

**Programmazione Orientata  
agli Oggetti**

**Il Framework ping  
Elementi di Base**

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: Elementi di Base >> Sommario

## Sommario

- Componenti Fondamentali
- Applicazione
- Il File di Configurazione
- Controllo e Azioni
- Gerarchia di Viste

G. Mecca - Programmazione Orientata agli Oggetti 2

Il Framework ping: Elementi di Base >> Componenti Fondamentali

## Componenti Fondamentali

- I componenti fondamentali di ping
  - ⇒ sono contenuti nel package `it.unibas.ping.framework`
  - ⇒ la classe Applicazione
  - ⇒ la classe Controllo
  - ⇒ la classe Modello
  - ⇒ le classi Vista


G. Mecca - Programmazione Orientata agli Oggetti 3

Il Framework ping: Elementi di Base >> Applicazione

## Applicazione


- La classe Applicazione
  - ⇒ componente unico con visibilità globale ("singleton")
  - ⇒ funge da punto di accesso ai componenti
- Attenzione
  - ⇒ la classe Applicazione contiene il metodo `main()` e avvia l'applicazione
  - ⇒ caratteristica tipica dei framework (il metodo `main()` non viene scritto dall'utente)


G. Mecca - Programmazione Orientata agli Oggetti 4

Il Framework ping: Elementi di Base >> Applicazione 

## Applicazione


- Punto di accesso all'infrastruttura
  - ⇒ per ottenere il riferimento all'oggetto Applicazione: `Applicazione.getInstance()`
  - ⇒ consente l'accesso alle classi dell'infrastruttura MVC attraverso i metodi `getControllo()`, `getModello()`, `getVista()`
  - ⇒ che restituiscono rispettivamente un riferimento al Controllo, al Modello e alla Vista

G. Mecca - Programmazione Orientata agli Oggetti  5

Il Framework ping: Elementi di Base >> Il File di Configurazione 

## Il File di Configurazione

- ping-config.xml
  - ⇒ file .xml che contiene una descrizione dichiarativa di molti aspetti dell'applicazione
  - ⇒ valido rispetto ad un opportuno DTD `ping1.0.dtd`
  - ⇒ deve essere raggiungibile attraverso il classpath come `/ping-config.xml`
  - ⇒ viene caricato all'avvio dalla classe `Applicazione`

G. Mecca - Programmazione Orientata agli Oggetti  6

Il Framework ping: Elementi di Base >> Il File di Configurazione

## Il File di Configurazione

- Nota
  - ⇒ se all'avvio l'oggetto applicazione non riesce a completare la configurazione si verifica un'eccezione
- PingException
  - ⇒ eccezione sollevata dal framework per segnalare malfunzionamenti
  - ⇒ è un'eccezione NON controllata, che corrisponde sempre a condizioni irrimediabili

G. Mecca - Programmazione Orientata agli Oggetti 7

Il Framework ping: Elementi di Base >> Il File di Configurazione

## Il File di Configurazione

- Contenuto della configurazione
  - ⇒ elenco delle viste (per consentire l'inizializzazione automatica della mappa delle viste)
  - ⇒ elenco delle azioni, con i relativi attributi (opzionali)
  - ⇒ idea: questi dati vengono "descritti" al framework lasciando che faccia lui parte del lavoro relativo

G. Mecca - Programmazione Orientata agli Oggetti 8

Il Framework ping: Elementi di Base >> Il File di Configurazione

## Il File di Configurazione

- Struttura del file
  - ⇒ elemento principale <pingConfig> consente di fornire informazioni sull'applicazione
  - ⇒ elemento <viste>: attributo vistaPrincipale ed elenco delle viste (per ciascuna un elemento < sottoVista> con attributo id)
  - ⇒ elemento <azioni>: elenco delle azioni (per ciascuna un elemento <azione> con id e attributi (nome, descrizione, mnemonico, acceleratore, icona – opzionali)

G. Mecca - Programmazione Orientata agli Oggetti 9

Il Framework ping: Elementi di Base >> Il File di Configurazione

## Il File di Configurazione

>> indovinaIlNumero: ping1.0.dtd  
>> indovinaIlNumero: ping-config.xml

- Una caratteristica di ping
  - ⇒ gli id dei componenti coincidono con i nomi delle classi
  - ⇒ obiettivo: favorire il refactoring (>>)
  - ⇒ sia per le viste che per le azioni è possibile specificare un package comune; l'id viene ottenuto concatenando il package comune con il valore dell'attributo id

G. Mecca - Programmazione Orientata agli Oggetti 10

Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

- Una caratteristica fondamentale di ping
  - ⇒ concetto generalizzato di azione applicativa
  - ⇒ distinzione tra azione e listener
- Lo strato di controllo
  - ⇒ è fatto di azioni “ping”
  - ⇒ che NON sono oggetti di tipo `javax.swing.Action`
  - ⇒ nè listener secondo la gerarchia di Java

G. Mecca - Programmazione Orientata agli Oggetti 11

Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

- Azioni di ping
  - ⇒ oggetti di tipo `it.unibas.ping.azioni.IAzionePing`
  - ⇒ il metodo che esegue l'azione applicativa è `public void esegui(java.util.EventObject evento)`
  - ⇒ ogni azione ha un id, che corrisponde al nome (completamente qualificato) della classe corrispondente
- Id dell'azione = Comando
  - ⇒ stringa che identifica l'azione

G. Mecca - Programmazione Orientata agli Oggetti 12

Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

- Normalmente le azioni dell'utente
  - ⇒ estendono la classe astratta  
it.unibas.ping.azioni.AzionePingAstratta
  - ⇒ sovrascrivendo il metodo esegui()
  - ⇒ in questo modo ereditano una serie di caratteristiche
  - ⇒ tra cui, in particolare, un riferimento controllo, un riferimento modello e un riferimento vista (>>)

G. Mecca - Programmazione Orientata agli Oggetti 13

Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

**ATTENZIONE**  
alla differenza tra  
azioni ping e listener

- Di conseguenza
  - ⇒ le azioni di ping NON sono listener
  - ⇒ le azioni di ping non sono specifiche di un tipo di evento (es: ActionEvent)
  - ⇒ sono eseguibili in corrispondenza di ogni tipo di evento
  - ⇒ attenzione: per essere utilizzate devono essere associate ad un listener

G. Mecca - Programmazione Orientata agli Oggetti 14

Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

- Utilizzo delle azioni
  - ⇒ lo sviluppatore scrive il codice delle azioni ping
  - ⇒ per poterle utilizzare, durante lo sviluppo della vista deve associare ai componenti della vista opportuni listener (Action, SelectionListener, ItemListener ecc.)
  - ⇒ questi listener devono gestire gli eventi corrispondenti eseguendo ciascuno un'azione opportuna

G. Mecca - Programmazione Orientata agli Oggetti 15

Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

- Listener di ping
  - ⇒ oggetti che fungono esclusivamente da tramite tra i componenti grafici e il Controllo
  - ⇒ ogni listener è associato ad un'azione di ping
  - ⇒ l'unica funzione del listener è chiedere al Controllo l'esecuzione dell'azione ping corrispondente
  - ⇒ ovvero inviare al Controllo un "comando" (che corrisponde all'id dell'azione)

G. Mecca - Programmazione Orientata agli Oggetti 16



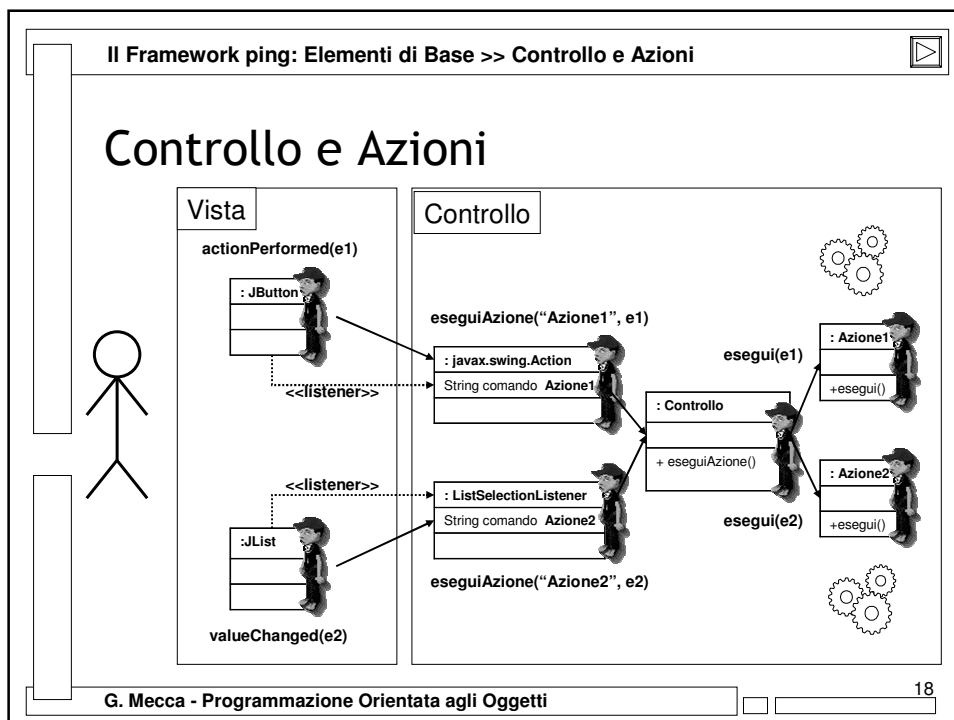
Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

- La classe Controllo
  - ⇒ mantiene la mappa delle azioni ping
  - ⇒ è in grado di fabbricare listener di vario tipo che “incapsulano” azioni applicative
- Metodi del Controllo
 

```
javax.swing.Action getAzioneSwing(String idAzione)
AdjustmentListener getAdjustmentListener(String idAzione)
ChangeListener getChangeListener(String idAzione)
ListSelectionListener getListSelectionListener(String idAzione)
...
```

G. Mecca - Programmazione Orientata agli Oggetti 17



Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

- Vantaggi di questo approccio
  - ⇒ le azioni di ping sono gestori di eventi "generici"
  - ⇒ totalmente disaccoppiati dalla vista
  - ⇒ non sono direttamente sottoscrittori dei componenti grafici
  - ⇒ consentono di gestire un'ampia gamma di eventi (potenzialmente anche eventi che non hanno origine nell'interfaccia grafica)

G. Mecca - Programmazione Orientata agli Oggetti 19

Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

- Azioni di Swing
  - ⇒ in tutti i casi in cui un'azione applicativa deve essere utilizzata nell'applicazione come azione Swing è possibile specificarne le proprietà nel file di configurazione
  - ⇒ nome, descrizione, mnemonico, acceleratore, icona
  - ⇒ il Controllo si occupa di configurare l'oggetto di tipo `javax.swing.Action` impostando i parametri specificati

G. Mecca - Programmazione Orientata agli Oggetti 20

Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

- Il processo di sviluppo
  - ⇒ lo sviluppatore scrive il codice delle azioni ping
  - ⇒ nel caso in cui le azioni debbano essere usate come azioni Swing, definisce le proprietà nel file di configurazione
  - ⇒ nella creazione della vista, registra nei componenti grafici i listener opportuni associati alle azioni da eseguire

G. Mecca - Programmazione Orientata agli Oggetti 21


Il Framework ping: Elementi di Base >> Controllo e Azioni

## Controllo e Azioni

- >> indovinaIlNumero: ping-config.xml
- >> indovinaIlNumero: vista
- >> indovinaIlNumero: controllo

- Il processo (continua)
  - ⇒ i listener sono ottenuti dall'oggetto controllo
  - ⇒ da quel momento in poi, ogni gesto dell'utente scatena un evento, che viene notificato ad un listener, che richiede al Controllo l'esecuzione di un'azione
  - ⇒ viene eseguita l'azione ping corrispondente

G. Mecca - Programmazione Orientata agli Oggetti 22


Il Framework ping: Elementi di Base >> Controllo e Azioni 

## Controllo e Azioni

>> indovinaIlNumero: ping-config.xml

- Esistono due azioni particolari
- L'azione iniziale
  - ⇒ oggetto di tipo `it.unibas.ping.azioni.IAzioneIniziale`
  - ⇒ eseguita alla costruzione del Controllo – utile per disabilitare i componenti nello schermo di avvio
- L'azione finale
  - ⇒ oggetto di tipo `it.unibas.ping.azioni.AzioneFinale`:
  - ⇒ eseguita alla distruzione del Controllo
  - ⇒ sono entrambe opzionali


G. Mecca - Programmazione Orientata agli Oggetti 23

Il Framework ping: Elementi di Base >> Controllo e Azioni 

## Controllo e Azioni

- Nota
  - ⇒ nella maggior parte dei casi l'azione iniziale può essere evitata utilizzando l'attributo "disabilitataAllAvvio" dell'elemento `azioneSwing`
  - ⇒ consente di disabilitare alla creazione una azione Swing

G. Mecca - Programmazione Orientata agli Oggetti 24


Il Framework ping: Elementi di Base >> Controllo e Azioni 

## Controllo e Azioni

>> indovinaIlNumero: azioni predefinite

- Le azioni predefinite di ping
  - ⇒ ping fornisce una serie di azioni già pronte, che corrispondono a compiti standard e possono essere utilizzate senza doverle ridefinire
  - ⇒ it.unibas.ping.azioni.AzioneEsci: System.exit()
  - ⇒ it.unibas.ping.azioni.AzioneInformazioniApplicazione: finestra di dialogo che descrive l'applicazione
  - ⇒ it.unibas.ping.azioni.AzioneInformazioniPing: finestra di dialogo che descrive il framework

G. Mecca - Programmazione Orientata agli Oggetti 25

Il Framework ping: Elementi di Base >> Gerarchia di Viste 

## Gerarchia di Viste

- Le viste di ping
  - ⇒ devono essere oggetti di tipo it.unibas.ping.framework.ISottoVista
- Il metodo fondamentale
  - ⇒ void inizializza()
  - ⇒ il programmatore deve sovrascriverlo in ciascuna vista per disporre i componenti

G. Mecca - Programmazione Orientata agli Oggetti 26

Il Framework ping: Elementi di Base >> Gerarchia di Viste

## Gerarchia di Viste

- Esistono tre tipi di viste
  - ⇒ classi astratte che è possibile estendere
  - ⇒ `it.unibas.ping.framework.FramePing`: frame principale, estende `JFrame`
  - ⇒ `it.unibas.ping.framework.PannelloPing`: pannello, estende `JPanel`
  - ⇒ `it.unibas.ping.framework.FinestraDiDialogoPing`: finestra secondaria, estende `JDialog`
  - ⇒ forniscono riferimenti a modello e controllo


G. Mecca - Programmazione Orientata agli Oggetti 27

Il Framework ping: Elementi di Base >> Gerarchia di Viste

## Gerarchia di Viste


- Creazione della gerarchia di viste
  - ⇒ nel file di configurazione viene descritta l'id della vista principale (di tipo `FramePing`)
  - ⇒ e l'elenco degli id delle sottoviste
- La classe Vista
  - ⇒ si incarica di inizializzare tutte le viste
  - ⇒ e di creare e mantenere la mappa delle viste dell'applicazione


G. Mecca - Programmazione Orientata agli Oggetti 28

Il Framework ping: Elementi di Base >> Gerarchia di Viste 

## Gerarchia di Viste


- Una caratteristica importante di ping
  - ⇒ la classe Vista è in grado di navigare automaticamente le sottoviste
- Schermo di ping
  - ⇒ un metodo di una qualsiasi delle sottoviste
  - ⇒ con tipo di ritorno void
  - ⇒ e senza parametri


G. Mecca - Programmazione Orientata agli Oggetti  29

Il Framework ping: Elementi di Base >> Gerarchia di Viste 

## Gerarchia di Viste



- Esecuzione dello schermo successivo
  - ⇒ le azioni chiamano il metodo
    - void eseguiSchermo(String nomeschermo)
    - sull'oggetto vista
  - ⇒ la vista ispeziona la gerarchia delle viste e appena trova un metodo con il nome specificato lo esegue
  - ⇒ è quindi indispensabile evitare che sottoviste diverse abbiano schermi con nomi uguali


G. Mecca - Programmazione Orientata agli Oggetti  30

Il Framework ping: Elementi di Base >> Gerarchia di Viste 

## Gerarchia di Viste



- Acquisizione dei valori forniti dall'utente
  - ⇒ analogamente, le azioni acquisiscono i valori dalla vista chiamando i metodi  
`public Object getValore(String nomeComponente);`
  - ⇒ per farlo però, i componenti relativi devono essere stati identificati con una chiamata al metodo `void setName(String nome)`
  - ⇒ naturalmente non possono esserci due componenti con lo stesso nome

G. Mecca - Programmazione Orientata agli Oggetti   31


Il Framework ping: Elementi di Base >> Gerarchia di Viste 

## Gerarchia di Viste

- Componenti attualmente ispezionati
  - ⇒ `JTextField`, `JPasswordField`, `JLabel`: restituisce un riferimento `String`, corrispondente al valore del testo
  - ⇒ `JComboBox`: restituisce il riferimento all'oggetto selezionato
  - ⇒ `JCheckBox`: restituisce un riferimento a `Boolean`, a seconda che sia selezionato o no
  - ⇒ `JRadioButton`: restituisce un riferimento a `Boolean`, a seconda che sia selezionato o no

G. Mecca - Programmazione Orientata agli Oggetti   32




Il Framework ping: Elementi di Base >> Gerarchia di Viste 

## Gerarchia di Viste >> indovinaIlNumero

- Selezione in una lista o da una tabella
  - ⇒ funziona in modo analogo
  - ⇒ `public Integer getIndiceSelezione(String nomeComponente);`
  - ⇒ cerca il componente con il nome specificato che abbia un modello di selezione (JList o JTable) e restituisce l'indice dell'elemento o della riga selezionata

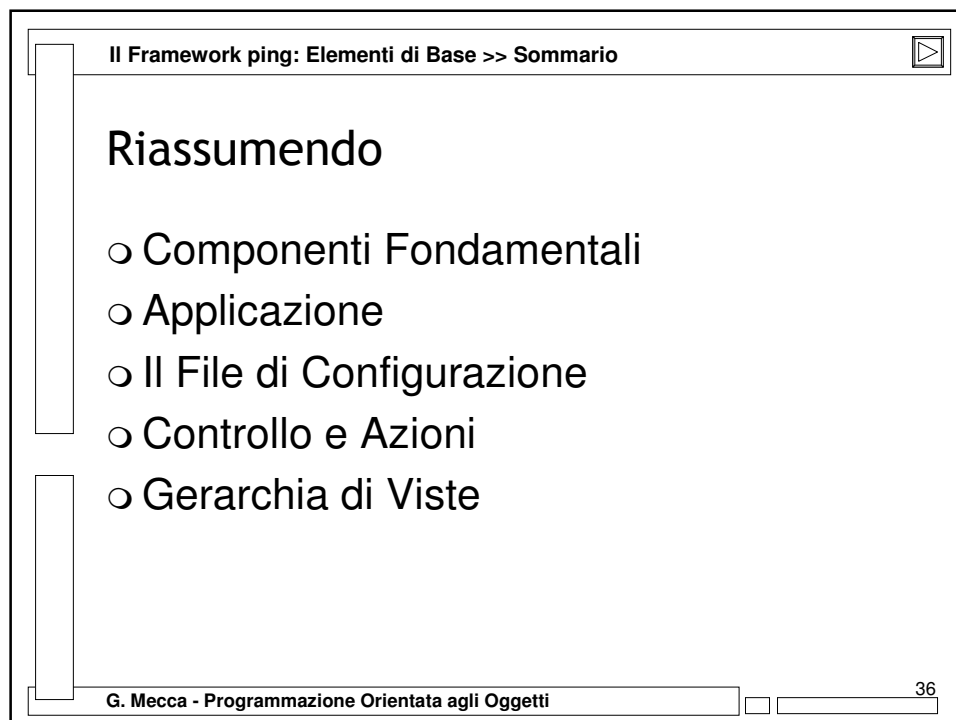
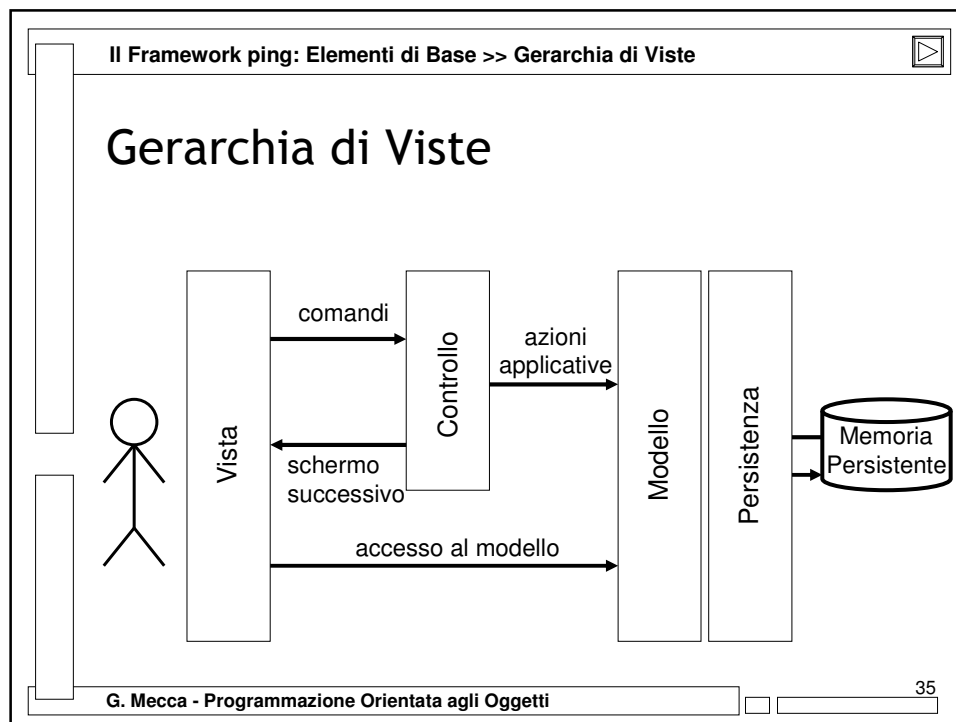
G. Mecca - Programmazione Orientata agli Oggetti 33


Il Framework ping: Elementi di Base >> Gerarchia di Viste 

## Gerarchia di Viste

- Schematizzando
  - ⇒ vista e controllo sono del tutto disaccoppiati
  - ⇒ la vista invoca le azioni nel controllo specificando comandi che sono stringhe
  - ⇒ le azioni invocano gli schermi successivi nella vista specificando nomi che sono stringhe

G. Mecca - Programmazione Orientata agli Oggetti 34



**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 37