

JSF КОМПОНЕНТИ НА ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙС

Съдържание

Част 1:

- JSF въведение
- JSF правила за навигация
- JSF управляеми компоненти (managed beans)
- **JSF компоненти на потребителски интерфейс**

ВИЗУАЛИЗАЦИЯ НА КОМПОНЕНТИТЕ

- Отделянето на състоянието и поведението от визуализацията води до следните преимущества:
 - ▣ Веднъж написано поведение на компонент може да бъде визуализирано:
 - По различни начини (например, като препратка (Hyperlink) или бутон)
 - За различен клиент (**HTML** клиент, **WML** клиент)
 - ▣ Дизайнерите могат да изберат подходящата визуализация в зависимост от дизайна на сайта за достигане до желаното от тях поведение

Декларативни езици за изгледи (View Declaration Languages)

8

□ JSP

□ Facelets

- ▣ xhtml – валиден и добре форматиран (valid and well formed)
- ▣ Обработка на грешки (вкл. номер на ред на грешката)
- ▣ EL директно в страницата `#{myBean.propertyName}`
- ▣ Улеснено създаване на темплейти (`ui:composition`, `ui:define`, `ui:insert`, `ui:include`, `ui:repeat`)

EL в атрибути на тагове

9

- Всички изрази могат да се използват като стойности на атрибутите.
- Какъв вид израз и как се оценява – незабавно или отложено (immediately или deferred), се определя от типа дефиниция на тага във **View Description Language (VDL)**, дефинираща тага.

EL - VDL

10

- Потребителски тагове – за всеки се описва какъв тип израз е.

Attribute Type	Example Expression	Type Attribute Definition
Dynamic	"literal"	<rtexprvalue>true</rtexprvalue>
Dynamic	\${literal}	<rtexprvalue>true</rtexprvalue>
Deferred value	"literal"	<deferred-value> <type>java.lang.String</type> </deferred-value>
Deferred value	#{customer.age}	<deferred-value> <type>int</type> </deferred-value>
Deferred method	"literal"	<deferred-method> <method-signature> java.lang.String submit() </method-signature> </deferred-method>
Deferred method	#{customer.calcTotal}	<deferred-method> <method-signature> double calcTotal(int, double) </method-signature> </deferred-method>

Запазени думи - EL

11

and	or	not	eq
ne	lt	gt	le
ge	true	false	null
instanceof	empty	div	mod

JavaServer Faces HTML Tags Reference

<http://www.exadel.com/tutorial/jsf/jsftags-guide.html>

UIInput

inputText

```
<h:inputText id="address"
  value="#{jsfexample.address}" />
```

```
<input type="text" name="jsftags:_id1"
  value="123 JSF Ave" />
```

inputSecret

```
<h:inputSecret redisplay="false"
  value="#{jsfexample.password}" />
```

```
<input id="jsftags:password"
  type="password"
  name="jsftags:password"
  value="secret" />
```

inputHidden

```
<h:inputHidden id="hidden"
  value="userPreference" />
```

No Renderer

```
<input id="jsftags:hidden"
  type="hidden"
  name="jsftags:hidden"
  value="userPreference" />
```

inputTextArea

```
<h:inputTextarea id="textArea"
  rows="4" cols="7"
  value="Text goes here.."/>
```

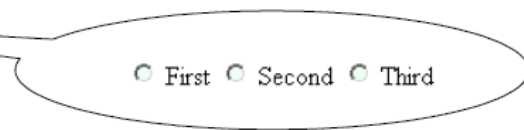
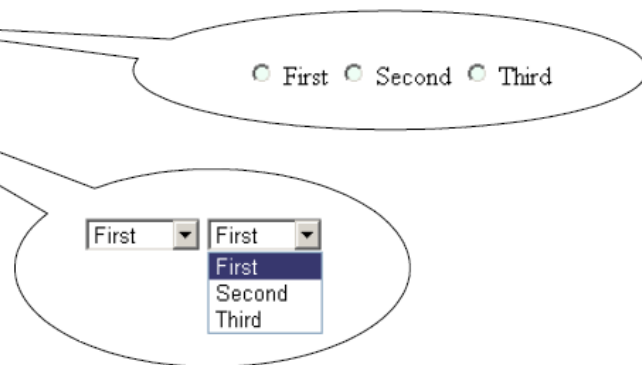
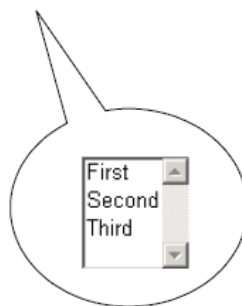
```
<textarea id="jsftags:textArea"
  name="jsftags:textArea"
  cols="5" rows="3">
  Text goes here..
</textarea>
```

UIOutput

RENDER KIT

13

- Определя как се визуализират компонентите за даден клиент
- **JSF** имплементацията включва вградена поддръжка за визуализация към **HTML** клиент
- Набор от класове, чрез които компонентите се визуализират
- Един компонент може да бъде визуализиран по няколко начина:
 - ▣ `UISelectOne` – избор на една стойност от множество
 - `h:selectOneRadio`
 - `h:selectOneMenu`
 - `h:selectOneListbox`



Демо/Задача за изпълнение

14

- Задача 1*: разгледайте компонентите в пространство на имената h:*
- Задача 2*: Направете jsf страница, която използва голяма част от тях;
- Задача 3*: Направете регистрационна форма, използвайки компонентите от задача 2;
- **Задача 4:** Направете страница за вписване (login page) , с полета име, парола и роля с която да влезнете (например admin, user, guest) ; направете сесиен компонент (managed bean @SessionScope), в който се запазва ролята на потребителя; направете по няколко страници (1-2) за всяка от ролите;
- **Забележка:** * - задачите са по желание

```

package mymodel;

public class TemperaturConvertor {
    private double celsius;
    private double fahrenheit;
    private boolean initial= true;

    public double getCelsius() {
        return celsius;
    }
    public void setCelsius(double celsius) {
        this.celsius = celsius;
    }
    public double getFahrenheit() {
        return fahrenheit;
    }

    public boolean getInitial(){
        return initial;
    }

    public String reset (){
        initial = true;
        fahrenheit =0;
        celsius = 0;
        return "reset";
    }
    public String celsiusToFahrenheit(){
        initial = false;
        fahrenheit = (celsius *9 / 5) +32;
        return "calculated";
    }
}

```

Пример - Model

(*TemperaturConvertor.java*)

1.4.2015 г.

Пример

ManagedBean

Managed Bean Elements

The following managed beans are defined

- session
- temparaturConverter
- request
- application
- none

Add

Remove

Managed Bean

This section describes general configuration of this managed bean

Managed Bean name*: temparaturConverter

Managed Bean class*: mymodel.TemparaturConverter

Browse...

Managed Bean scope*: session

Initialization

You can initialize the managed bean's properties or itself if it is a subclass of java.util.Map or java.util.List

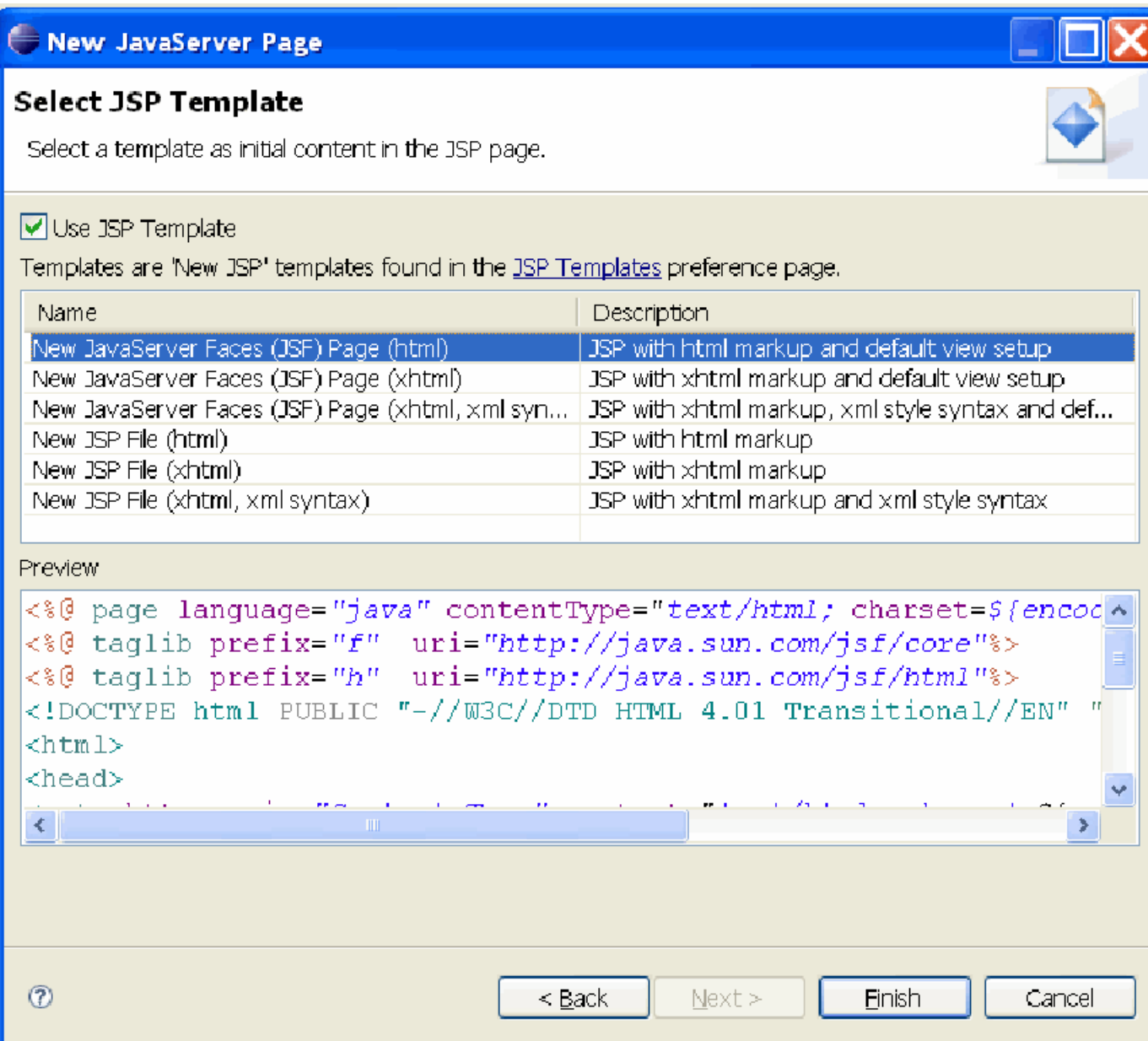
Managed Bean class type: ☒ General class ☐ Map ☐ List

Name	Class	Value

Add...

Edit...

Remove



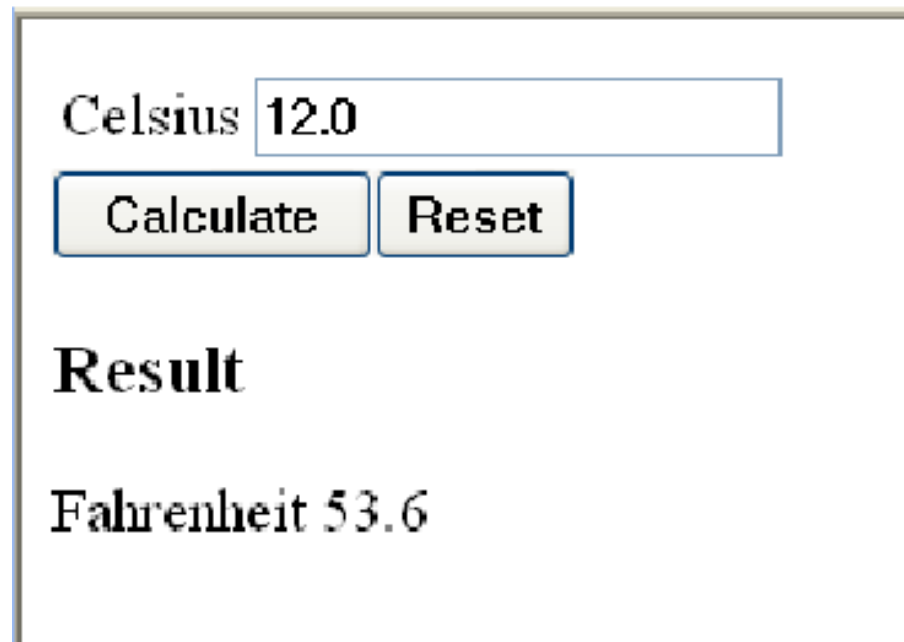
Пример –
създаване на
JSP

Пример –JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<%@ taglib prefix="h" uri="http://java.sun.com/jsf/html"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Celsius to Fahrenheit Converter</title>
</head>
<body>
<f:view>
    <h:form>
        <h:panelGrid columns="2">
            <h:outputLabel value="Celsius"></h:outputLabel>
            <h:inputText value="#{temperaturConverter.celsius}"></h:inputText>
        </h:panelGrid>
        <h:commandButton action="#{temperaturConverter.celsiusToFahrenheit}" value="Calculate"></h:commandButton>
        <h:commandButton action="#{temperaturConverter.reset}" value="Reset"></h:commandButton>
        <h:messages layout="table"></h:messages>
    </h:form>

    <h:panelGroup rendered="#{temperaturConverter.initial!=true}">
        <h3> Result </h3>
        <h:outputLabel value="Fahrenheit "></h:outputLabel>
        <h:outputLabel value="#{temperaturConverter.fahrenheit}"></h:outputLabel>
    </h:panelGroup>
</f:view>
</body>
</html>
```

Пример - результат



A screenshot of a web application for temperature conversion. It features a text input field labeled "Celsius" containing the value "12.0". Below the input are two buttons: "Calculate" and "Reset". Underneath the buttons, the word "Result" is displayed in a bold serif font. At the bottom, the text "Fahrenheit 53.6" shows the converted value.

Celsius 12.0

Calculate Reset

Result

Fahrenheit 53.6

JSF ТАГОВЕ

- Списък на всички стандартни **JSF** тагове:

http://java.sun.com/javaee/javaxserverfaces/1.2_MR1/docs/tlddocs

- Примери за визуализация:

<http://www.exadel.com/tutorial/jsf/jsftags-guide.html>

В лекцията бяха разгледани

- Концепция за изгледи
- Визуални компоненти – основни понятия
- Видове визуални компоненти
- Пример
- JSF тагове за визуализация

23

Допълнителни слайдове

Допълнение – panelGrid компонент

24

- Съдържание
 - ▣ Цел на компонента
 - ▣ JSF таг – пример за таблица
 - ▣ атрибути

Графичен компонент за изграждане на таблица

25

- Съдържание
 - ▣ Цел на компонента
 - ▣ JSF таг – пример за таблица
 - ▣ атрибути

графични елементи

□ **<h:panelGrid>**

- таг се използва за създаването на съставни компоненти и за тяхната подредба (layout)
- Този таг се използва за създаването на html таблица със специфициран брой на колоните;
- Вложените елементи се подреждат като клетки в таблицата, според зададения брой колони, например:

<h:panelGrid title="Цена" columns="1" border="1">

- Например ако са специфицирани брой на колоните 1 и има зададени 3 елемента – те ще се разпределят на 3 реда. Ако имаме например columns="2" и 3 елемента – на вторият ред 2-рата колона ще е празна.
- По подразбиране броя на колоните в panelGrid е „1“

JSF tag: <h:panelGrid> (таблица)

```
<h:panelGrid title="Легенда на отстъпките"
columns="2" border="1" rules="all"
width="100%">
    <f:facet name="header">
        <h:outputText value="Отстъпки:"/>
    </f:facet>
    <h:outputText id="messageLabel"
value="#{count.message}" />
    <h:outputText id="out1"
value="#{count.count}" />
</h:panelGrid>
```

Основни атрибути на panelGrid (1/4)

- **id** : идентификатор на компонента – трябва да е уникален за текущия контейнер
- **binding** : взема стойността на свързващия израз на визуалния компонент със свойство на съответния компонент на сървъра (property of the backing bean)
- **rendered** : булев атрибут със стойност по подразбиране - true. Указва дали този компонент да се изчислява (render) или не.
- **columns** : брой на колоните в таблицата
- **border**: int- указва ширината на таблицата
- **bgcolor**: background цвят на таблицата
- **cellpadding**: указва разстоянието между съдържанието на таблицата и границите на клетката (border of the cell).
- **cellspacing**: разстоянието между отделните клетки в таблицата

Основни атрибути на panelGrid (2/4)

- **frame:** специфицира частта от рамката, която обхваща таблицата дали да е видима
- **rules:** специфицира кои линии от клетките в таблицата да са видими
- It can take some values :
 - 1) **none** без страни на клетката (по подразбиране)
 - 2) **groups** между групи от клетки
 - 3) **rows** само между клетката
 - 4) **cols** само между колоните
 - 5) **all** между редовете и колоните
- **summary :** Резюме на съдържанието на таблицата.
- **width :** ширина на таблицата – може да е в пиксели или в проценти

Основни атрибути на panelGrid (3/4)

- **columnClasses** : използва се списък от имена, разделени със запетайка от CSS стилови класове, да бъдат приложени на колоните на таблицата
- **headerClass** : отделен списък с CSS класове за заглавният ред на таблицата.
footerClass : отделен списък с CSS класове за завършващият (последен) ред на таблицата.
- **rowClasses** : използва се списък от имена, разделени със запетайка от CSS стилови класове, да бъдат приложени на клетките на таблицата
- **dir** : посока на визуализирания текст – стойностите могат да са LTR(left to right) и RTL (right to left).
- **lang** : базов език за визуализация на компонентата
- **style** : използва се за установяване на CSS стил на компонентата.
- **title** : стандартен HTML атрибут – за показване на подсказващ текст (tooltip)
- **styleClass** : използва се за указване на CSS клас на компонента
- **onclick** : скрипт, който се извиква след като се кликне върху компонента
ondblclick : скрипт (javascript), който се извиква след като се двойно се кликне върху компонента

Основни атрибути на panelGrid (4/4)

Други събития:

onmousedown : It is used for Java Script code to be invoked when the pointing device is pressed over this element.

□ **onmouseup** : It is used for Java Script code to be invoked when the pointing device is released over this element.

□ **onmouseover** : It is used for Java Script code to be invoked when the pointing device is moved into this element.

□ **onmousemove** : It is used for Java Script code to be invoked when the pointing device is moved while it is in this element.

□ **onmouseout** : It is used for Java Script code to be invoked when the pointing device is moves out of this element.

□ **onkeypress** : It is used for Java Script code to be invoked when a key is pressed over this element.

□ **onkeydown** : It is used for Java Script code to be invoked when a key is pressed down over this element.

□ **onkeyup** : It is used for Java Script code to be invoked when a key is released over this element

Резултантен HTML код (за panelGrid)

```
<table border="1" rules="all" title="Легенда на отстъпките"  
width="100%">
```

```
<thead>
```

```
<tr><th colspan="2" scope="colgroup">Отстъпки:</th></tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td><span id="messageLabel">Няма избрани стоки</span></td>
```

```
<td><span id="out1">0</span></td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

Въпроси

