

Programowanie portali biznesowych

WYKŁAD 11

JSTL

- **JSTL (JavaServer Pages Tag Library)** – zbiór znaczników wykorzystywanych do budowy aplikacji webowych, pozwalających m.in. tworzyć struktury warunkowe oraz iteracyjne.

JSTL

- Dyrektywa dla znaczników podstawowych:

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

- Wybrane znaczniki podstawowe:
 - **<c:out>** - wyświetlanie wartości zmiennej
 - **<c:set>** - ustawianie wartości zmiennej
 - **<c:remove>** - usuwanie wartości zmiennej
 - **<c:if>** - instrukcja warunkowa
 - **<c:choose>**, **<c:when>**, **<c:otherwise>**
 - instrukcja switch
 - **<c:forEach>** - instrukcja pętli

JSTL

- Dyrektywa dla znaczników formatowania:

```
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
```

- Wybrane znaczniki formatowania:
 - **<fmt:formatDate>** - formatowanie daty
 - **<fmt:parseDate>** - parsowanie daty
 - **<fmt:formatNumber>** - formatowanie liczby
 - **<fmt:parseNumber>** - parsowanie liczby

JSTL

- Dyrektywa dla funkcji:

```
<%@ taglib prefix="fn"  
uri="http://java.sun.com/jsp/jstl/functions" %>
```

- Wybrane funkcje dla łańcuchów znaków:
 - `fn:contains()`
 - `fn:contains()`
 - `fn:join()`
 - `fn:replace()`
 - `fn:split()`
 - `fn:toLowerCase()`
 - `fn:toUpperCase()`

Przekazywanie danych z modelu do widoku

- **Model**
- Przykład:

```
public String pobierz(Model model)
{
    Map<String, String> mapa = new HashMap<>();
    mapa.put("spring", "mvc");
    model.addAttribute("komunikat", "Spring");
    model.mergeAttributes(mapa);
    return "widok";
}
```

Przekazywanie danych z modelu do widoku

- **ModelMap**

- Przykład:

```
public String pobierz(ModelMap mapa)
{
    mapa.addAttribute("komunikat", "Spring");
    return "widok";
}
```

Przekazywanie danych z modelu do widoku

- **ModelAndView**

- Przykład:

```
public ModelAndView pobierz()  
{  
    ModelAndView mw = new ModelAndView("widok");  
    mw.addObject("komunikat", "Spring");  
    return mw;  
}
```

Przekazywanie danych z modelu do widoku

- *Przekazywanie niejawne*
- Przykład metody kontrolera:

```
public List<Ocena> pobierz()  
{  
    EntityManagerFactory emf=  
        Persistence.createEntityManagerFactory("SpringMVCTest2PU");  
    EntityManager em=emf.createEntityManager();  
    List<Ocena> lista=em.createQuery("SELECT o FROM Ocena o",  
        Ocena.class).getResultList();  
    return lista;  
}
```

Przekazywanie danych z modelu do widoku

- *Przekazywanie niejawne:*
 - jeśli metoda obsługująca żądanie zwraca obiekt lub kolekcję obiektów, zwracana wartość umieszczana jest w modelu
 - klucz dla wartości w modelu ustalany jest na podstawie zwracanego typu
 - nazwa widoku ustalana jest na podstawie ścieżki żądania

Przekazywanie danych z modelu do widoku

- *Przekazywanie niejawne*

- Przykład widoku:

```
<html> <!-- ... -->
<body>
<c:forEach items="{ocenaList}" var="ocena">
<c:out value="{fn:toUpperCase(ocena.przedmiot)}"/>:
<c:out value="{ocena.ocena}"/>
<br/>
</c:forEach>
</body>
</html>
```