

# **Budowa aplikacji wielowarstwowych. zastosowanie komponentów złożonych JSF.**

**Laboratorium 10**  
**Technologie internetowe**  
**Zofia Kruczkiewicz**

Wykaz pytań dotyczących materiału wykorzystanego w lab10, które należy opracować (m.in.wykłady: 8).

1. Należy wyjaśnić rolę znaczników **<composite:interface**, **<composite:attribute** i **<composite:implementation** w budowie strony dodaj\_produk2.xhtml. Należy podać, jakie elementy strony są zdefiniowane za pomocą tych znaczników i w jaki sposób są wykorzystane przy budowie konkretnej strony.
2. Należy wyjaśnić rolę znaczników **<composite:interface**, **<composite:attribute** i **<composite:implementation** w budowie strony rezultat2.xhtml. Należy podać, jakie elementy strony są zdefiniowane za pomocą tych znaczników i w jaki sposób są wykorzystane przy budowie konkretnej strony.

**Czynności początkowe przed wykonaniem p. 1.  
Wykonany program jest przeznaczony do rozwijania wg p.1-6.**

- Należy wykonać kopię programu, wykonanego podczas lab9, jako **Sklep\_10** (wg wskazówek z instrukcji do lab. 2, slajd 4: [http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/ti/LAB\\_TINT\\_2.pdf](http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/ti/LAB_TINT_2.pdf)). Nazwa projektu w tej instrukcji nie pokrywa się z proponowaną nazwą projektu – w instrukcji wykorzystano program wykonany wg instrukcji do lab4.
- Ustawić kodowanie UTF-8; po zaznaczeniu nazwy projektu w oknie **Projects** prawym klawiszem myszy wybrać pozycję **Properties/Sources/Encoding/UTF-8**

Na podstawie przykładu ze str. 15-29 należy w projekcie z lab.9 zmodyfikować strony **dodaj\_produkt2.xhtml** oraz **rezultat2.xhtml**. Podobnie należy zmodyfikować definicję strony **lista\_produktow.xhtml** - z wykorzystaniem komponentów kompozytowych.

# Złożone komponenty JSF

wg

<https://docs.oracle.com/javaee/7/JEETT.pdf>

<http://www.coreservlets.com>

**Wg wykładu**  
**Technologie internetowe 8**

# **Opis znaczników obsługiwanych przez Facelets (tutorial EE 7)**

# Przegląd znaczników JSF (UI) - composite

wykład3:

[http://zofia.kruckiewicz.staff.iar.pwr.wroc.pl/wyklady/ti /TINT\\_3.pdf](http://zofia.kruckiewicz.staff.iar.pwr.wroc.pl/wyklady/ti /TINT_3.pdf)

Znacznik	Funkcja znaczników szablonu
<b>composite:interface</b>	Definicja jednego komponentu jako połączenie cech wielu komponentów
<b>composite:implementation</b>	Definiuje implementację kompozytowego komponentu. W przypadku definicji <b>composite:interface</b> implementacja musi być zgodna z tą definicją
<b>composite:attribute</b>	Deklaracja atrybutu instancji komponentu, do którego ten znacznik jest przypisany
<b>composite:insertChildren</b>	Dowolny komponent lub tekst szablonu ze znacznikiem kompozytowym w używanej stronie jest powtarzany w punkcie umieszczenia tego znacznika w ramach znacznika <b>composite:implementation</b>

# Przegląd znaczników JSF (UI) - composite (cd)

<b>composite:valueHolder</b>	Deklaracja znacznika wewnątrz znacznika <b>composite:interface</b> . Definicja implementacji <b>ValueHolder</b> właściwego dla obiektów używanych na stronie
<b>composite:editableValueHolder</b>	Deklaracja znacznika wewnątrz znacznika <b>composite:interface</b> . Definicja implementacji <b>EditableValueHolder</b> właściwego dla obiektów używanych na stronie
<b>composite:actionSource</b>	Deklaracja znacznika wewnątrz znacznika <b>composite:interface</b> . Definicja implementacji <b>actionSource</b> właściwego dla obiektów używanych na stronie



# Biblioteki znaczników obsługiwanych przez Facelets

wykład3:

[http://zofia.kruckiewicz.staff.iar.pwr.wroc.pl/wyklady/ti /TINT\\_3.pdf](http://zofia.kruckiewicz.staff.iar.pwr.wroc.pl/wyklady/ti /TINT_3.pdf)

Biblioteki znaczników	URI	Prefiks	Przykład	Zawartość
Composite Component Tag Library	<a href="http://xmlns.jcp.org/jsf/composite">http://xmlns.jcp.org/jsf/composite</a>	<b>cc:</b>	cc:interface	Znaczniki wspierające komponenty kompozytowe

# Atrybuty komponentu złożonego

Nazwa atrybutu	Opis
<b>name</b>	Specyfikuje: <ul style="list-style-type: none"><li>• nazwę atrybutu komponentu złożonego, używanego przez stronę, wykorzystującej dany component.</li><li>• alternatywnie, ten atrybut może reprezentować nazwę standardowego słuchacza zdarzeń: action lub ActionListener,</li><li>• nazwę ManagedBean</li></ul>
<b>default</b>	Specyfikuje domyślną nazwę atrybutu komponentu złożonego
<b>required</b>	Określa, kiedy nazwa atrybutu jest obowiązkowa
<b>method-signature</b>	Specyfikuje podklasę klasy java.lang.Object, jako atrybut komponentu złożonego i deklaruje definicję metody. Atrybuty <b>type</b> i <b>method-signature</b> wzajemnie się wykluczają. Jeśli specyfikuje się te dwa atrybuty jednocześnie, atrybut method-signature jest ignorowany. Domyślny typ atrybutu jest java.lang.Object. <b>Uwaga:</b> Wyrażenie specyfikujące metodę jest podobne do specyfikacji atrybutu, ale zamiast definiować dynamikę właściwości metody, wspiera wywołanie metody podanego obiektu, podając specyfikację zbioru przekazywanych parametrów i zwracany wynik wywołania tej metody (jeśli ta metoda zwraca wynik).

# Przykłady definicji atrybutów

## default:

```
<composite:attribute name="username" default="admin"/>
```

## method-signature:

```
<composite:attribute name="myaction"  
    method-signature="java.lang.String action()"/>
```

## type:

```
<composite:attribute name="dateofjoining"  
    type="java.util.Date"/>
```

# Odwołanie do komponentu typu Managed Bean

**Aby komponent złożony mógł realizować przetwarzanie po stronie serwera, należy go powiązać z komponentem typu Managed Bean w następujący sposób:**

- przekazanie referencji komponentu Managed Bean do złożonego komponentu
- bezpośrednie użycie właściwości komponentu typu Managed Bean w komponencie złożonym

# Walidacja wartości komponentu złożonego

Walidacja w technologii JSF jest używana w złożonych komponentach wejściowych zdefiniowanych za pomocą znaczników typu

- **composite:valueHolder**
- **composite:editableValueHolder.**

## Znaczniki walidacji

Nazwa	Opis
<b>f:validateBean</b>	Deleguje walidację lokalnych wartości Bean Validation API
<b>f:validateRegex</b>	Używa wzorca atrybutu do walidacji komponentu. Wejściowy wzorzec odpowiada wartości łańcuchowej komponentu. Jeśli wartości są zgodne, walidacja przebiega poprawnie.
<b>f:validateRequired</b>	Sprawdza obecność wartości atrybutu – równoważny efekt to ustawienie elementu atrybutu komponentu na wartość true.

# Przykład

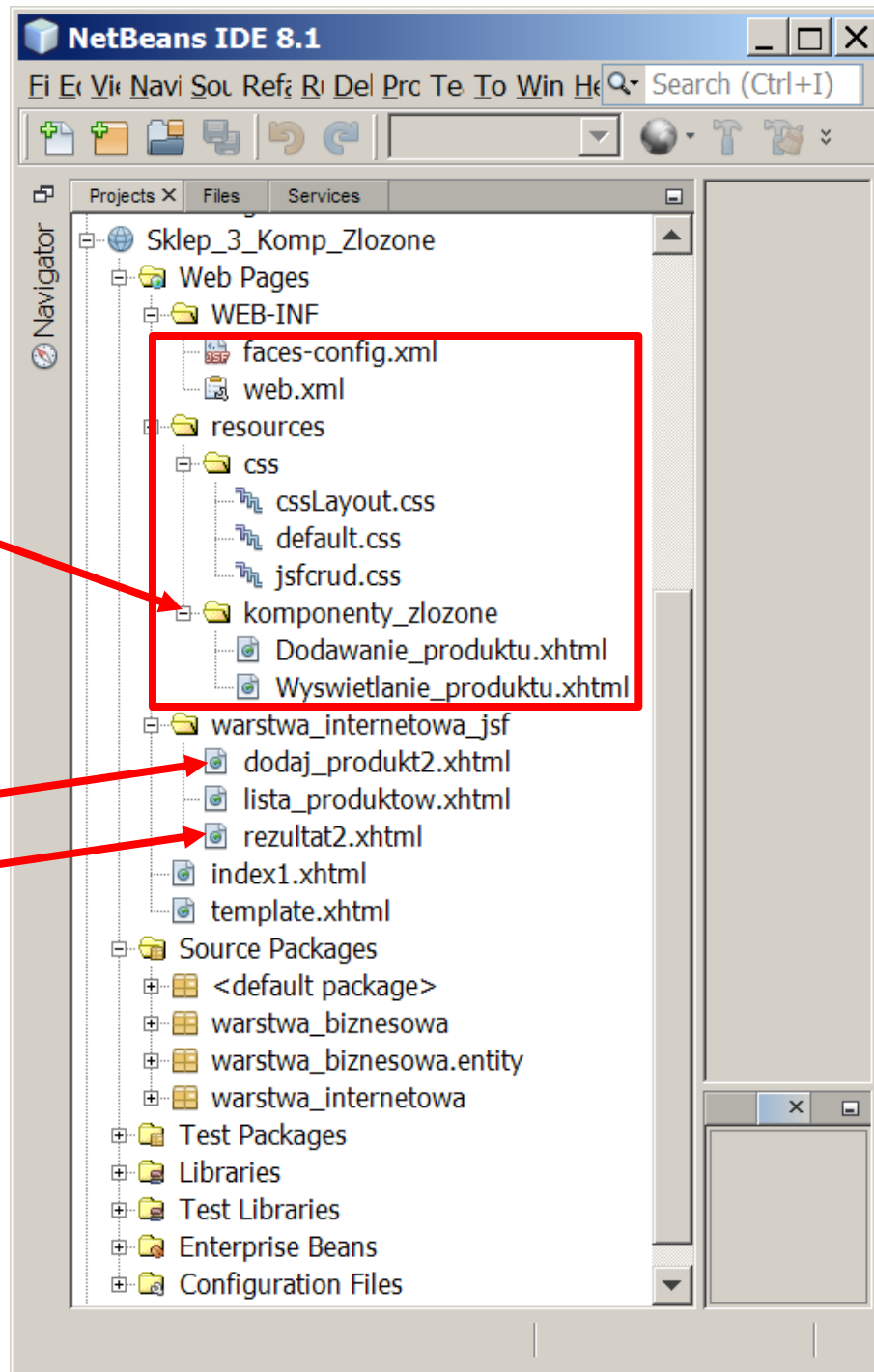
## Procedura tworzenia i wykorzystanie komponentu złożonego:

1. **Wykonanie folderu resources** w katalogu głównym **Web Pages** projektu i **zagnieżdżonego folderu**, gdzie będą umieszczane pliki z definicją komponentu złożonego.
2. **Wykonanie pliku.xhtml** typu JSF Page w zagnieżdżonym w folderze resources i zadeklarowanie przestrzeni nazw komponentu złożonego
3. Wykorzystanie znaczników **composite:interface**, **composite:attribute** i **composite:implementation**, do zdefiniowania zawartości komponentu złożonego.
3. Znacznik **composite:interface** są używany do deklaracji wartości konfiguracyjnych. Znacznik **composite:implementation** jest używany do deklaracji znaczników XHTML. Wykorzystuje on atrybuty znacznika **composite:interface** za pomocą wyrażenia: **{cc.attrs.attributeName}**.
4. **W celu użycia komponentu kompozytowego** należy wykonać kolejny plik typu **.xhtml** i w przestrzeni nazw strony umieścić wyrażenie reprezentujące miejsce definicji komponentu złożonego np **http://xmlns.jcp.org/jsf/composite/nazwa\_folderu**, gdzie **nazwa\_folderu** jest nazwą folderu wykonanego w folderze **resources** (p. 1),
5. Zastosowanie zdefiniowanego komponentu złożonego, podobnie jak komponenty JSF

# Przykład – Sklep3\_Komp\_Zlozone

Definicje  
komponentów  
złożonych

Wykorzystanie  
komponentów  
złożonych do  
definicji stron



## Plik dodaj\_produk2.xhtml przed zmianą

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html">
<body>
<ui:composition template="../template.xhtml">
<ui:define name="title">#{bundle['dodaj_produk2.tytul']}
```



## Dodawanie\_produkту.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://xmlns.jcp.org/jsf/html"
  xmlns:composite="http://xmlns.jcp.org/jsf/composite">
  <composite:interface >
    <composite:attribute name="nameLable" />
    <composite:attribute name="nameValue" />
    <composite:attribute name="titleName" />
    <composite:attribute name="nameMessage" />
    <composite:attribute name="cenaLable" />
    <composite:attribute name="cenaValue" />
    <composite:attribute name="titleCena" />
    <composite:attribute name="cenaMessage" />
    <composite:attribute name="promocjaLable" />
    <composite:attribute name="promocjaValue" />
    <composite:attribute name="titlePromocja" />
    <composite:attribute name="promocjaMessage" />
    <composite:attribute name="dodajButtonText" />
    <composite:attribute name="dodajButtonAction" method-signature="java.lang.String action()" />
  </composite:interface>
```

```
<composite:implementation >
```

```
<h:form>
```

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

```
  #{cc.attrs.nameLabel} :
```

```
  <h:inputText id="name" value="#{cc.attrs.nameValue}"
```

```
    title="#{cc.attrs.titleName}"
```

```
    required="true" requiredMessage="#{cc.attrs.nameMessage}"/>
```

```
  #{cc.attrs.cenaLabel} :
```

```
  <h:inputText id="cena" value="#{cc.attrs.cenaValue}"
```

```
    title="#{cc.attrs.titleCena}"
```

```
    required="true" requiredMessage="#{cc.attrs.cenaMessage}"/>
```

```
  #{cc.attrs.promocjaLabel} :
```

```
  <h:inputText id="promocja" value="#{cc.attrs.promocjaValue}"
```

```
    title="#{cc.attrs.titlePromocja}"
```

```
    required="true" requiredMessage="#{cc.attrs.promocjaMessage}"/>
```

```
</h:panelGrid>
```

```
<h:commandButton action="#{cc.attrs.dodajButtonAction}"
```

```
  value="#{cc.attrs.dodajButtonText}"/>
```

```
</h:form>
```

```
</composite:implementation>
```

```
</html>
```

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
  xmlns:h="http://xmlns.jcp.org/jsf/html"
  xmlns:komponenty_zlozone=
    "http://xmlns.jcp.org/jsf/composite/komponenty_zlozone">
<body>
<ui:composition template=" ../template.xhtml">
  <ui:define name="title"> #{bundle['dodaj_produk2.tytul']} </ui:define>
  <ui:define name="content">
    <komponenty_zlozone: Dodawanie_produkту
      nameLabel="#{bundle['dodaj_produk2.nazwa']}"
      nameValue="#{managed_produk2.nazwa}"
      titleName="#{bundle['dodaj_produk2.nazwa1']}"
      nameMessage="#{bundle['dodaj_produk2.blad_nazwa']}"
```

## `dodaj_produkt2.xhtml` po zastosowaniu komponentów złożonych (cd)

```
cenaLabel="#{bundle['dodaj_produkt2.cena']}"
cenaValue="#{managed_produkt.cena}"
titleCena="#{bundle['dodaj_produkt2.cena1']}"
cenaMessage="#{bundle['dodaj_produkt2.blad_cena']}"
promocjaLabel="#{bundle['dodaj_produkt2.promocja']}"
promocjaValue="#{managed_produkt.promocja}"
titlePromocja="#{bundle['dodaj_produkt2.promocja1']}"
promocjaMessage="#{bundle['dodaj_produkt2.blad_promocja']}"
dodajButtonText="OK"
dodajButtonAction="#{managed_produkt.dodaj_produkt}"
/>
</ui:define>
</ui:composition>
</body>
</html>
```

## Zrzut z ekranu kodu strony `dodaj_produk2.xhtml` po zastosowaniu komponentów złożonych

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
  xmlns:h="http://xmlns.jcp.org/jsf/html"
  xmlns:komponenty_zlozone=
    "http://xmlns.jcp.org/jsf/composite/komponenty_zlozone">
  <body>
    <ui:composition template="../../../template.xhtml">
      <ui:define name="title">    #{bundle['dodaj_produk2.tytul']} </ui:define>
      <ui:define name="content">
        <komponenty_zlozone:Dodawanie_produktu
          nameLabel=#{bundle['dodaj_produk2.nazwa']}
          nameValue=#{managed_produk2.nazwa}
          titleLabel=#{bundle['dodaj_produk2.nazwa1']}
          nameMessage=#{bundle['dodaj_produk2.blad_nazwa']}
          cenaLabel=#{bundle['dodaj_produk2.cena']}
          cenaValue=#{managed_produk2.cena}
          titleCena=#{bundle['dodaj_produk2.cena1']}
          cenaMessage=#{bundle['dodaj_produk2.blad_cena']}
          promocjaLabel=#{bundle['dodaj_produk2.promocja']}
          promocjaValue=#{managed_produk2.promocja}
          titlePromocja=#{bundle['dodaj_produk2.promocja1']}
          promocjaMessage=#{bundle['dodaj_produk2.blad_promocja']}
          dodajButtonText="OK"
          dodajButtonAction=#{managed_produk2.dodaj_produk2}
        />
      </ui:define>
    </ui:composition>
  </body>
</html>
```

## Plik rezultat2.xhtml przed zmianą

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:h="http://java.sun.com/jsf/html">
<body>
  <ui:composition template="../../../template.xhtml">
    <ui:define name="title"> #{bundle['rezultat2.tytul']} </ui:define>
    <ui:define name="content">
      <h:form>
        <h:outputText escape="false" value="#{bundle['rezultat2.nie_dodano']}"
          rendered="#{managed_produkt.stan==0}"/>
        <h:panelGrid columns="2" rendered="#{managed_produkt.stan!=0}">
          <h:outputLabel value="#{bundle['rezultat2.nazwa_produktu']}" for="nazwa" />
          <h:outputText id="nazwa" value="#{managed_produkt.nazwa}"/>
          <h:outputLabel value="#{bundle['rezultat2.cena']}" for="cena" />
          <h:outputText id="cena" value="#{managed_produkt.cena}"/>
          <h:outputLabel value="#{bundle['rezultat2.promocja']}" for="promocja" />
          <h:outputText id="promocja" value="#{managed_produkt.promocja}"/>
          <h:outputLabel value="#{bundle['rezultat2.cena_brutto']}" for="brutto" />
          <h:outputText id="brutto" value="#{managed_produkt.cena_brutto}"/>
          <h:commandButton id="powrot" value="#{bundle['rezultat2.powrot']}"
            action="/faces/index1"/>
        </h:panelGrid>
      </h:form>
    </ui:define>
  </ui:composition>
</body>
</html>
```

## Wyswietlanie\_produkту.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:composite="http://xmlns.jcp.org/jsf/composite">
  <composite:interface>
    <composite:attribute name="nameLabel" />
    <composite:attribute name="nameValue" />
    <composite:attribute name="cenaLabel" />
    <composite:attribute name="cenaValue" />
    <composite:attribute name="promocjaLabel" />
    <composite:attribute name="promocjaValue" />
    <composite:attribute name="cena_bruttoLabel" />
    <composite:attribute name="cena_bruttoValue" />
    <composite:attribute name="pokazButtonText" />
    <composite:attribute name="pokazButtonAction" method-signature="java.lang.String action()"/>
    <composite:attribute name="brak"/>
    <composite:attribute name="brakdanych"/>
    <composite:attribute name="standanych1"/>
    <composite:attribute name="standanych2"/>
  </composite:interface>
```

**<composite:implementation >**

<h:form>

<h:outputText escape="#{cc.attrs.brak}" value="#{cc.attrs.brakdanych}"  
rendered="#{cc.attrs.standanych1}"/>

<h:panelGrid columns="2" rendered="#{cc.attrs.standanych2}">

**#{cc.attrs.nameLabel} :**

<h:outputText id="name" value="#{cc.attrs.nameValue}" />

**#{cc.attrs.cenaLabel} :**

<h:outputText id="cena" value="#{cc.attrs.cenaValue}" />

**#{cc.attrs.promocjaLabel} :**

<h:outputText id="promocja" value="#{cc.attrs.promocjaValue}" />

**#{cc.attrs.cena\_bruttoLabel} :**

<h:outputText id="cena\_brutto" value="#{cc.attrs.cena\_bruttoValue}" />

</h:panelGrid>

<h:commandButton action="#{cc.attrs.pokazButtonAction}"  
value="#{cc.attrs.pokazButtonText}"/>

</h:form>

**</composite:implementation>**

</html>



```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
  xmlns:komponenty_zlozone=
    "http://xmlns.jcp.org/jsf/composite/komponenty_zlozone">
<body>
  <ui:composition template="./../template.xhtml">
    <ui:define name="title"> #{bundle['rezultat2.tytul']} </ui:define>
    <ui:define name="content">
      <komponenty_zlozone:Wyswietlanie_produktu
        brak="false"
        brakdanych="#{bundle['rezultat2.nie_dodano']}"
        standanych1="#{managed_produkt.stan==0}"
        standanych2="#{managed_produkt.stan!=0}"
```

## rezultat2.xhtml po zastosowaniu komponentów złożonych (cd)

```
nameLabel="#{bundle['rezultat2.nazwa_produkту']}"
nameValue="#{managed_produkт.nazwa}"
cenaLabel="#{bundle['rezultat2.cena']}"
cenaValue="#{managed_produkт.cena}"
promocjaLabel="#{bundle['rezultat2.promocja']}"
promocjaValue="#{managed_produkт.promocja}"
cena_bruttoLabel="#{bundle['rezultat2.cena_brutto']}"
cena_bruttoValue="#{managed_produkт.cena_brutto}"
pokazButtonText="#{bundle['rezultat2.powrot']}"
pokazButtonAction="#{managed_produkт.powrot}"
/>
```

```
</ui:define>
```

```
</ui:composition>
```

```
</body>
```

```
</html>
```

```
public String powrot()
```

```
{
```

```
    return "/faces/index1";
```

```
}
```

***Metoda dodana do klasy Managed\_produkт***

## Zrzut z ekranu kodu strony `rezultat2.xhtml` po zastosowaniu komponentów złożonych

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
  xmlns:komponenty_zlozone=
    "http://xmlns.jcp.org/jsf/composite/komponenty_zlozone">
  <body>
    <ui:composition template="../../../template.xhtml">
      <ui:define name="title"> #{bundle['rezultat2.tytul']} </ui:define>
      <ui:define name="content">
        <komponenty_zlozone:Wyswietlanie_produkту
          brak="false"
          brakdanych="#{bundle['rezultat2.nie_dodano']}"
          standanych1="#{managed_produkт.stan==0}"
          standanych2="#{managed_produkт.stan!=0}"
          nameLabel="#{bundle['rezultat2.nazwa_produkту']}"
          nameValue="#{managed_produkт.nazwa}"
          cenaLabel="#{bundle['rezultat2.cena']}"
          cenaValue="#{managed_produkт.cena}"
          promocjaLabel="#{bundle['rezultat2.promocja']}"
          promocjaValue="#{managed_produkт.promocja}"
          cena_bruttoLabel="#{bundle['rezultat2.cena_brutto']}"
          cena_bruttoValue="#{managed_produkт.cena_brutto}"
          pokazButtonText="#{bundle['rezultat2.powrot']}"
          pokazButtonAction="#{managed_produkт.powrot}"
        />
      </ui:define>
    </ui:composition>
  </body>
</html>
```

