Laboratorio di Basi di Dati

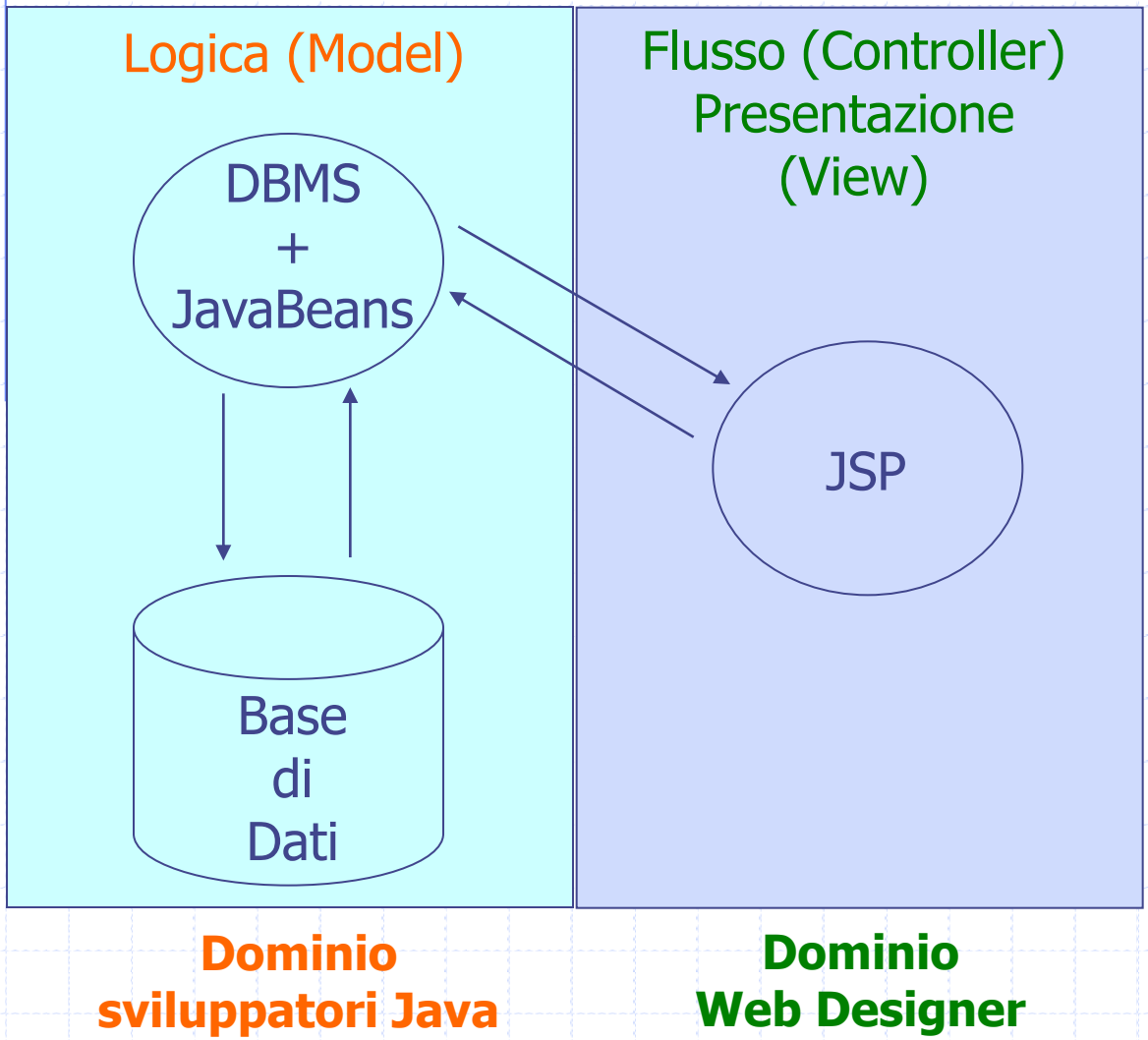
Docente: Alberto Belussi

Lezione 9

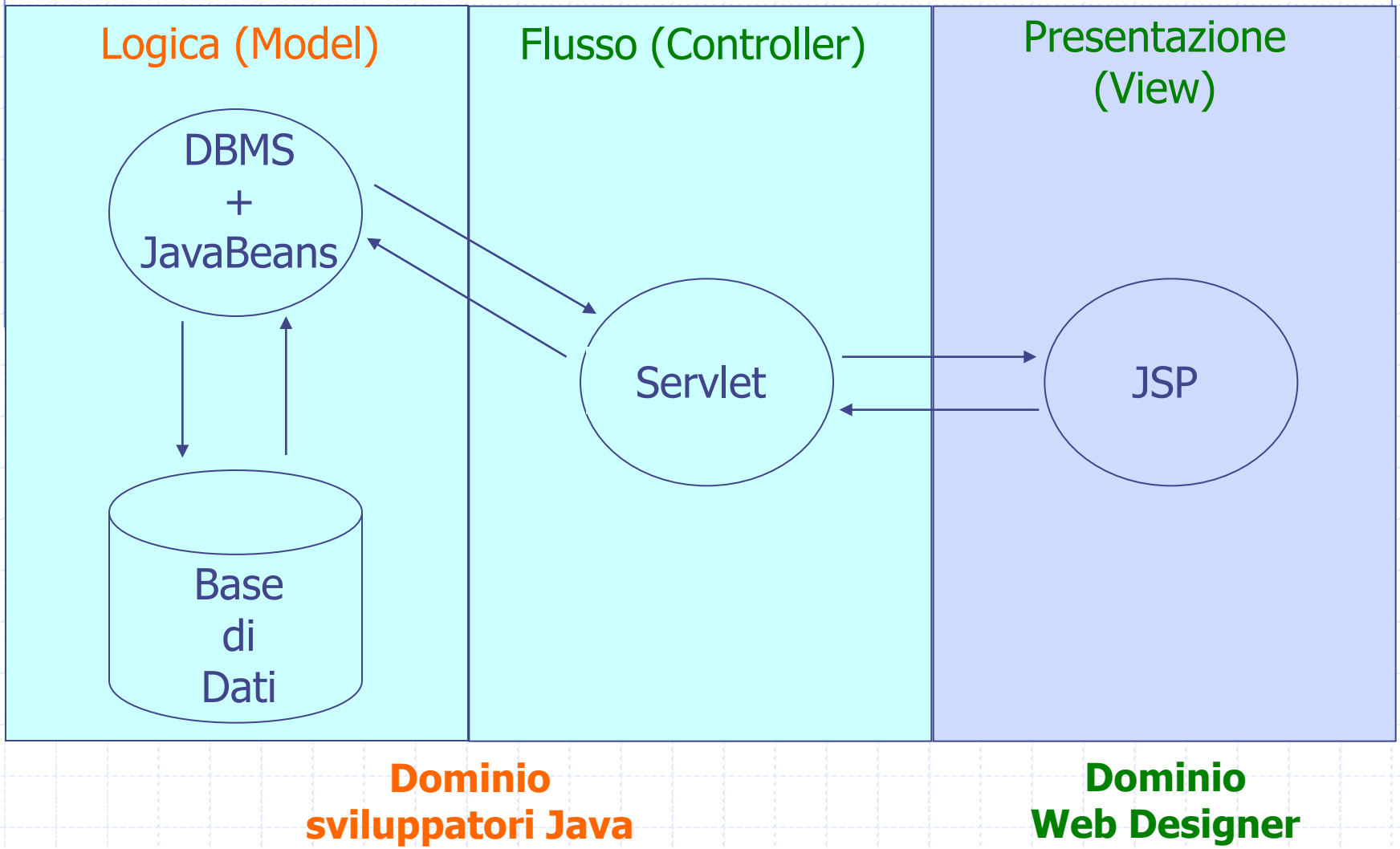
Architettura Model-View-Controller (MVC)

- Adottando l'architettura MVC e la tecnologia Servlet-JSP, un'applicazione web può essere realizzata secondo diversi approcci.
- I due approcci più significativi sono:
  - page-centric
  - servlet-centric

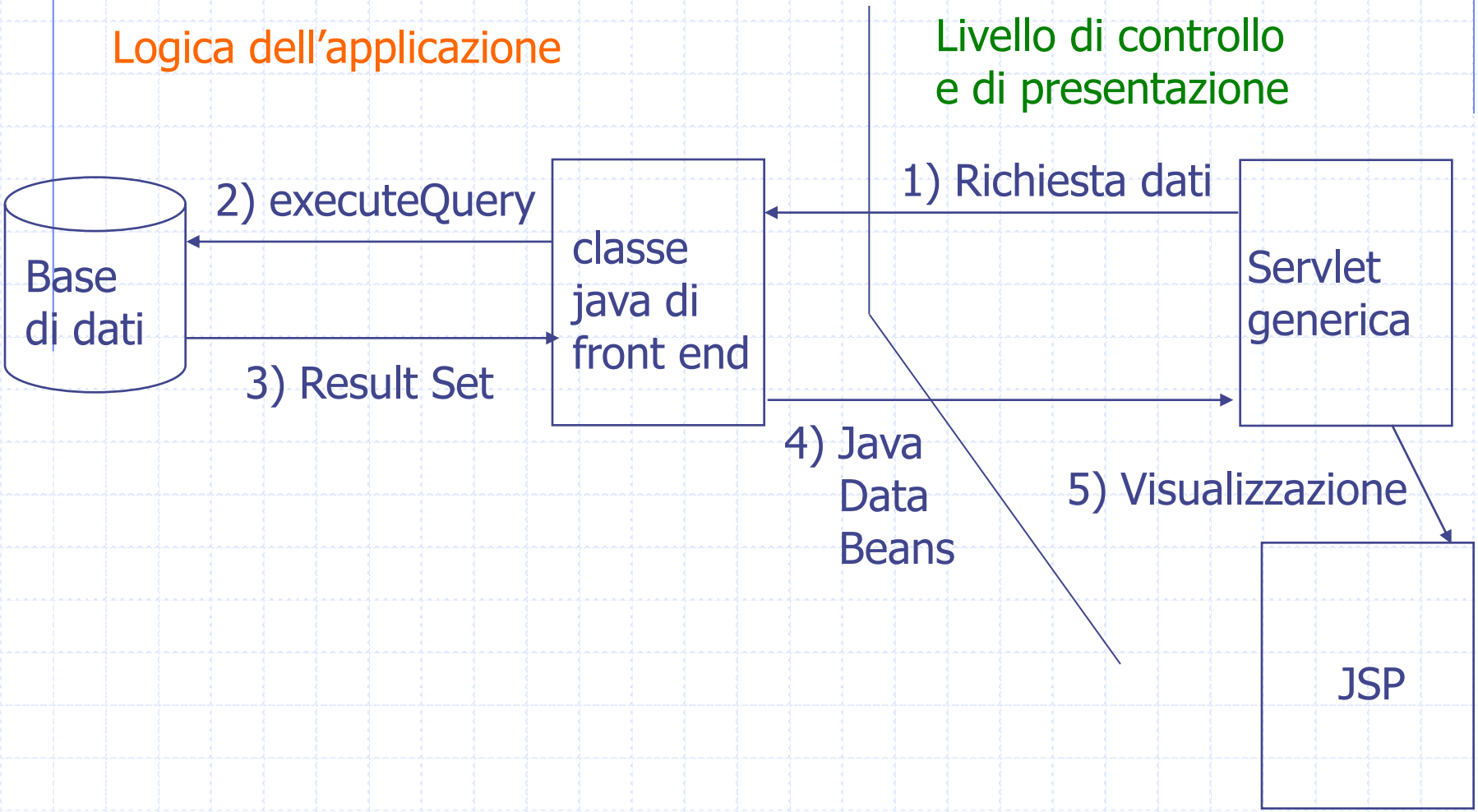
Approccio Page-centric



Approccio Servlet-centric (1)



Approccio Servlet-centric (2)



Approccio Servlet-centric (3)

- L'approccio servlet-centric prevede di utilizzare le pagine JSP solo per la presentazione e delegare il controllo ad una o ad una servlet. Le servlet quindi:
  - gestiscono le richieste (vengono cioè invocate tramite URI)
  - elaborano i dati necessari a soddisfare le richieste (utilizzando i JavaDataBean come componenti per rappresentare le informazioni di interesse)
  - trasferiscono il controllo alla JSP designata a presentare i risultati.

Approccio Servlet-centric (4)

- **Passaggio dati fra servlet-JSP:**

  i JavaDataBean istanziati dalla servlet devono essere passati alla JSP prima di trasferire ad essa il controllo. A tal fine esiste una coppia di metodi della classe HttpServletRequest che permettono di inserire/recuperare in/da request (oggetto implicito della JSP) un numero arbitrario di oggetti. Questi metodi sono:

  - setAttribute(String, Object)
  - getAttribute(String)

Approccio Servlet-centric (5)

- **Trasferimento del controllo dalla servlet alla JSP:** quando all'interno di una servlet, dopo aver preparato i dati e averli inseriti nell'oggetto *request*, si vuole richiamare una JSP per visualizzare i dati, si dice che si *trasferisce il controllo (forward)* alla JSP.

Approccio Servlet-centric (6)

- Per trasferire il controllo è necessario creare un oggetto di tipo RequestDispatcher associato alla JSP che si vuole 'invocare'.
- Ci sono due modi equivalenti per definire un oggetto RequestDispatcher associato ad una JSP all'interno di una servlet:
  - RequestDispatcher rd = request.getRequestDispatcher("PathRelativoJSP")
  - RequestDispatcher rd = getServletContext.getRequestDispatcher("PathAssolutoJSP")

Approccio Servlet-centric (7)

- Una volta ottenuto l'oggetto RequestDispatcher, è sufficiente invocare il metodo

  forward(HttpServletRequest, HttpServletResponse)

  per trasferire MOMENTANEAMENTE il controllo alla JSP.

- **Attenzione!** Non è un browser redirect e nemmeno una terminazione del metodo doGet o doPost della servlet... è una semplice chiamata di metodo. Tutto il codice presente DOPO forward(HttpServletRequest, HttpServletResponse) verrà eseguito dopo che la JSP ha finito la sua esecuzione!

Azioni per l'uso dei Bean nelle JSP

<jsp:**useBean** id="nome_bean"
class="nome_classe" scope="context"/>

dove:

- *id*: definisce un nome univoco da assegnare al bean
- *class*: specifica la classe del bean
- *scope*: indica il periodo di vita del bean, può essere:
  - page (default): il bean viene creato ad ogni richiesta della pagina
  - **request**: in questo caso il bean viene recuperato dall'oggetto request (utile nell'approccio MVC)
  - ….

Azioni per l'uso dei Bean nelle JSP

<jsp:**getProperty** name="nome_bean"
property="nome_prop"/>

dove:

- *name*: è il nome del bean da cui leggere la proprietà (assegnato nell'azione useBean attraverso l'attributo *id*)
- *property*: è la proprietà da leggere

Esempi da scaricare

1. Scaricare nella directory ~/tomcat/webapps/CorsoStudi la JSP: daFare.jsp dalla pagina web del corso.
2. La JSP daFare.jsp segnala che la funzionalità è da implementare.
3. Per far funzionare questo esempio è necessario:
    1. Scaricare il file main.java dalla pagina web del corso nella directory ~/tomcat/src/CorsoStudi.
    2. Compilare il package did e la servlet main nel seguente modo: javac –d ../../webapps/CorsoStudi/WEB-INF/classes main.java ./did/*.java
    3. Dichiarare la servlet main nel file web.xml in webapps/CorsoStudi/WEB-INF

Esempi da scaricare

4. Per vedere le pagine web prodotte dall'applicazione (come dichiarato nel file web.xml): http://localhost:8080/CorsoStudi/servlet/main
5. Completare la conversione dell'applicazione CorsoStudi all'architettura MVC servlet-centric (vedi esercizio proposto).
   - Modificare la servlet Main per la gestione del flusso di esecuzione, con i parametri indicati nel testo dell'esercizio.
   - Aggiungere le JSP mancanti per la visualizzazione delle informazioni.
     ATTENZIONE: le JSP non devono interagire direttamente con la classe DBMS ma ricevono i dati dalla servlet main attraverso l'oggetto request.

Consegna Esercitazione 9

Inviare via email al docente un file di nome

"ES9-<matricola>.zip"

contenente tutti i file dell'applicazione sviluppata presi dalle directory:

~/tomcat/src/CorsoStudi e

~/tomcat/webapps/CorsoStudi.

Il messaggio dovrà soddisfare il seguente formato:

- Oggetto: <Matricola> - Esercitazione 9
- Contenuto: <Matricola> - <Cognome> - <Nome>
- Allegato: file di nome ES9-<Matricola>.zip

Il messaggio email va spedito entro le 24.00 del giorno 1 giugno 2013.

Riferimenti

- Marty Hall. “CORE. Servlets and JavaServer Pages”. Sun Microsystems Press.
- Phil Hanna. “JSP. La guida Completa.” McGraw-Hill.