ⇒ alla registrazione l'osservatore inizializza il valore del componente prelevando il valore della proprietà dal bean
 - ⇒ successivamente il bean notifica eventi che l'osservatore gestisce prelevando i valori modificati e aggiornando il componente

G. Mecca - Programmazione Orientata agli Oggetti 37

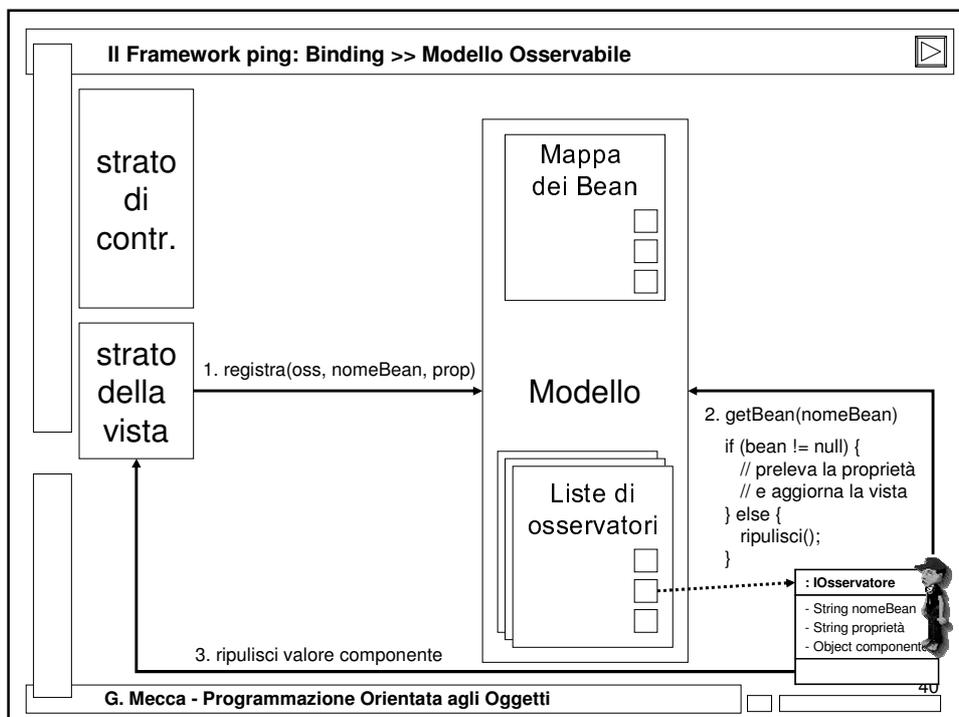


Il Framework ping: Binding >> Modello Osservabile

Modello Osservabile

- Caso particolare n. 1: bean nullo
 - ⇒ alla registrazione dell'osservatore, il bean non esiste nel modello
 - ⇒ nonostante il bean non esista, l'osservatore riesce comunque a lavorare
 - ⇒ in particolare, in questo caso l'osservatore inizializza il componente grafico attribuendogli il valore standard (es: stringa vuota o selezione predefinita)

G. Mecca - Programmazione Orientata agli Oggetti 39

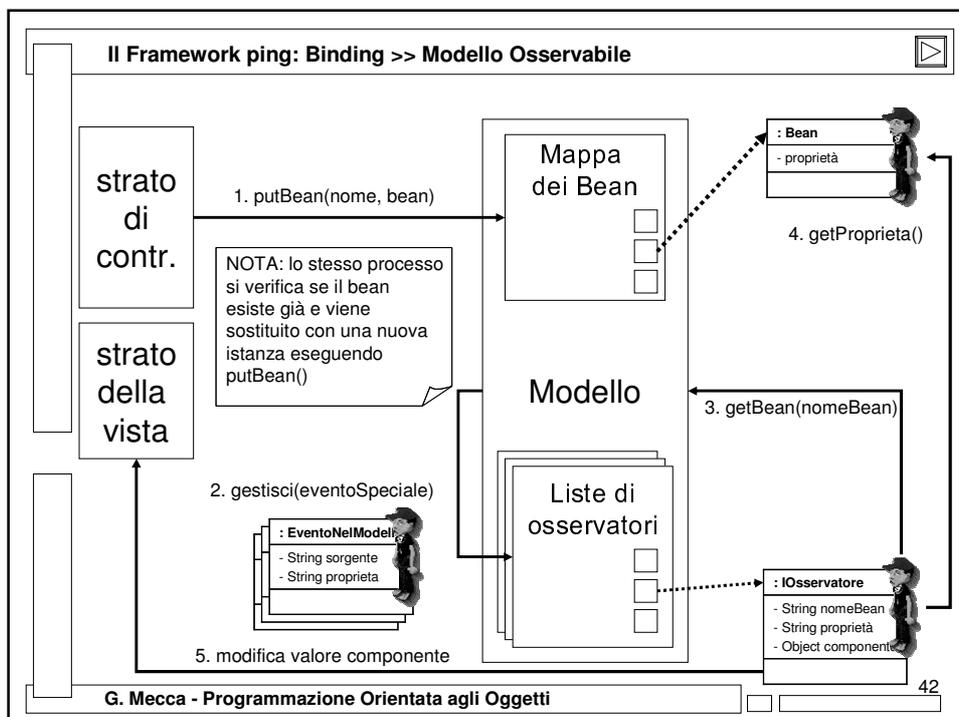


Il Framework ping: Binding >> Modello Osservabile

Modello Osservabile

- Caso particolare n. 2: putBean
 - ⇒ il bean viene creato nel modello quando esistono già osservatori per quel bean
 - ⇒ in questo caso il modello notifica agli osservatori una serie di eventi speciali per modificare la vista
 - ⇒ un evento di modifica per la proprietà Modello.TUTTE_LE_PROPRIETA
 - ⇒ un evento di modifica collezione per l'intervallo 0, Modello.ULTIMO_INDICE
 - ⇒ un evento di modifica celle per Modello.TUTTE_LE_CELLE

G. Mecca - Programmazione Orientata agli Oggetti 41

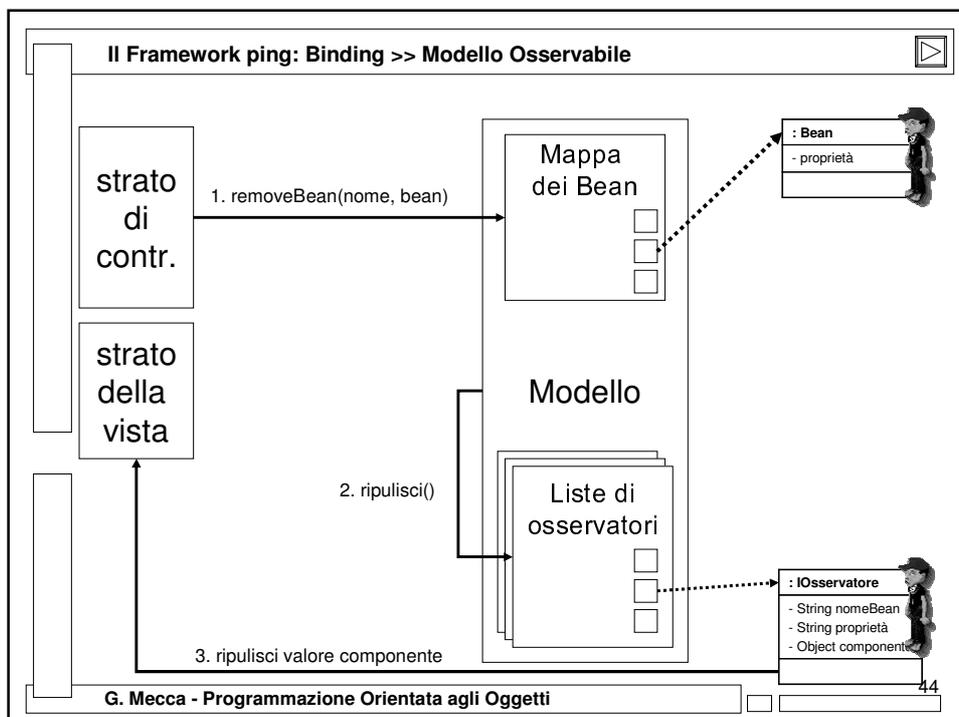


Il Framework ping: Binding >> Modello Osservabile

Modello Osservabile

- Caso particolare n. 3: removeBean
 - ⇒ in questo caso, il modello segnala a tutti gli osservatori del bean di “ripulire” il componente della vista in modo da riportarlo al valore iniziale
 - ⇒ questo per evitare che il componente della vista rispecchi il valore di un bean che non esiste più

G. Mecca - Programmazione Orientata agli Oggetti 43



Il Framework ping: Binding >> Modello Osservabile

Modello Osservabile

- A questo punto
 - ⇒ chiarito i concetti alla base, possiamo vedere come materialmente sviluppare il binding
- Ci sono due attività principali
 - ⇒ creare gli osservatori nella vista, associandoli ai componenti grafici e alle proprietà dei bean
 - ⇒ richiedere la notifica di eventi quando vengono modificate le proprietà dei bean

G. Mecca - Programmazione Orientata agli Oggetti 45

Il Framework ping: Binding >> Sommario

Riassumendo

- Terminologia
- Osservatori
- Modello Osservabile

G. Mecca - Programmazione Orientata agli Oggetti 46

Termini della Licenza 

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.

G. Mecca - Programmazione Orientata agli Oggetti  47