

# Rinktiniai Java skyriai

Pratybos #5

2007-11-08

JSP tagai

# Prezentacijos logika

Prisiminimui: kam reikalingas JSP ?

# Prezentacijos logika

- JSP leidžia efektyviai atskirti informacijos atvaizdavimą (prezentaciją) nuo programos vidinės logikos
- Todėl HTML puslapių generavimui naudojame JSP, o ne servletus
- Tačiau ir čia ne viskas rožėmis klota. Yra toks dalykas kaip *prezentacijos logika*. Net ir atvaizduojant informaciją, tenka programuoti:
  - objektų laukų nuskaitymas
  - skirtingas pateikimas, priklausomai nuo sąlygų
  - iteravimas per duomenų struktūras (spausdinant sąrašą)
  - ir pan.

# Prezentacijos logika

- Prezentacijos logiką galima realizuoti su JSP reiškiniiais ir skriptletais, bet...
  - tuomet gauname HTML turinį, sumaišytą su Java kodu
  - nebelieka aiškaus programos logikos ir prezentacijos atskyrimo
  - tampa sunkiau atlikti pakeitimus puslapio dizaine
    - Didesniame projekte dizaineriai dirba atskirai nuo programuotojų. Dizaineriai nesupranta java kodo ir neturi žinoti, kaip jis veikia, ką galima liesti, o ko ne, ir pan.
- Išvada - prezentacijos logikai realizuoti reikalingos atskiros priemonės, kurios nereikalautų tiesioginio Java programavimo

# JSP tagai

- JSP tagai ateina į pagalbą
- JSP tagai leidžia abstrahuoti bet kokį Java kodą
- JSP tagai naudoja XML sintaksę, todėl natūraliai įsipaišo į HTML turinį
- JSP tagai dar vadinami *JSP action'ais*

# JSP tagai

- JSP tagas atrodo kaip paprastas XML elementas
  - Jis turi namespace'o prefiksą, elemento vardą ir atributus
  - Namespace'as identifikuoja tagų biblioteką (taglib), kuriai tas tagas priklauso
  - Tago vardas identifikuoja Java klasę - taip vadinamą "tago handlerį". Ši klasė realizuoja visą tago logiką.
  - Atributai atitinka tago klasės laukus, tiksliau, tų laukų setter'ius (pagal JavaBeans konvenciją)

Šis tagas išspausdins žodį "Yahoo!" penkis kartus

```
<my:yahoo repeat="5" />
```

namespace'o  
prefiksas

tago vardas

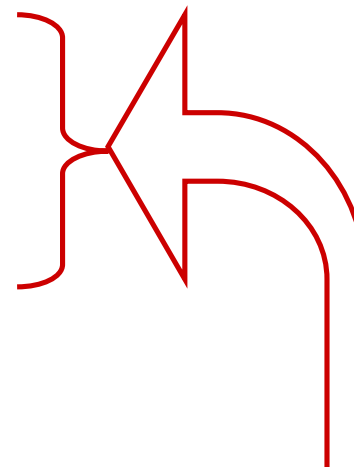
atributas

# JSP tagai

- JSP tagas gali turėti kūną, kuriame gali būti HTML turinys bei kiti JSP tagai
  - Tago kūną esantys JSP tagai vadinami jo "vaikais"

```
<my:list>  
  <li style="color: red">  
    <my:item/>  
  </li>  
</my:list>
```

Šis vaikinis tagas  
neturi kūno



Tago <my:list> kūnas:  
HTML ir/arba XML turinys

# Taglib'ai

- JSP @taglib direktyvoje namespace'o prefiksas yra susiejamas su namespace'o URI (Uniform Resource Identifier)
  - URI identifikuoja taglib'o (tagų bibliotekos) aprašo failą
  - URI gali būti arba simbolinis taglib'o vardas arba konkretus failo adresas
- Viename JSP puslapyje galima naudoti daug skirtingų namespace'ų, taigi ir daug skirtingų taglib'ų

```
<%@ page import="test.YahooHandler" %>
<%@ taglib prefix="my" uri="/WEB-INF/MyTagLibrary.tld" %>
<html>
<body>
  <p>
    <my:yahoo repeat="5" />
  <p>
</body>
</html>
```



# Taglib'o aprašas

- Taglib'o aprašas yra XML failas su plėtiniu "tld". Jis paprastai dedamas į katalogą /WEB-INF
- Taglib'o apraše nurodomi visi bibliotekos tag'ai, jų handler'iai ir atributai (žr. XML komentarus apačioje)

```
<?xml version="1.0"?>
<taglib version="2.0" xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-jsptaglibrary_2_0/xsd">

  <tlib-version>1.0</tlib-version>
  <short-name>my-taglib</short-name>

  <tag> <!-- kiekvienam tagui po tokį elementą -->
    <name>yahoo</name> <!-- tago vardas -->
    <tag-class>test.YahooHandler</tag-class> <!-- tago handler'is -->
    <body-content>empty</body-content> <!-- ar turi kūną -->
    <attribute> <!-- kiekvienam tago atributui po tokį elementą -->
      <name>repeat</name> <!-- atributo vardas -->
      <required>true</required> <!-- ar tai būtinas atributas -->
    </attribute>
  </tag>
</taglib>
```

# Tago handler'is

Turint tokį taglib'o aprašą, tago iškvietimas:

```
<my:yahoo repeat="5" />
```

... yra ekvivalentus kodui:

```
<%  
    YahooHandler yahoo = new YahooHandler();  
    yahoo.setRepeat("5");  
    yahoo.doTag();  
%>
```

# Tago handler'is

- Tago handler'is yra klasė, kuri implementuoja JspTag interfeisą. Paprastiems tagams patogiau paveldėti SimpleTagSupport klasę, perrašyti metodą doTag() ir kiekvienam tago atributui sukurti po setXxx() metodą.

```
package test;

import java.io.*;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;

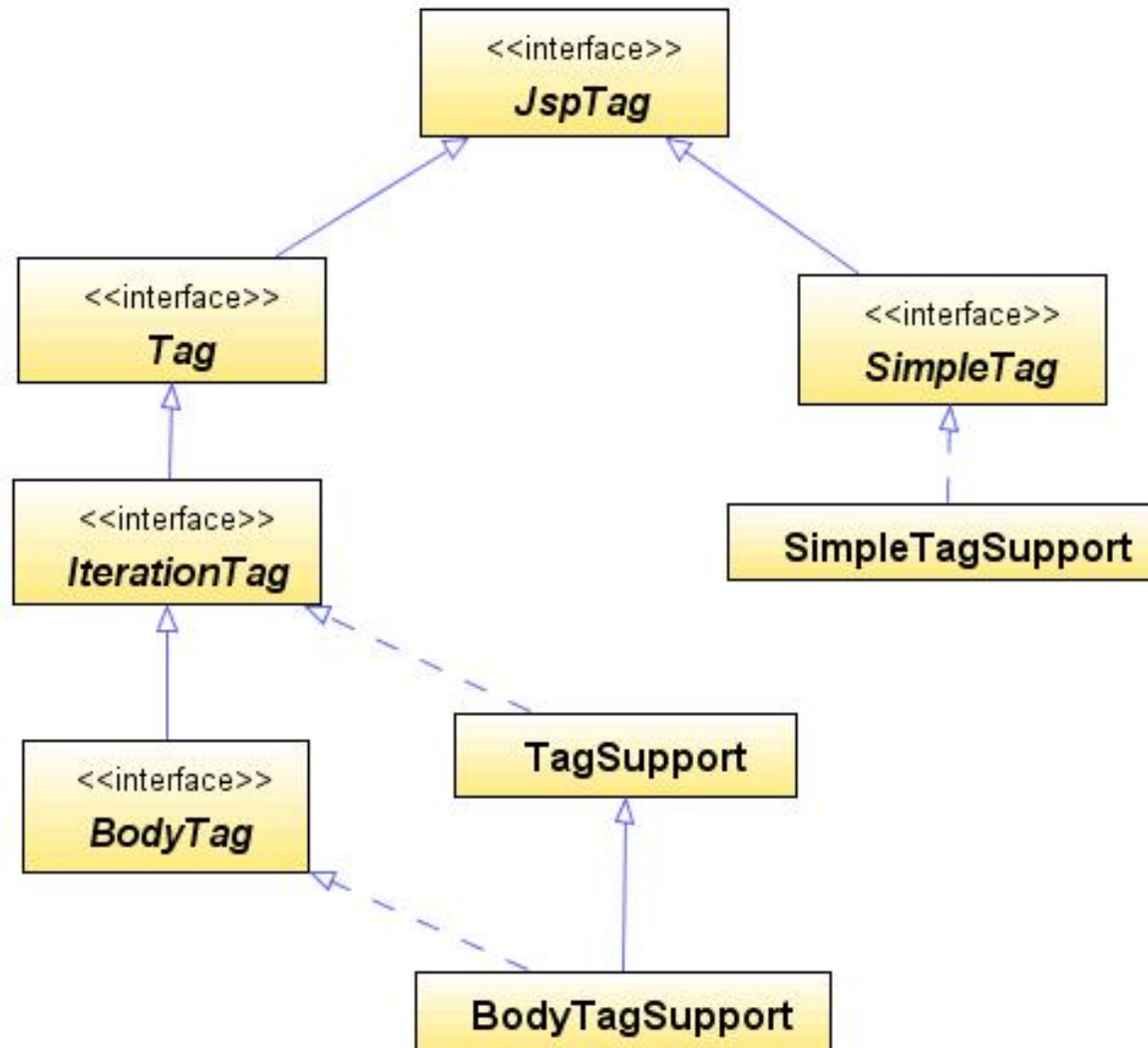
public class YahooHandler extends SimpleTagSupport {

    private int repeat = 1;

    public void setRepeat(String value) {
        this.repeat = Integer.parseInt(value);
    }

    public void doTag() throws JspException, IOException {
        JspWriter out = getJspContext().getOut();
        for (int i = 0; i < repeat; i++) {
            out.println("Yahoo! ");
        }
    }
}
```

# Tagų handler'ių interfeisai ir klasės



# Tagai su prefiksu "jsp:"

JSP taip pat turi kelis tagus su prefiksu "jsp:".

Tokius tagus galima naudoti JSP puslapiuose neaprašant nei namespace'o, nei taglib'o.

Toliau pateikiami du tokių tagų pavyzdžiai.

# Tagai su prefiksu "jsp:"

`<jsp:include>` kreipiasi į kitą JSP puslapį (arba bet kokį kitą failą) ir įterpia jo output'ą

```
<jsp:include page="Header.jsp" />
```

```
<jsp:include page="Header.jsp">  
  <jsp:param name="color" value="red" />  
</jsp:include>
```

# Tagai su prefiksu "jsp:"

`<jsp:forward>` perduoda valdymą kitam JSP puslapiui (negrįžtamai)

```
<jsp:forward page="Process.jsp" />
```

```
<jsp:forward page="Process.jsp">  
  <jsp:param name="id" value="<%= id %>" />  
</jsp:forward>
```

Tag'ų atributų reikšmės gali būti nurodomos dinamiškai:  
arba su JSP reiškiniais,  
arba su Expression Language

# Tagai su prefiksu "jsp:"

<jsp:useBean>, <jsp:getProperty> ir <jsp:setProperty> gali būti naudojami klasės laukams skaityti / rašyti. Tokia klasė turi laikytis JavaBean konvencijos.

```
<jsp:useBean id="stud" class="test.Student" scope="request"/>
<jsp:getProperty name="stud" property="firstName"/>
<jsp:setProperty name="stud" property="firstName" value="Petras"/>
```

Tai ekvivalentu kodui:

```
<%
    test.Student stud = request.getAttribute("stud");
    out.print(stud.getFirstName());
    stud.setFirstName("Petras");
%>
```

- ➔ Šie tagai leidžia programuoti deklaratyviai, tačiau labai jau griozdiskai. Elegantiškesnis ir glaustesnis sprendimas - JSTL su Expression Language.