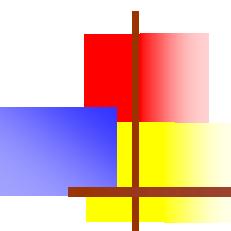


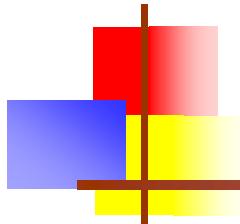
Java Server Pages





Prednosti JSP

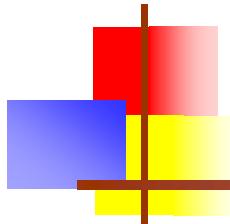
- Pomoću JSP se mogu uraditi iste stvari kao i pomoću servleta, ali korišćenje JSP olakšava
 - pisanje samog HTML koda
 - čitanje i održavanje HTML koda
- U odnosu na korišćenje servleta imamo:
 - Jednostavnije kreiranje HTML koda
 - Korišćenje standardnih alata (npr. Adobe Dreamweaver)
 - Podela posla između Java programera i dizajnera (HTML)
- JSP programeri moraju poznavati servlet programiranje



Osnovna JSP sintaksa

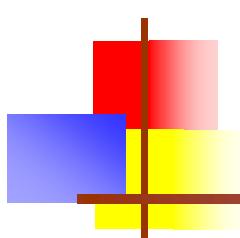
- Komentari
 - <%-- Komentar --%>
- Vrste dinamičkih elemenata
 - izrazi (expressions)
 - skriptleti (scriptlets)
 - deklaracije (declarations)
 - direktive (directives)

<%= java_izraz %>
<% java_kod %>
<%! int a; %>
<%@ direktiva attr="" %>



Prvi primer - JSP izrazi

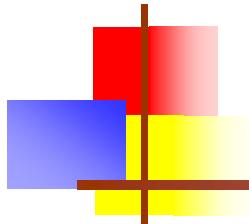
- Napisati JSP stranicu koja ispisuje datum i trenutno vreme koristeći izraze.
- `<%= new java.util.Date() %>`



Drugi primer - JSP skriptleti

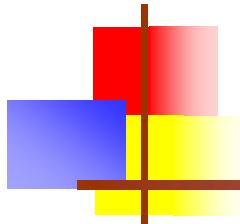
- Napisati JSP stranicu koja u zavisnosti od Java ugrađene metode Math.random() ispisuje "Dobar dan!" ako je slučajan broj manji od 0.5 ili "Dobro vece!" ako je broj veći od 0.5.
- Zadatak rešiti koristeći JSP skriptlete i IF-ELSE petlju.

- ```
<% if (Math.random() < 0.5) { %>
 Dobar dan!
<% } else { %>
 Dobro vece!
<% } %>
```



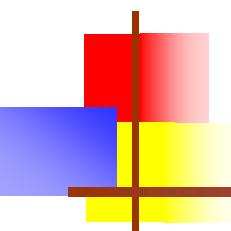
## Treći primer - JSP skriptleti

- Na svetskom prvenstvu u rukometu 2009. godine u gradu Poreču igraju nacionalni timovi Srbije, Danske, Norveške, Brazila i Egipta. JSP stranicom prikazati tabelu ove grupe (Redni broj, Ime tima) prema poretku navedenom u zadatku.



## Četvrti primer - JSP deklaracije

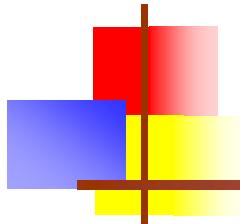
- Napisati JSP stranicu kojom ćete koristeći JSP deklaracije definisati naziv predmeta i ocenu, a zatim dodeliti vrednosti predmeta koji ste poslednji polagali i ocene na istom. Zatim prikazati te promenljive na ekranu.



## Peti primer

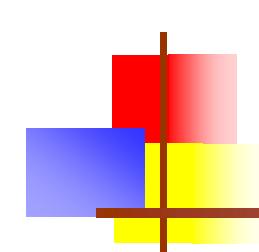
---

- Realizovati sabiranje 2 sabirka, celobrojnog tipa, koristeći JSP stranicu. Na jednoj JSP stranici definisati formu (kao što je rađeno u HTML jeziku), a zatim pozvati drugu JSP stranicu koja će generisati rezultat i ispisati ga.



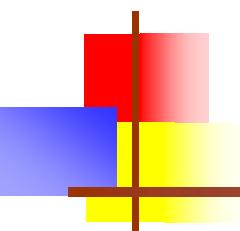
## Šesti primer

- Koristeći sesiju napisati JSP aplikaciju koja na jednoj stranici traži od korisnika da unese ime, a zatim u zavisnosti od unetog imena prikazuje poruku: “ Zdravo, :ime\_korisnika: ”



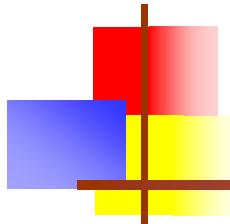
# Rad sa bean-ovima

- Šta su **beanovi** ?
- To su Java klase koje moraju ispunjavati nekoliko pravila:
  - moraju imati prazan konstruktor (bez argumenata)  
Ovaj zahtev se može ispuniti eksplicitnim definisanjem takvog konstruktora ili izostavljanjem svih konstruktora;
  - ne bi trebalo da ima public promenljive instanci (polja)
  - neposredne vrednosti bi trebale da se dobijaju pomoću metoda nazvanih **getImePolja** i **setImePolja**
  - ako klasa ima metod getNaslov koji kao rezultat vraća String, kaže se da klasa poseduje **property** nazvanu naslov
  - boolean properties koriste ImePolja umesto getImePolja



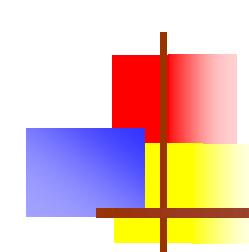
# Osnovna sintaksa

- Element **jsp:useBean** pravi novi bean
  - `<jsp:useBean id="beanName" class="package.Class" />`
- Element **jsp:getProperty** čita i prikazuje vrednosti definisanih bean property-ija
  - `<jsp:getProperty name="beanName" property="propertyName" />`
- Element **jsp:setProperty** menja vrednost bean property-ija
  - `<jsp:setProperty name="beanName" property="propertyName" value="propertyValue"/>`



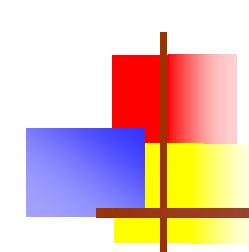
# jsp:useBean

- Sintaksa
  - <jsp:useBean id="name" class="package.Class" />
- Upotreba
  - Dozvoljava instanciranje Java klasa bez eksplicitnog Java programiranja
- Napomene
  - Interpretacija:  
`<jsp:useBean id="hello" class="binovi.HelloBean"/>`
  - Može se dobiti i pomoću skriptleta:  
`<% binovi.HelloBean hello=new binovi.HelloBean(); %>`



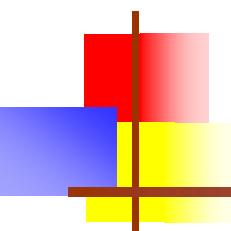
# jsp:getProperty

- Sintaksa
  - <jsp:getProperty name="name" property="property" />
- Upotreba
  - Dozvoljava pristup bean property-ijima bez eksplicitnog Java programiranja
- Napomene
  - <jsp:getProperty name="hello" property="name"/>
  - ekvivalentno je sa sledećim JSP izrazom:  
`<%= hello.getName() %>`



# jsp:setProperty

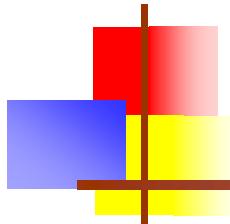
- Sintaksa
  - <jsp:setProperty name="name" property="property" value="value" />
- Upotreba
  - Dozvoljava postavljanje bean property-ijima bez eksplicitnog Java programiranja
- Napomene
  - <jsp:setProperty name="hello" property="name" value="Drazen"/>
  - ekvivalentno je sledećem skriptletu  
<% hello.setName("Drazen"); %>



# Napomene

---

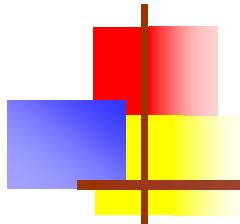
- Beanovi se nalaze u sledećem Java direktorijumu:
  - /build/web/WEB-INF/classes/binovi/HelloBean.class
- Beanovi (i pomoćne klase) moraju UVEK biti u okviru paketa!



# Postavljanje property-ija (1)

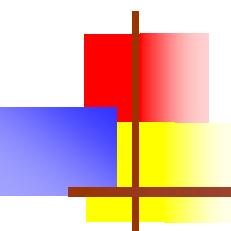
```
<jsp:useBean id="entry"
class="paket.SaleEntry" />

<jsp:setProperty
 name="entry"
 property="itemID"
 value='<%=request.getParameter("itemID")%'
/>
```



# Postavljanje property-ija (2)

```
<%
double discountCode = 1.0;
try{
 String discountString =
 request.getParameter("discountCode");
 discountCode=Double.parseDouble(discountString);
} catch (NumberFormatException nfe) { }
%>
<jsp:setProperty
 name = "entry"
 property = "discountCode"
 value = "<%= discountCode %>"
/>
```

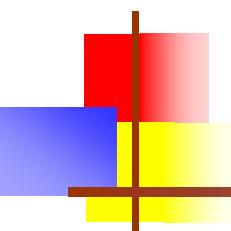


# Ulazni parametri

```
<jsp:setProperty
 name = "entry"
 property = "discountCode"
 param = "discountCode"
/>
```

- Simbol \* se može upotrebiti za vrednost property-ija atributa jsp:setProperty da bi se postiglo da se vrednost dobija preko parametra zahteva koji ima isto ime kao i ime property-ija

```
<jsp:setProperty
 name = "entry"
 property = "*" />
```

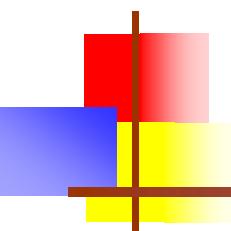


# Primer 7 - HelloBean.java

```
package binovi;
public class HelloBean {
 private String name = "World";

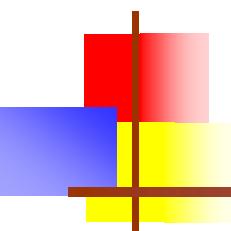
 public void setName(String name) {
 this.name = name;
 }

 public String getName() {
 return name;
 }
}
```



# Primer 7 - index.jsp

```
<%@page import="binovi.HelloBean" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:useBean id="hello" class="HelloBean">
 <jsp:setProperty name="hello" property="*" />
</jsp:useBean>
<html>
 <head><title>Hello</title></head>
 <body>
 <h1>Hello,
 <jsp:getProperty name="hello" property="name" />
 </h1>
 </body>
</html>
```

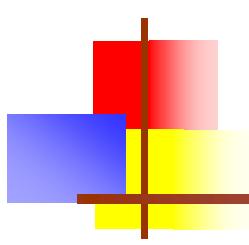


# Deljenje bean-ova

- Može se koristiti atribut scope da bi se definisale dodatne lokacije gde se može smestiti bean

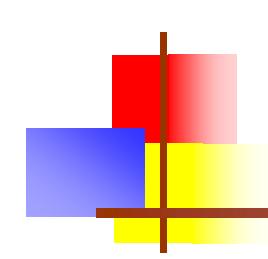
```
<jsp:useBean id="..."
class="..." scope="..." />
```

- Može se definisati da više servleta ili JSP stranica dele podatke
- Može se dozvoliti i uslovno kreiranje bean-a  
(kreirati novi objekat samo ako se ne može pronaći postojeći)



# Vrednosti atributa scope (1)

- page
  - <jsp:useBean ... scope="page" /> ili <jsp:useBean... />
  - Default vrednost. Bean objekat treba da bude smešten u okviru PageContext objekta za vreme trajanja zahteva.
- application
  - <jsp:useBean ... scope="application" />
  - Bean će biti smešten u ServletContext
  - ServletContext je deljen između svih servleta iste Web aplikacije



# Vrednosti atributa scope (2)

- session
  - <jsp:useBean ... scope="session" />
  - Bean će biti smešten u HttpSession objektu povezanim sa trenutnim zahtevom, gde mu se može pristupiti iz regularnog servlet koda sa metodama getAttribute i setAttribute kao bilo kom drugom objektu sesije
- request
  - <jsp:useBean ... scope="request" />
  - Bean objekat treba da bude smešten u HttpServletRequest objektu za vreme trajanja trenutnog zahteva, kada mu se može pristupiti pomoću getAttribute