

# Programowanie portali biznesowych

## WYKŁAD 4

# JavaServer Faces

**Specyfikacja JSR (Java Specification Request) 344 dla JSF:**

<https://jcp.org/en/jsr/detail?id=344>

## **JavaServer Faces**

- **JavaServer Faces (JSF)** – technologia należąca do platformy Java EE służąca do implementacji dynamicznych stron WWW.
- Technologia JSF stanowi rozszerzenie serwletów.

## Możliwości JSF

- Tworzenie stron WWW przy użyciu zbioru komponentów interfejsu użytkownika zgodnie ze wzorcem Model-Widok-Kontroler (*Model-View-Controller – MVC*).
- Wiązanie komponentów z modelem po stronie serwera.
- Obsługa nawigacji pomiędzy stronami.
- Zarządzanie stanami komponentów interfejsu użytkownika.
- Walidacja danych.

## Możliwości JSF

- Prosta obsługa błędów.
- Możliwość wykorzystania niestandardowych komponentów interfejsu użytkownika.
- Prosty model kojarzenia zdarzeń z ich obsługą.
- Wsparcie dla internacjonalizacji.

## Facelety

- **Facelety** – język deklarowania widoków (mechanizm obsługi widoków).
- Facelety zastępują technologię JSP.

## Facelety

- Strony faceletów tworzone są przy użyciu znaczników języka XHTML i kaskadowych arkuszy stylów (CSS).
- Strony XHTML tłumaczone są na strony HTML.
- Strony muszą być zgodne ze specyfikacją XHTML-1.0-Transitional DTD.
- Komponenty strony mogą być dodawane za pomocą znaczników faceletów (przedrostek **h**) oraz za pomocą znaczników języka HTML.

## Facety

- Standardowe biblioteki znaczników obsługiwanych przez facelety:

Przedrostek	Opis
h	komponenty JSF
f	akcje niestandardowe
c	znaczniki JSTL ( <i>core tags</i> )
fn	funkcje JSTL
ui	szablony

## JSTL - JSP Standard Tag Library

## Przykładowe komponenty

```
<h:form>
```

```
<h:panelGrid columns="2">
```

```
<h:inputText id="marka"  
value="#{samochod.marka}">
```

```
<h:outputText value="Marka: ">
```

```
<h:dataTable  
value="#{samochody.lista}" var="s">
```

## Znaczniki

- Dokumentacja:

<https://docs.oracle.com/javaee/7/javaserver-faces-2-2/vd1docs-facelets/toc.htm>

## Język EL

- **EL** (Expression Language) – język wyrażeń obsługiwany w faceletach służący do wiązania komponentów wewnętrznych z interfejsem użytkownika.
- W wyrażeniach języka EL można odwoływać się do komponentów CDI (*Contexts and Dependency Injection*) posiadających adnotację **@Named** lub komponentów EJB.

## Język EL

- Przykład 1 wyrażenia EL:

`#{samochod.marka}`

- odwołanie do pola **marka** komponentu

**Samochod:**

@Named

@SessionScoped

```
public class Samochod implements Serializable {  
    private String marka;  
    // ...  
}
```

Wartość wyrażenia jest wyznaczana przez kontroler w innych fazach czasu życia.

## Język EL

- Przykład 2 wyrażenia EL:

**$\${samocho\dot{d}.marka}$**

- odwołanie do pola **marka** komponentu **Samochod** (wartość wyrażenia jest wyznaczana natychmiast).

## Język EL

- Przykład 3 wyrażenia EL:

**`#{resource['obrazek.jpg']}`**

- odwołanie do zasobu

## Dołączanie CSS

- Przykład:

```
<h:outputStylesheet name="style.css"  
library="css"/>
```

# Dołączanie skryptów JavaScript

- Przykład:

```
<h:outputScript name="skrypt.js"  
library="skrypty" />
```

## Walidacja danych

- Przykład:

```
<h:inputText value="#{samochod.rocznik}"
required="true" id="rocznik">
<f:validateLongRange minimum="1900"
maximum="2019"/>
<f:validateLength minimum="4"
maximum="4"/>
</h:inputText>
```

Wyświetlenie błędu:

```
<h:message for="rocznik"/>
```

## Reguły nawigacji

- Przykład 1 (reguły w pliku `faces-config.xml`):

```
<navigation-rule>
```

```
  <from-view-id>/login.xhtml</from-view-id>
```

```
  <navigation-case>
```

```
    <from-action>#{LoginForm.login}</from-action>
```

```
    <from-outcome>success</from-outcome>
```

```
    <to-view-id>/strona1.xhtml</to-view-id>
```

```
  </navigation-case>
```

```
  <navigation-case>
```

```
    <from-action>#{LoginForm.logon}</from-action>
```

```
    <from-outcome>failure</from-outcome>
```

```
    <to-view-id>/strona2.xhtml</to-view-id>
```

```
  </navigation-case>
```

```
</navigation-rule>
```

## Reguły nawigacji

- Przykład 2 (nawigacja bezpośrednia):

```
<h:commandButton value="submit" action="odpowiedz">
```