

Instrukcja tworzenia aplikacji EE na bazie aplikacji prezentowanej na zajęciach lab.4 z PIO – zawierającej aplikację klienta typu EE oraz internetową aplikację uruchamianą za pomocą przeglądarki.

Projektowanie i wdrażanie systemów informatycznych

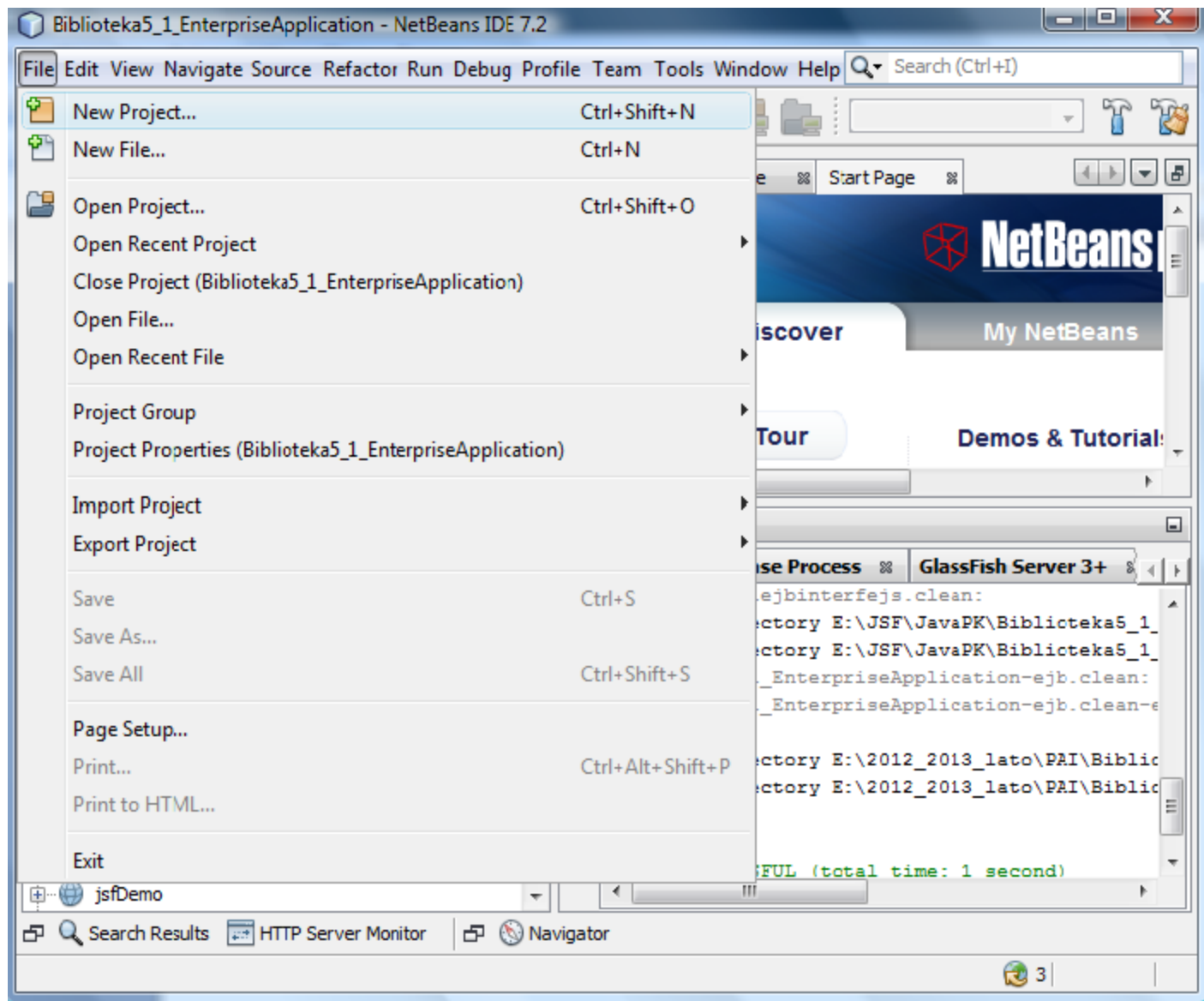
Dodanie aplikacji klienta uruchamianej przez przeglądarkę– kontynuacja projektu:

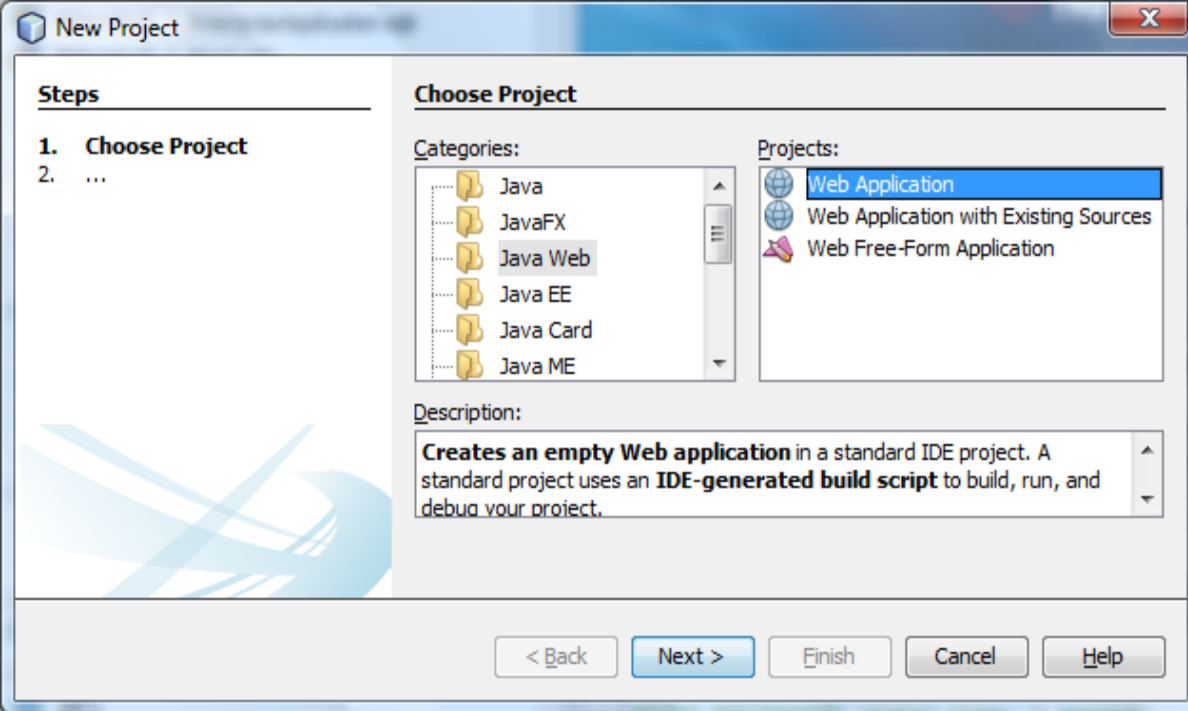
http://zofia.kruckiewicz.staff.iar.pwr.wroc.pl/wyklady/PiWSI/Budowa_aplikacjiEE_DAO.pdf

Zofia Kruckiewicz

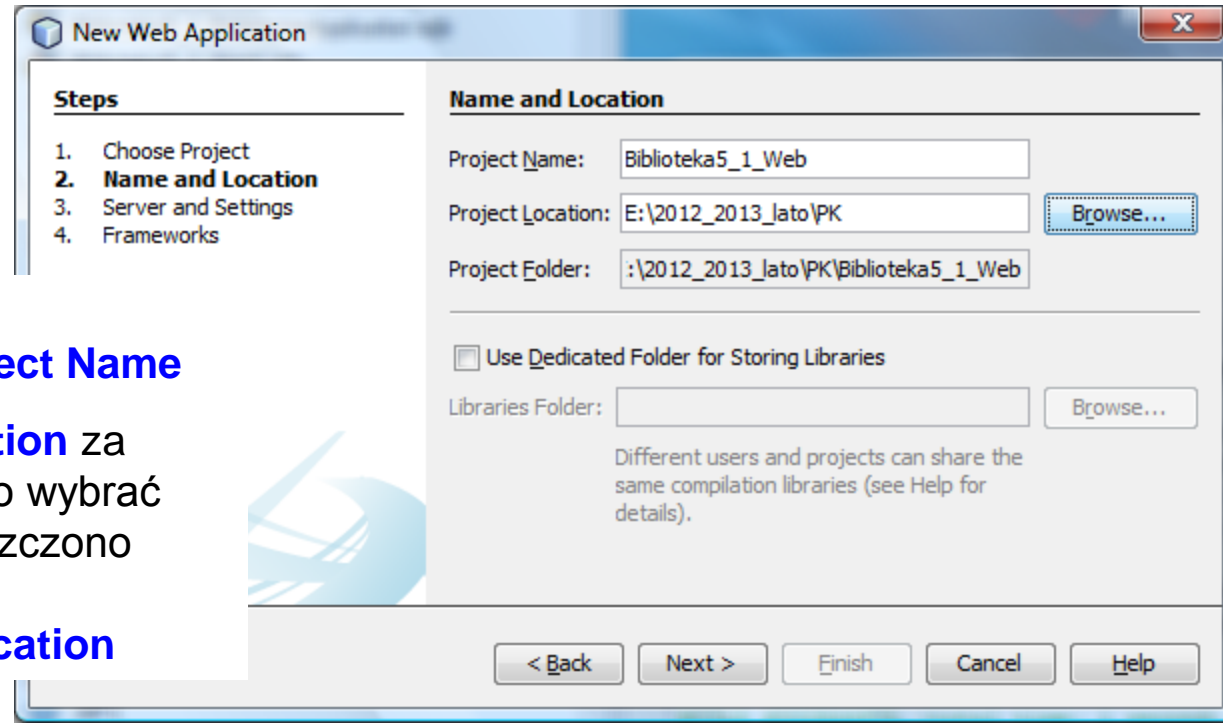
Utworzenie aplikacji internetowej
współpracującej z aplikacją typu Java EE 6

Utworzenie aplikacji internetowej – **File/New Project**





Utworzenie aplikacji internetowej –
File/New Project/Java Web/Web Application i Next

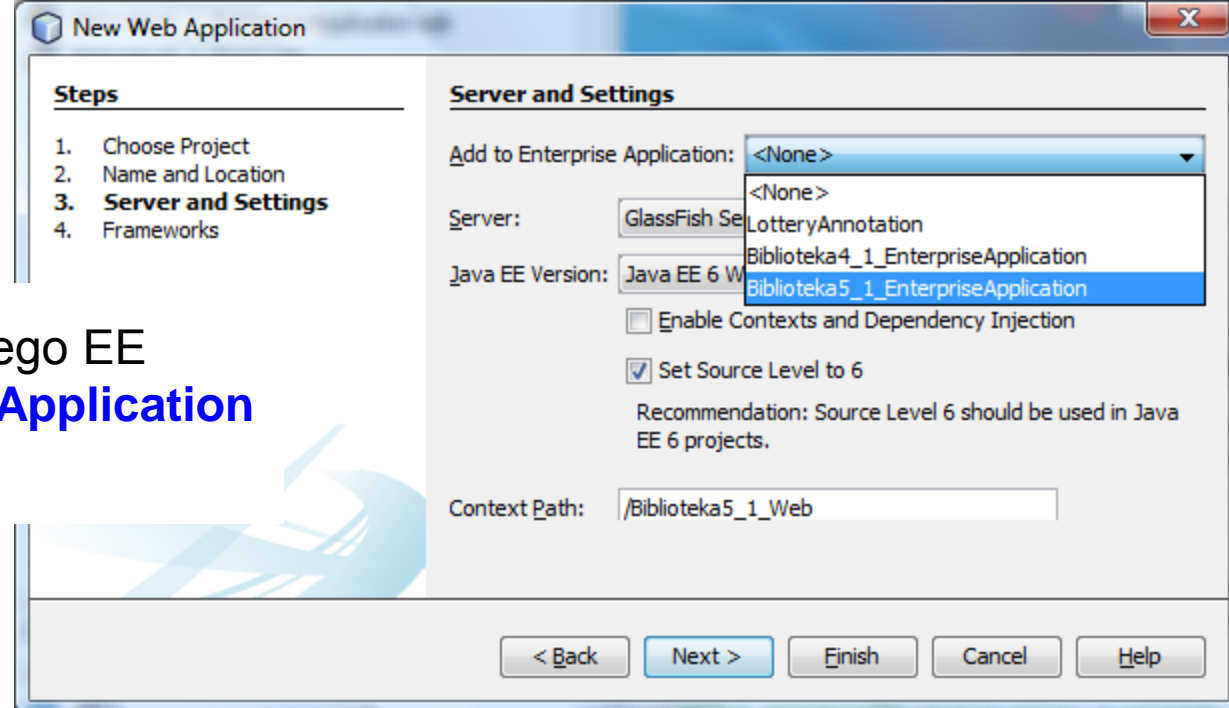


Wpisanie nazwy aplikacji
Biblioteka5_1_Web w polu **Project Name**

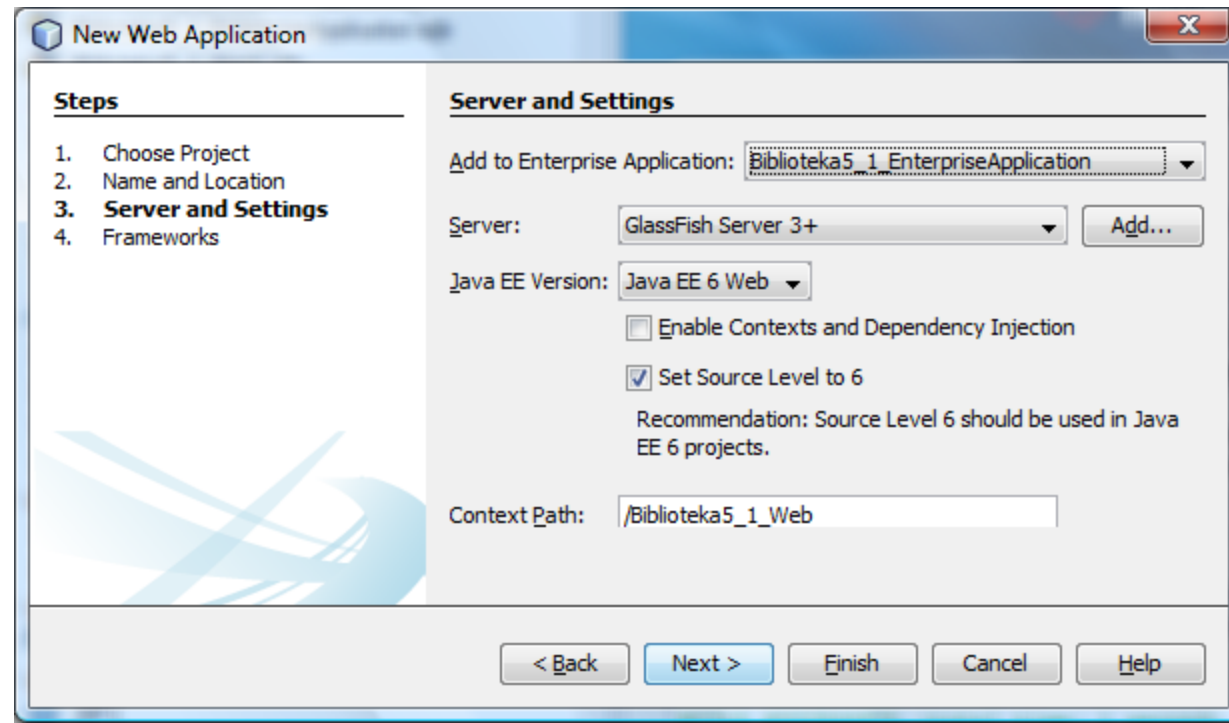
Należy wyznaczyć **Project Location** za pomocą klaswisz **Browse** – warto wybrać ten sam katalog, w którym umieszczono aplikacje projektu EE

Biblioteka5_1_EnterpriseApplication

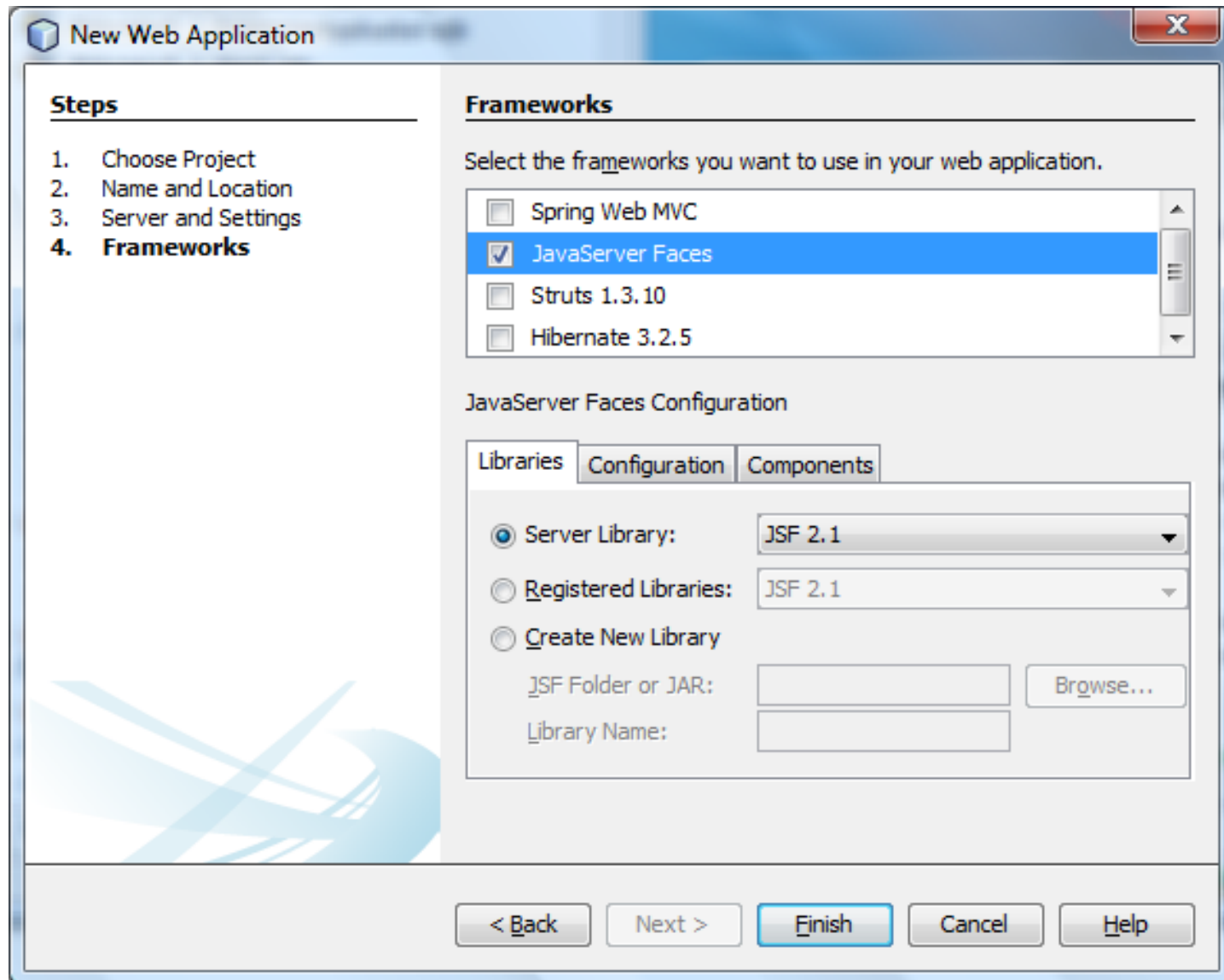
Wybór projektu głównego EE
Biblioteka5_1_EnterpriseApplication



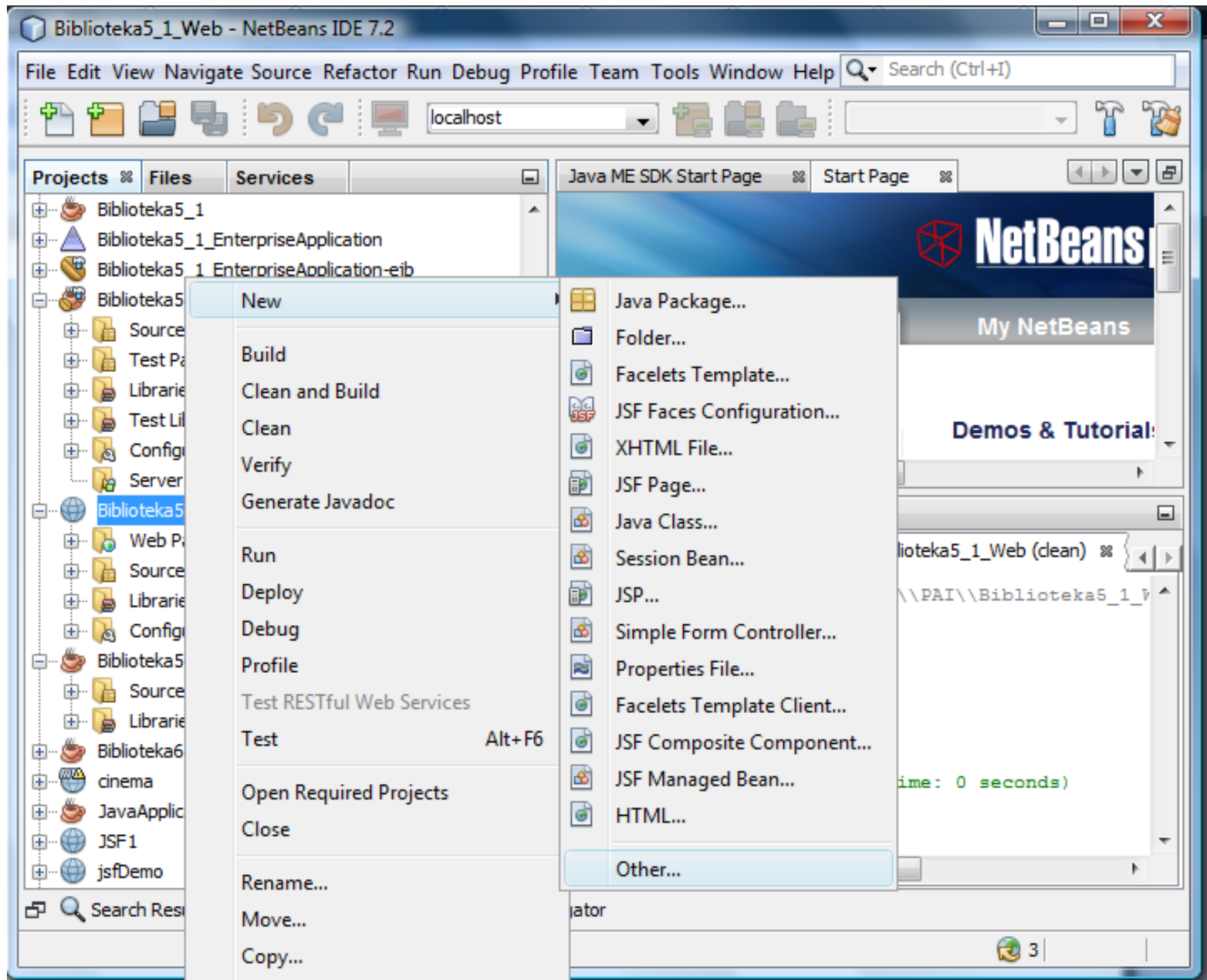
Wybór serwera i wersji
platformy Javy – Java EE 6



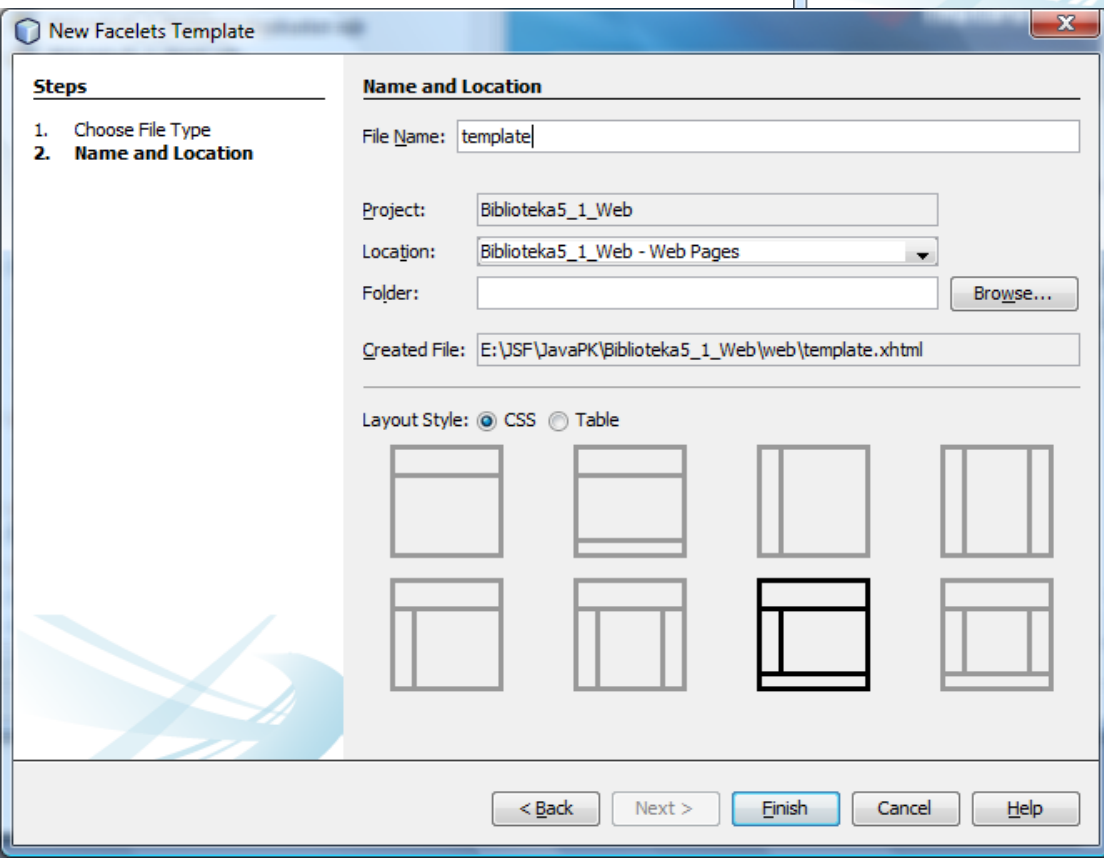
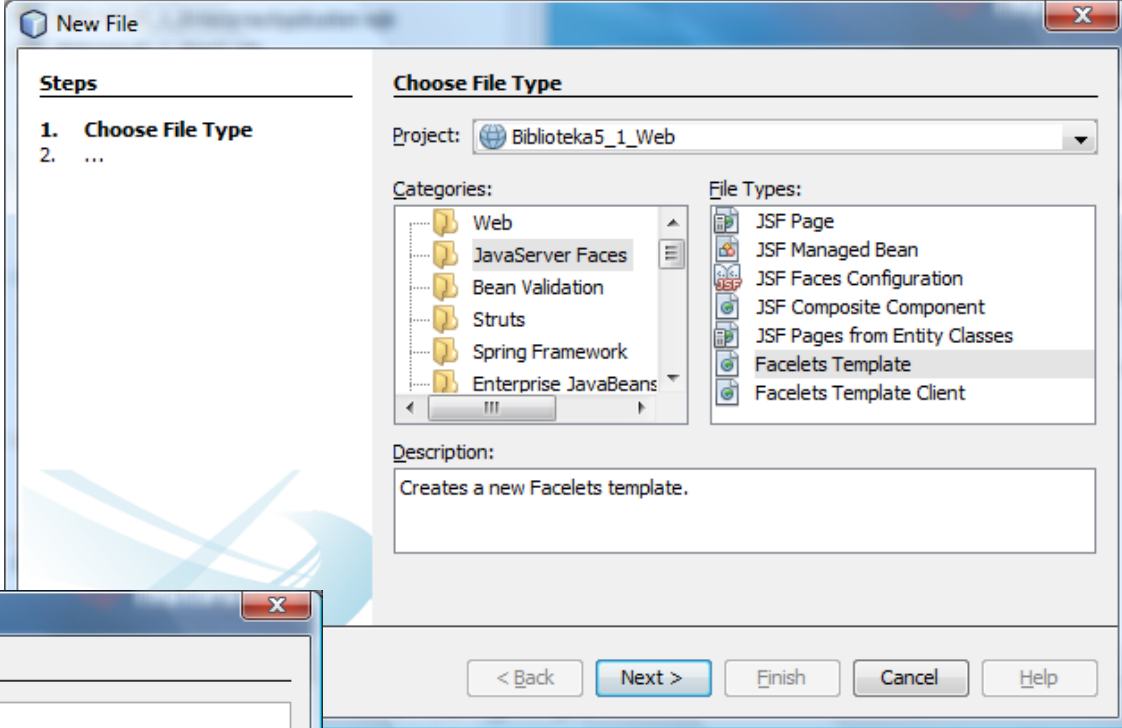
Wybór środowiska JSF 2.1 aplikacji internetowej



Dodanie do aplikacji szablonu stron – prawym klawiszem kliknąć na nazwę projektu i wybór **New/Other**

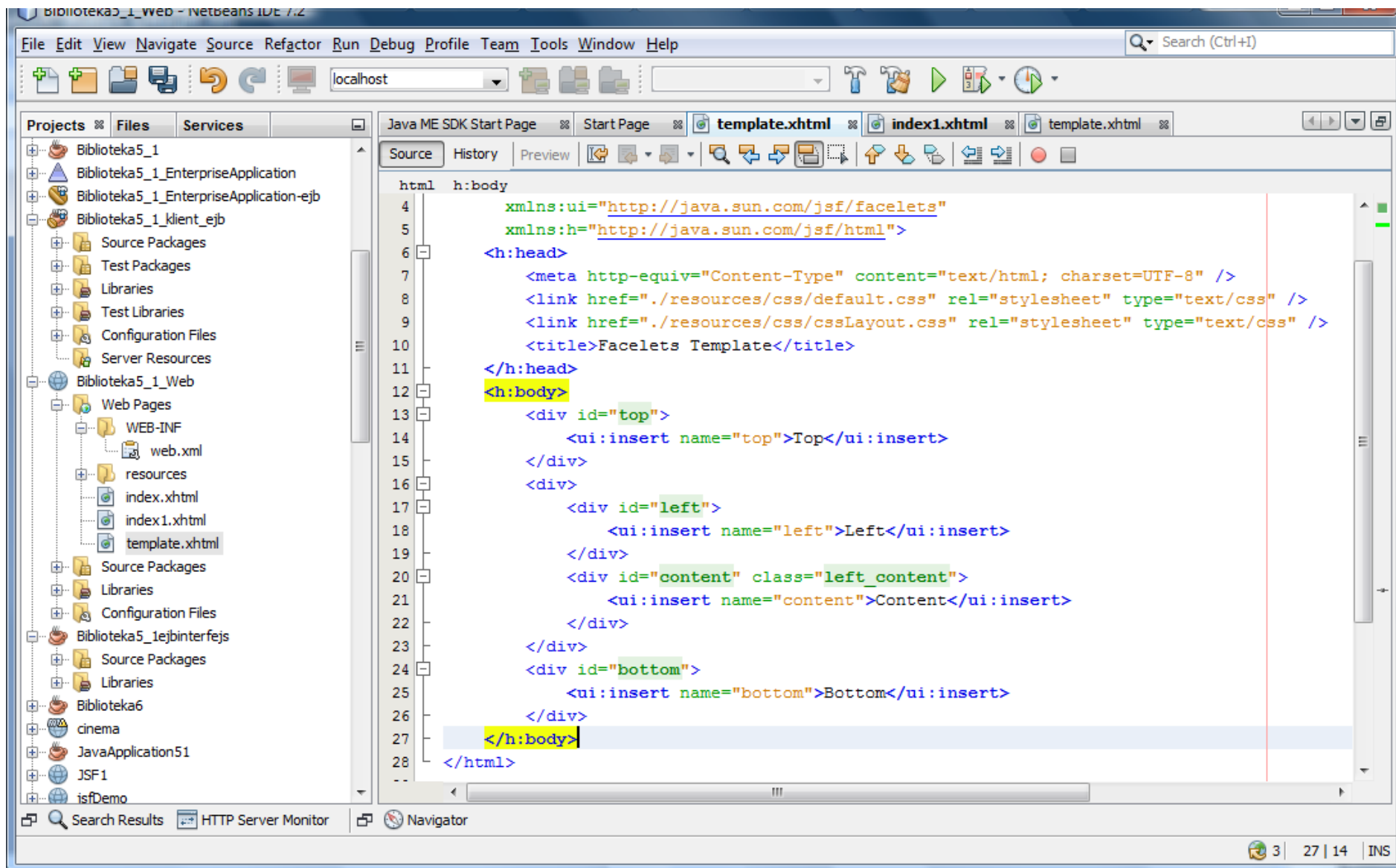


Wybór pliku JavaServer Faces/Facelets template i Next



Wybór szablonu z palety
oraz podanie nazwy
szablonu w polu **File Name**

Utworzony szablon stron JSF o nazwie **template.xhtml**



The screenshot displays the NetBeans IDE interface for a project named "Biblioteka5_1_Web". The left-hand "Projects" pane shows the project structure, with "template.xhtml" highlighted under the "Web Pages" folder. The main editor window shows the source code of "template.xhtml", which is an HTML page using JSF Facelets. The code includes XML namespace declarations for JSF and HTML, a head section with meta and link tags, and a body section with four named regions: "top", "left", "content", and "bottom". Each region contains a JSF insert tag with a corresponding text label.

```
html h:body
4      xmlns:ui="http://java.sun.com/jsf/facelets"
5      xmlns:h="http://java.sun.com/jsf/html">
6      <h:head>
7          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
8          <link href="./resources/css/default.css" rel="stylesheet" type="text/css" />
9          <link href="./resources/css/cssLayout.css" rel="stylesheet" type="text/css" />
10         <title>Facelets Template</title>
11     </h:head>
12     <h:body>
13         <div id="top">
14             <ui:insert name="top">Top</ui:insert>
15         </div>
16         <div>
17             <div id="left">
18                 <ui:insert name="left">Left</ui:insert>
19             </div>
20             <div id="content" class="left_content">
21                 <ui:insert name="content">Content</ui:insert>
22             </div>
23         </div>
24         <div id="bottom">
25             <ui:insert name="bottom">Bottom</ui:insert>
26         </div>
27     </h:body>
28 </html>
```

Zmodyfikowana zawartość strony szablonu **template.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:ui="http://java.sun.com/jsf/facelets"
  xmlns:h="http://java.sun.com/jsf/html">

<h:head>
  <meta http-equiv="Content-Type" content="text/html;
    charset=UTF-8" />
  <h:outputStylesheet name="css/default.css" />
  <h:outputStylesheet name="css/cssLayout.css"/>
  <title><ui:insert name="title">Facelets Template</ui:insert></title>
</h:head>

<h:body>
  <div id="top">
    <ui:insert name="top">Top</ui:insert>
  </div>
```

Definicja własna tytułu każdej strony opartej na szablonie **template.xhtml** określona za pomocą znacznika **<ui:define**

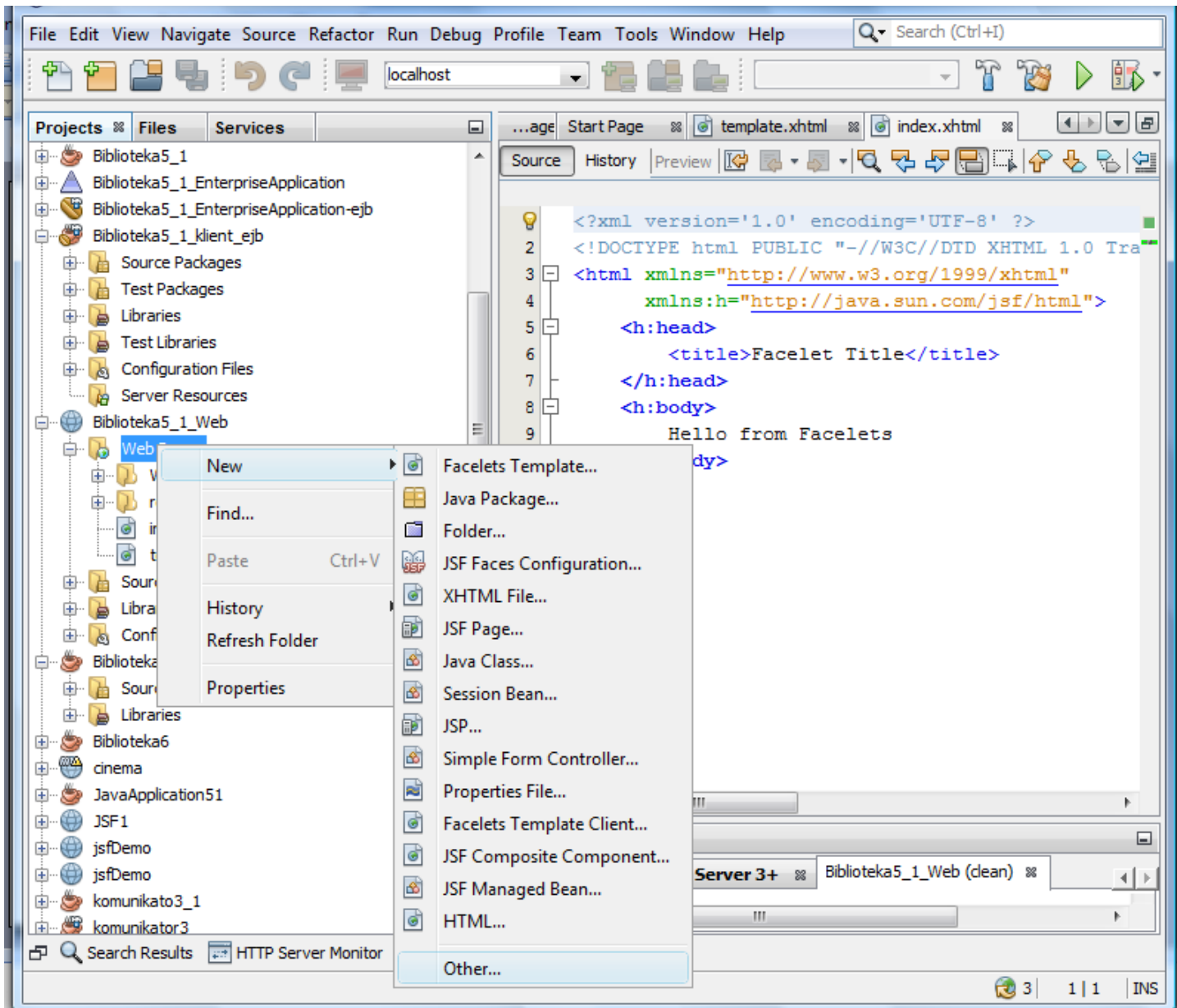
```
<div>
  <div id="left">
    <h:link outcome="/faces/jsf/dodaj_tytul"
      value="dodaj tytul"/> <br/>
    <h:link outcome="/faces/jsf/lista_tytulow"
      value="wyswietl tytuly"/>
  </div>
  <div id="content" class="left_content">
    <ui:insert name="content">Content</ui:insert>
  </div>
</div>
<div id="bottom">
  <ui:insert name="bottom">Bottom</ui:insert>
</div>
</h:body>
</html>
```

Część strony, używana na każdej stronie, powinna być zdefiniowana w szablonie - czyli linki pozwalające wybierać inne strony aplikacji

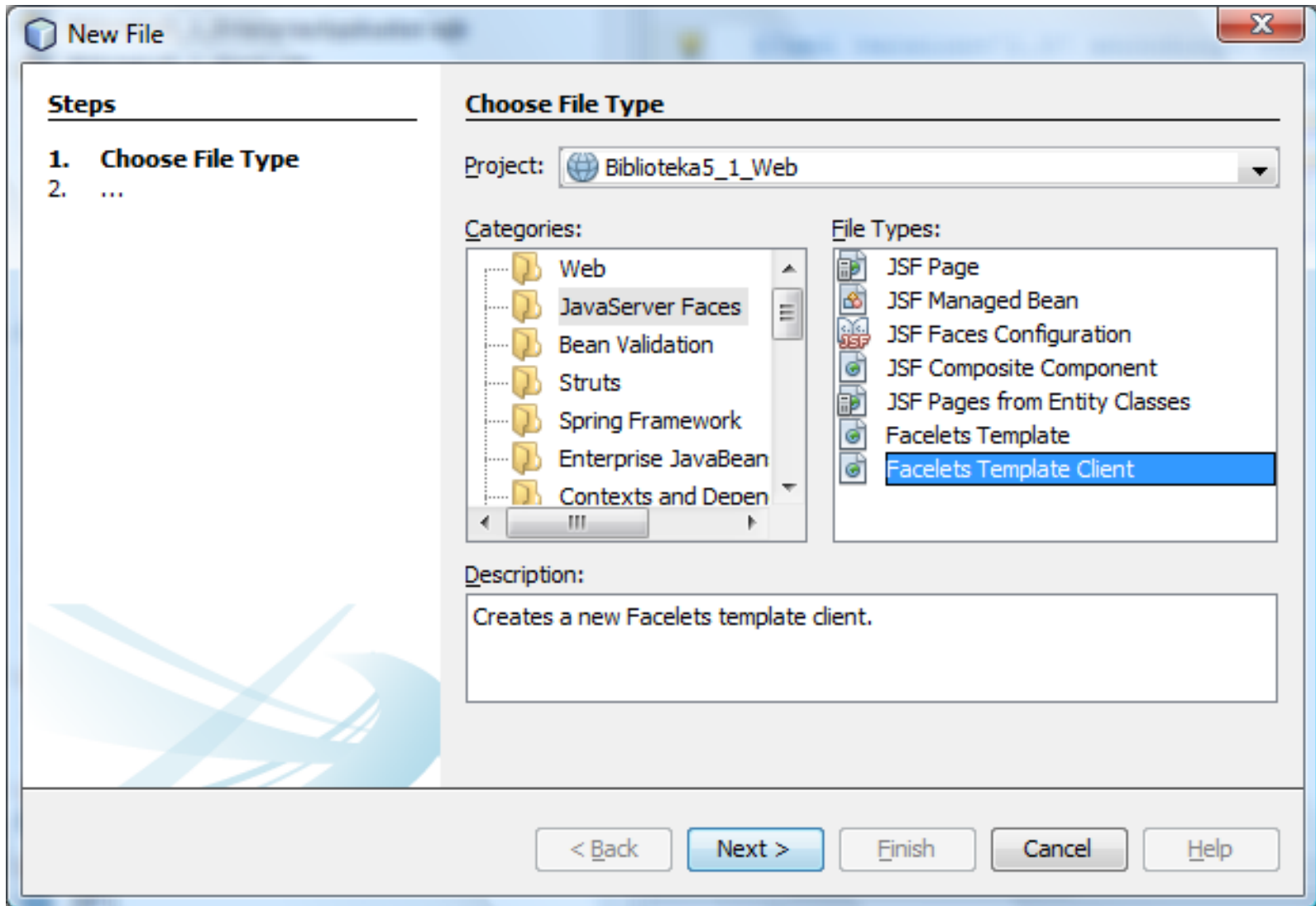
Definicja własna **content** każdej strony opartej na szablonie **template.xhtml** określona za pomocą znacznika **<ui:define**

Pozostałe znaczniki **<ui:insert (top, bottom)** nie zostały zdefiniowane na stronach opartych na szablonie **template.xhtml**

Wstawienie nowej strony startowej typu JSF jako **Facelets template client** -
prawym klawiszem kliknąć na nazwę projektu i wybór **New/Other**



Wybór typu strony – **JavaServer Faces/Facelets Template Client**



Nadanie nazwy **index1** stronie JSF w polu **File Name**

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name:

Project:

Folder:

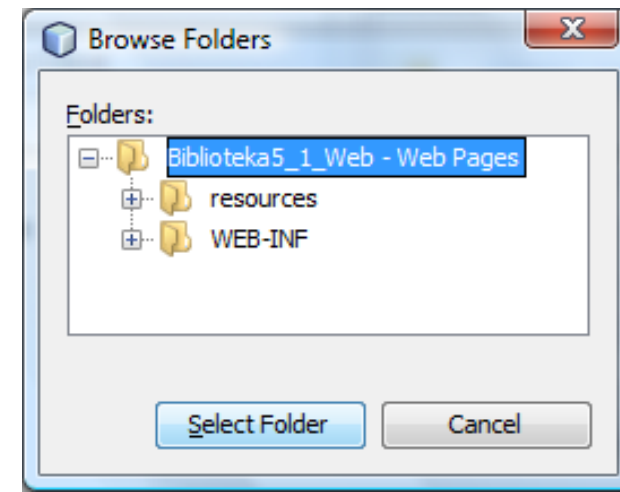
Created File:

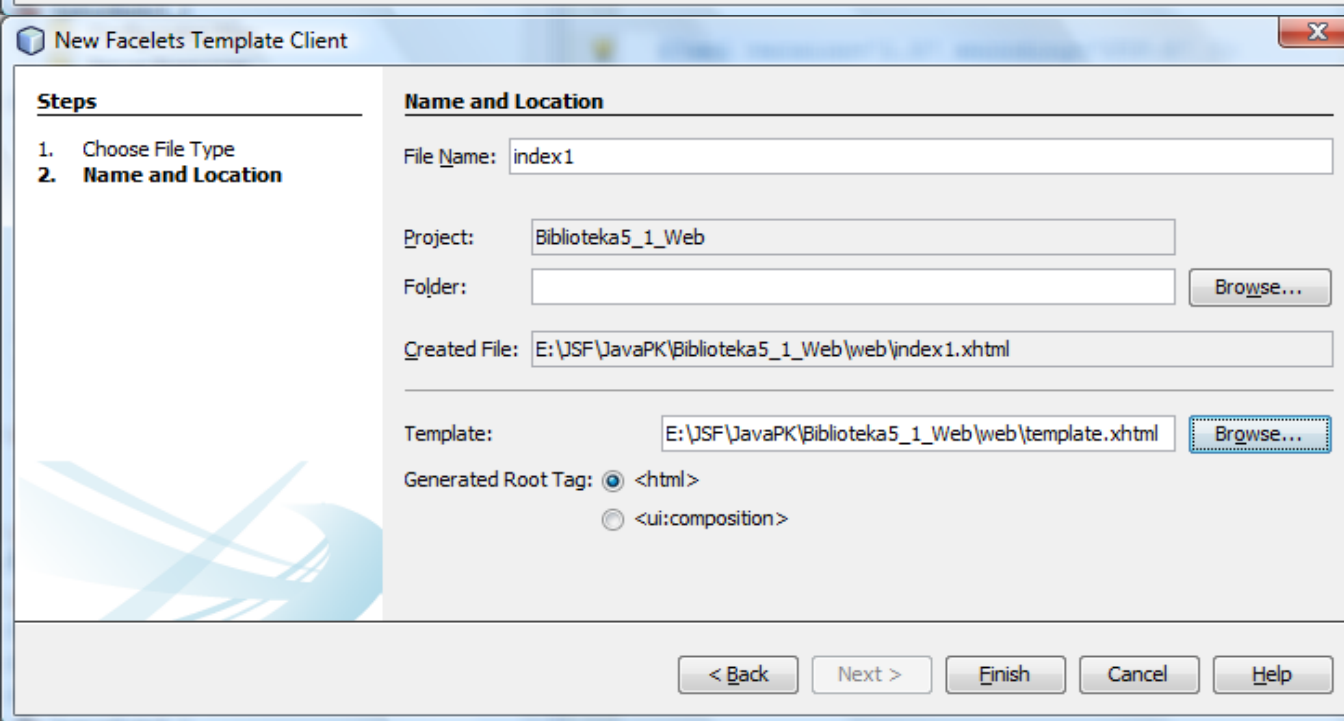
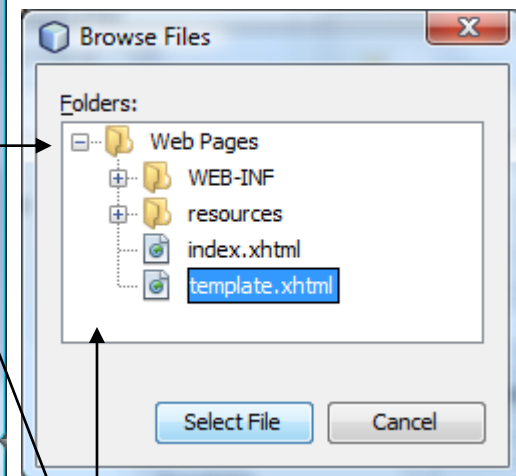
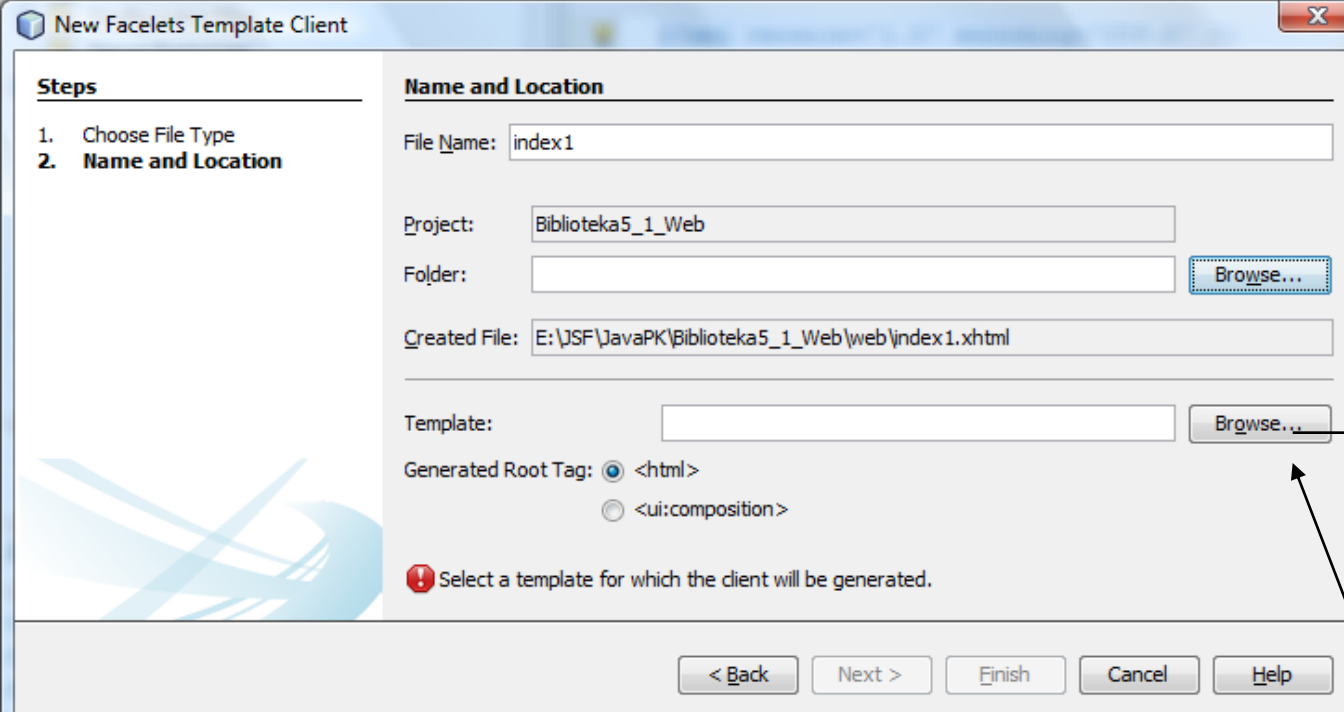
Template:

Generated Root Tag: <html>
 <ui:composition>

Select a template for which the client will be generated.

Wybór położenia strony **index1** w katalogu **Web Pages** w polu **Folder** za pomocą klawisza **Browse**





Wybór szablonu, wg którego powinna być zbudowana strona **index1**, w polu **Template** za pomocą klawisza **Browse**

Utworzona nowa strona JSF o nazwie **index1** jako **Facelets Template Client**

The screenshot displays the NetBeans IDE 7.2 interface. The main editor window shows the XML code for a new page named `index1.xhtml`. The code is as follows:

```
html body ui:composition
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:ui="http://java.sun.com/jsf/facelets">
5 <body>
6     <ui:composition template="./template.xhtml">
7         <ui:define name="top">
8             top
9         </ui:define>
10        <ui:define name="left">
11            left
12        </ui:define>
13        <ui:define name="content">
14            content
15        </ui:define>
16        <ui:define name="bottom">
17            bottom
18        </ui:define>
19    </ui:composition>
20 </body>
21 </html>
22
```

The left sidebar shows the project structure for `Biblioteka5_1_Web`, including `Web Pages`, `WEB-INF`, `resources`, `index.xhtml`, `index1.xhtml`, and `template.xhtml`. The bottom status bar indicates the current page is 3 out of 26, and the IDE is in the `INS` (Insert) state.

Modyfikowany kod strony **index1.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: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">
      <h:outputText value="#{bundle['index1.tytul']}"></h:outputText>
    </ui:define>
  </ui:composition>

</body>
</html>
```

Definicja własna tytułu
strony opartej na
szablonie
template.xhtml
określona za pomocą
znacznika **<ui:define**

Pozostałe części strony
(**top, left, content i
bottom**) oparte są na
definicjach zawartych w
szablonie

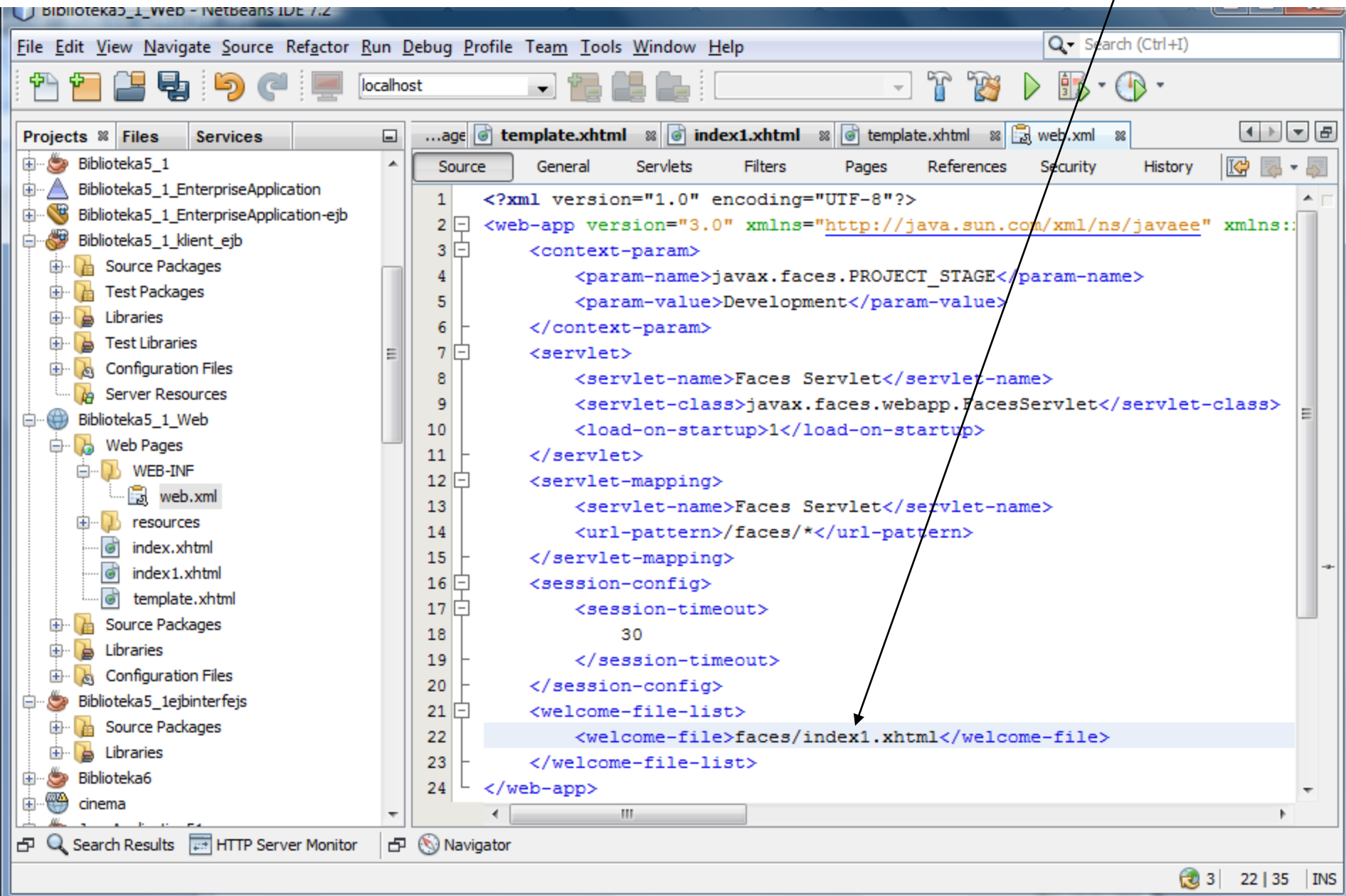
Strona **index1.xhtml** po modyfikacji

The screenshot displays the NetBeans IDE 7.2 interface. The main editor window shows the source code of `index1.xhtml`. The code is as follows:

```
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:ui="http://java.sun.com/jsf/facelets"
5       xmlns:h="http://java.sun.com/jsf/html">
6   <body>
7     <ui:composition template="./template.xhtml">
8       <ui:define name="title">
9         <h:outputText value="#{bundle['index1.tytul']}"></h:outputText>
10      </ui:define>
11    </ui:composition>
12  </body>
13 </html>
14
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help), a toolbar with various icons, and a project explorer on the left showing the file structure of `Biblioteka5_1_Web`. The project explorer highlights the `index1.xhtml` file under the `resources` folder. The status bar at the bottom indicates the current page is 9 of 54, and the language is set to INS.

Zadeklarowanie strony **index1** jako strony startowej w pliku **web.xml** (deskryptor aplikacji internetowej EE)

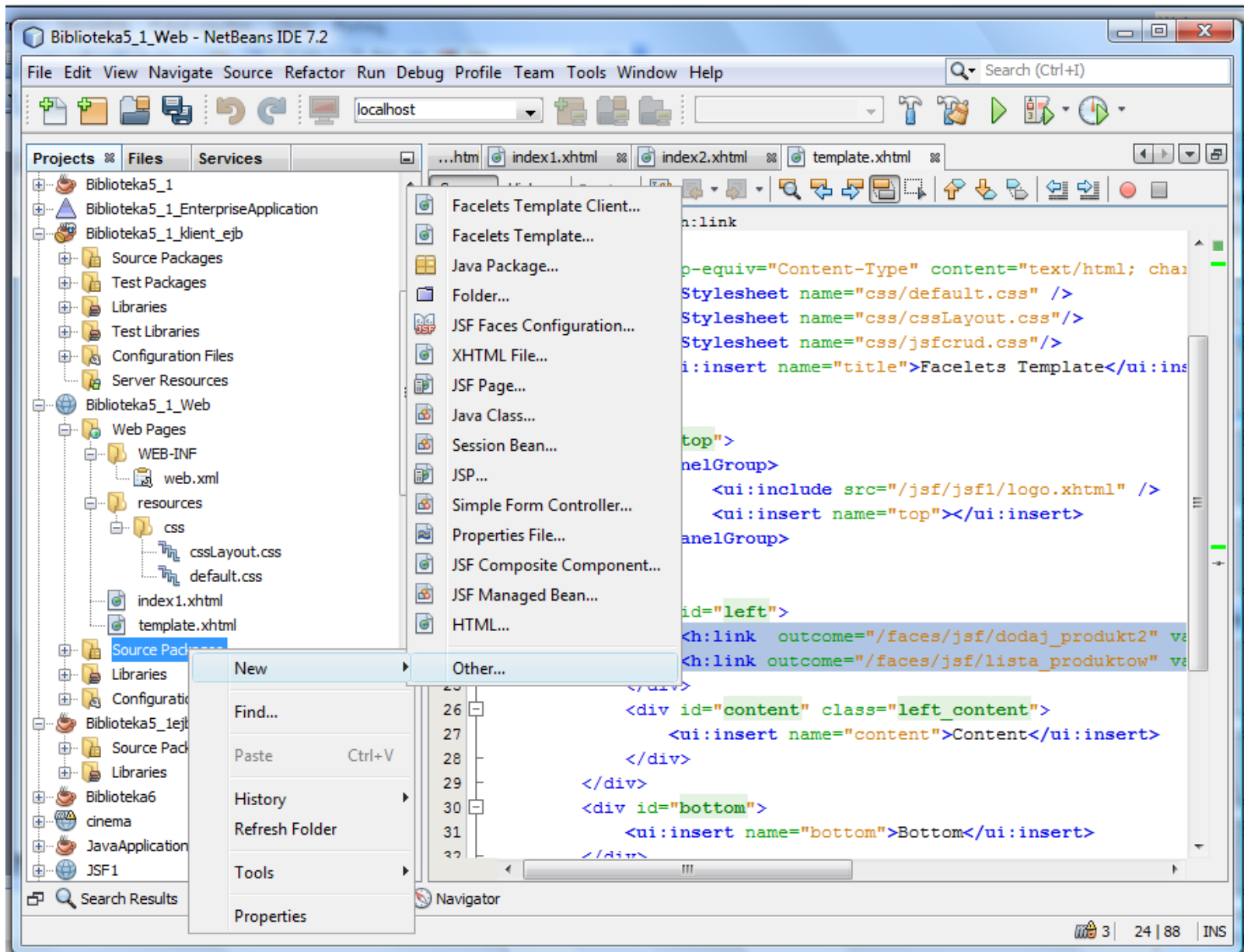


The screenshot shows the NetBeans IDE interface with the `web.xml` file open in the editor. The file is configured for a Java EE web application. The `<welcome-file>` element is highlighted, and an arrow points from the title of the slide to this element.

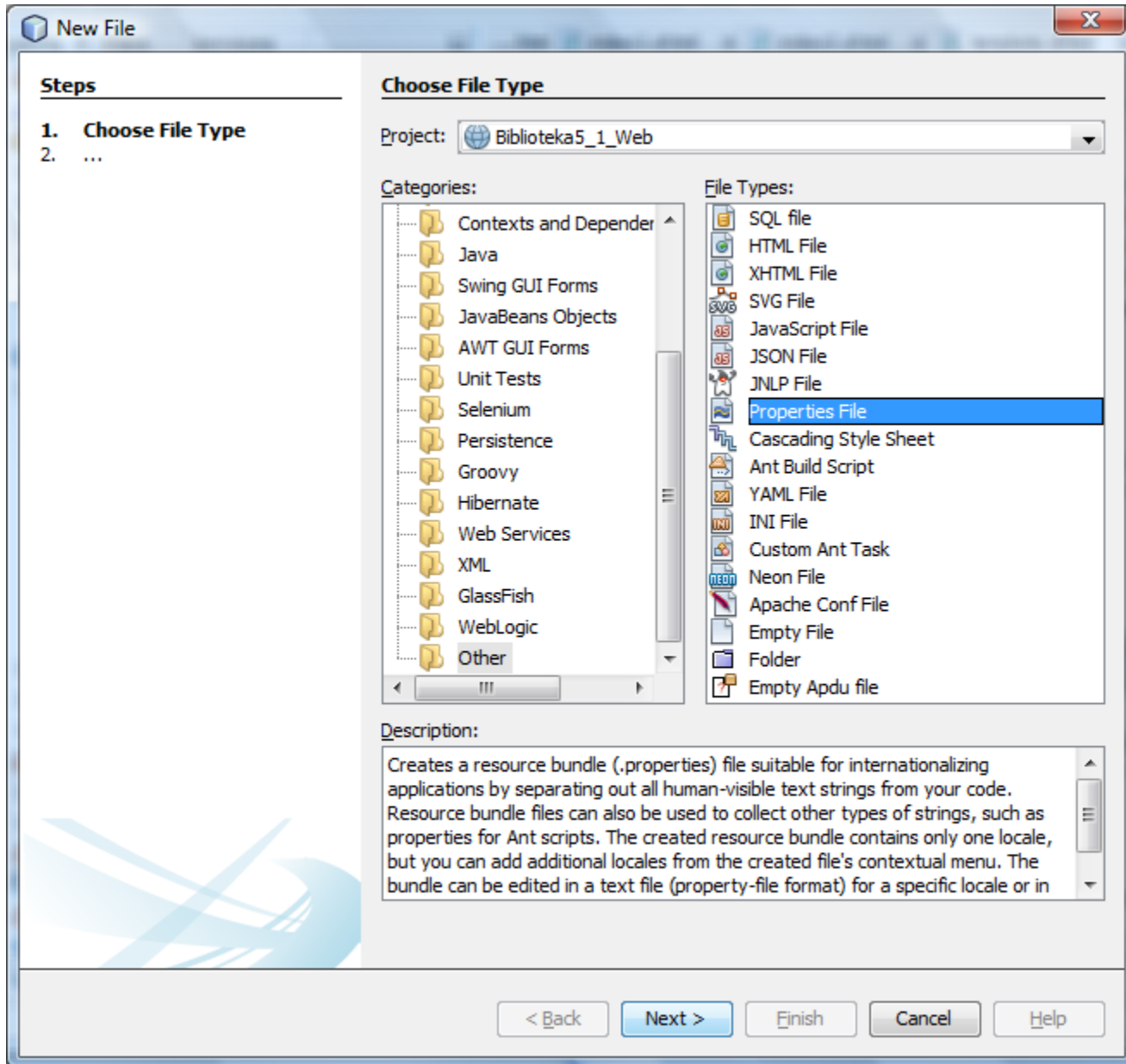
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:
3   <context-param>
4     <param-name>javax.faces.PROJECT_STAGE</param-name>
5     <param-value>Development</param-value>
6   </context-param>
7   <servlet>
8     <servlet-name>Faces Servlet</servlet-name>
9     <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
10    <load-on-startup>1</load-on-startup>
11  </servlet>
12  <servlet-mapping>
13    <servlet-name>Faces Servlet</servlet-name>
14    <url-pattern>/faces/*</url-pattern>
15  </servlet-mapping>
16  <session-config>
17    <session-timeout>
18      30
19    </session-timeout>
20  </session-config>
21  <welcome-file-list>
22    <welcome-file>faces/index1.xhtml</welcome-file>
23  </welcome-file-list>
24 </web-app>
```

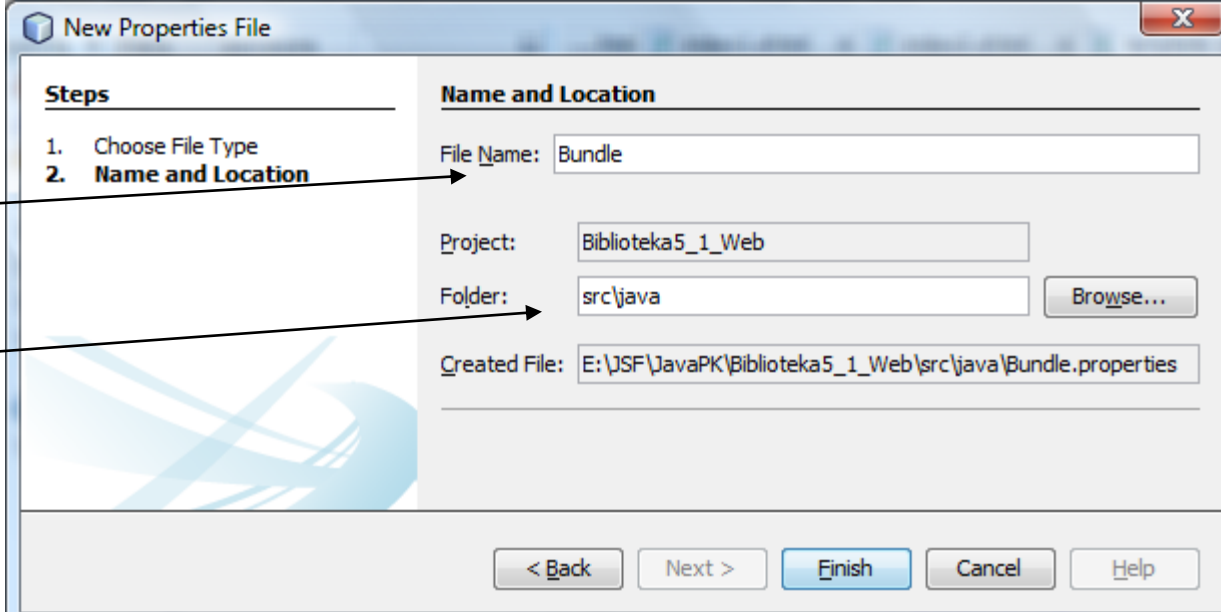
Wstawienie pliku typu **properties**
(**Bundle.properties**) do definiowania w
elastyczny sposób komunikatów
wyświetlanych na stronie oraz pliku **faces-**
config.xml do konfigurowania aplikacji JSF

Wstawienie pliku typu properties do przechowywania komunikatów etykiet na stronie JSF- prawym klawiszem kliknąć na nazwę projektu i wybór **New/Other**

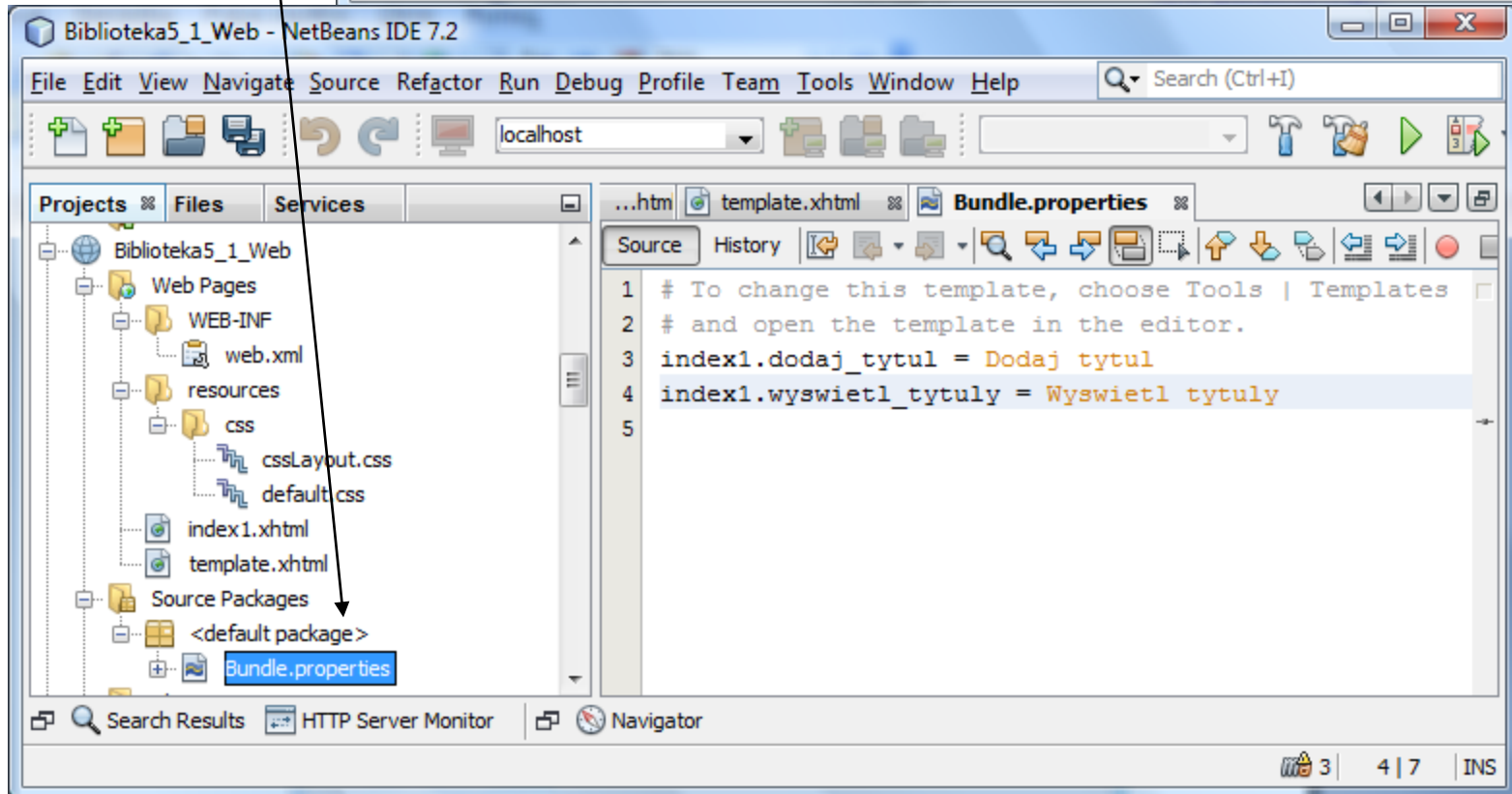


Wybór Other/Properties File





Podanie nazwy **Bundle** w polu **File Name** oraz położenia w folderze `src/java` (Source Package / <default package>)



Zawartość pliku **Bundle.properties**

To change this template, choose Tools | Templates and open the template in the editor.

index1.dodaj_tytul = Dodaj tytul

index1.wyswietl_tytuly = Wyswietl tytuly

index1.tytul = Biblioteka 5_1

jsf.dodaj_tytul.tytul = Dodawanie tytulow

jsf.dodaj_tytul.podaj_tytul = Podaj tytul ksiazki

jsf.dodaj_tytul.podaj_tytul_blad = Blad! Nie podano tytulou ksiazki

jsf.dodaj_tytul.podaj_nazwisko = Podaj nazwisko autora

jsf.dodaj_tytul.podaj_nazwisko_blad = Blad! Nie podano nazwiska autora

jsf.dodaj_tytul.podaj_imie = Podaj imie autora

jsf.dodaj_tytul.podaj_imie_blad = Blad! Nie podano imienia autora

jsf.dodaj_tytul.podaj_ISBN = Podaj ISBN tytulou

jsf.dodaj_tytul.podaj_ISBN_blad = Blad! Nie podano ISBN tytulou

jsf.dodaj_tytul.podaj_Wydawnictwo = Podaj wydawnictwo

jsf.dodaj_tytul.podaj_Wydawnictwo_blad = Blad! Nie podano wydawnictwa

jsf.dodaj_tytul.akcja = OK

jsf.lista_tytulow.tytul = Tytuly ksiazek

jsf.lista_produktow.pusta = Lista tytu\u0142ow jest pusta

jsf.lista_tytulow.tytul = Tytul ksiazki

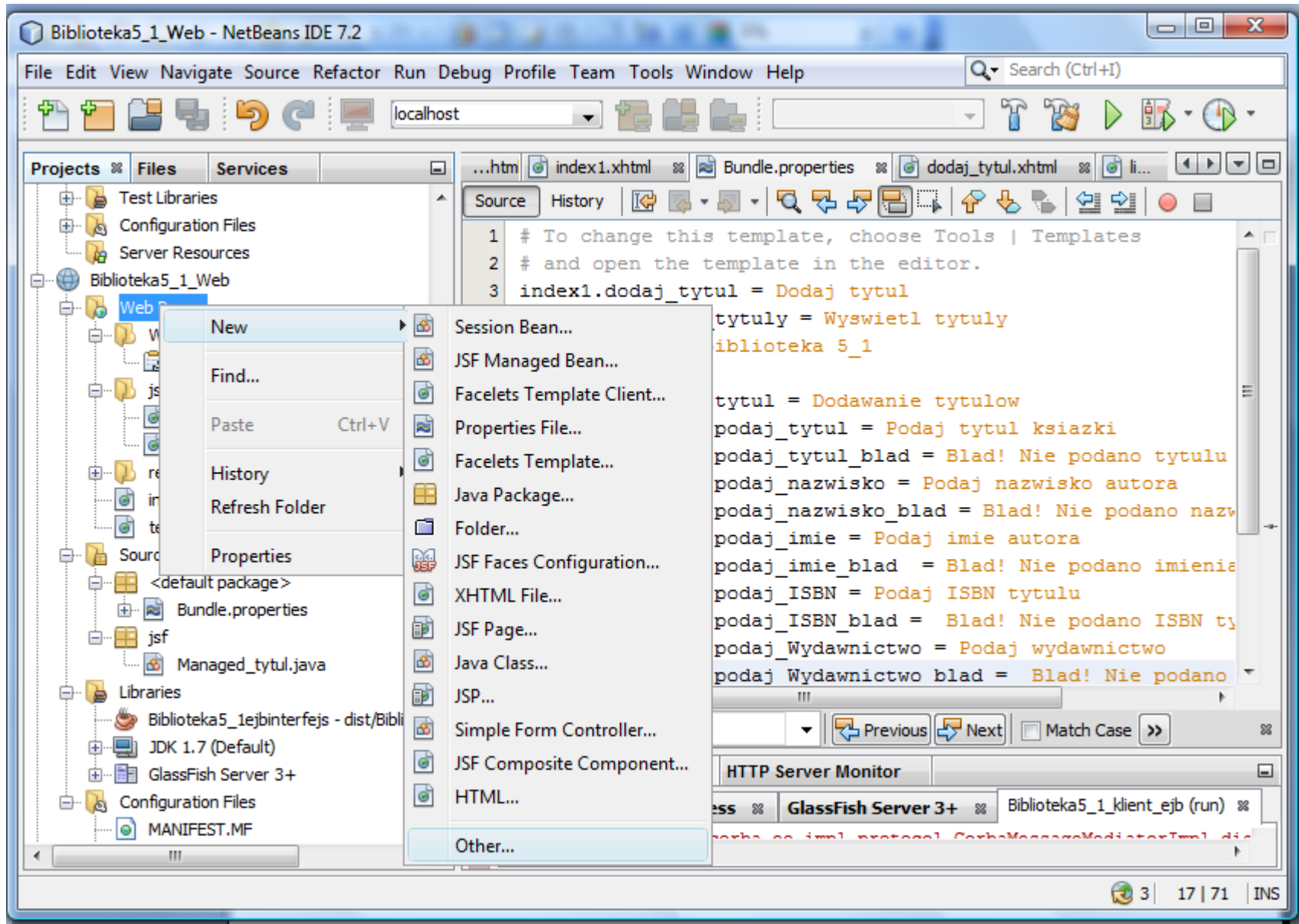
jsf.lista_tytulow.nazwisko = Nazwisko autora

jsf.lista_tytulow.imie = Imie autora

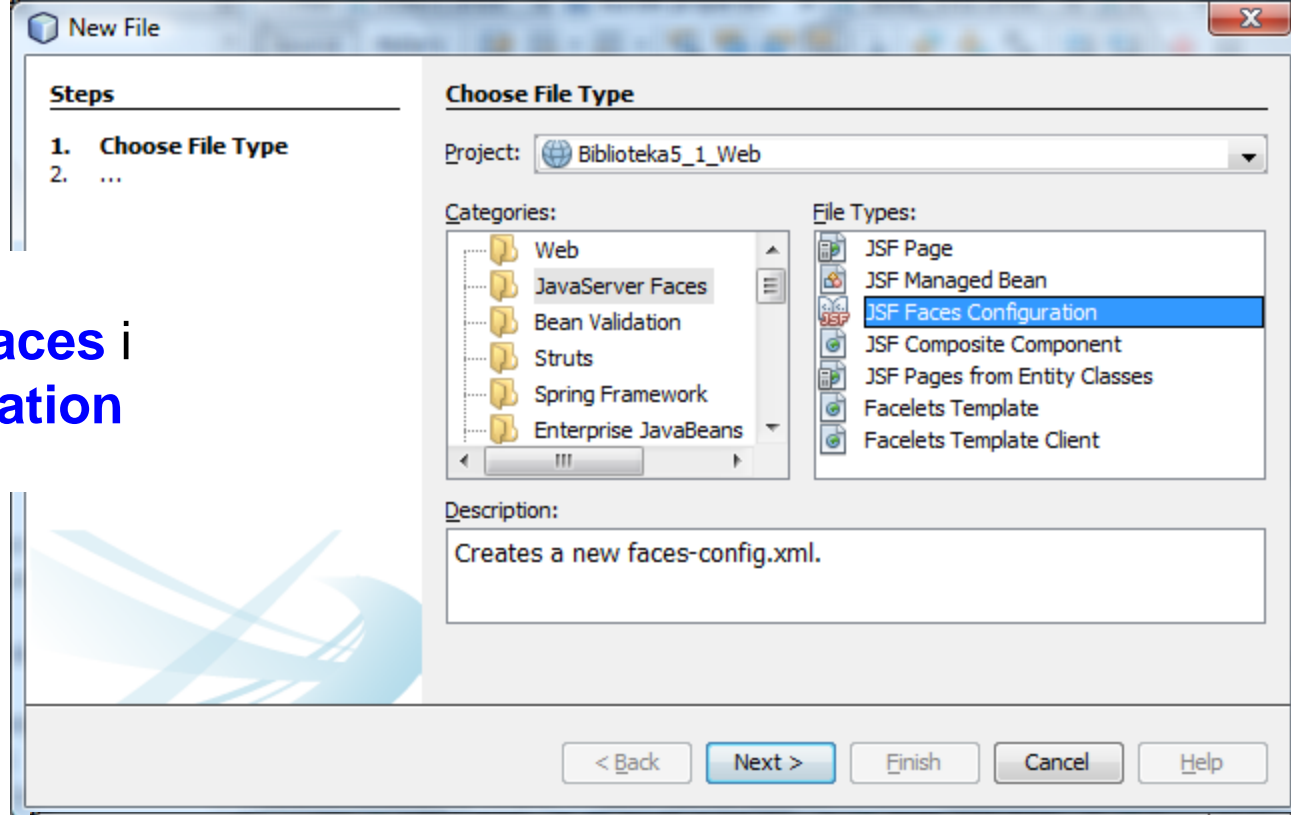
jsf.lista_tytulow.isbn = ISBN tytulou ksiazki

jsf.lista_tytulow.wydawnictwo = Wydawnictwo

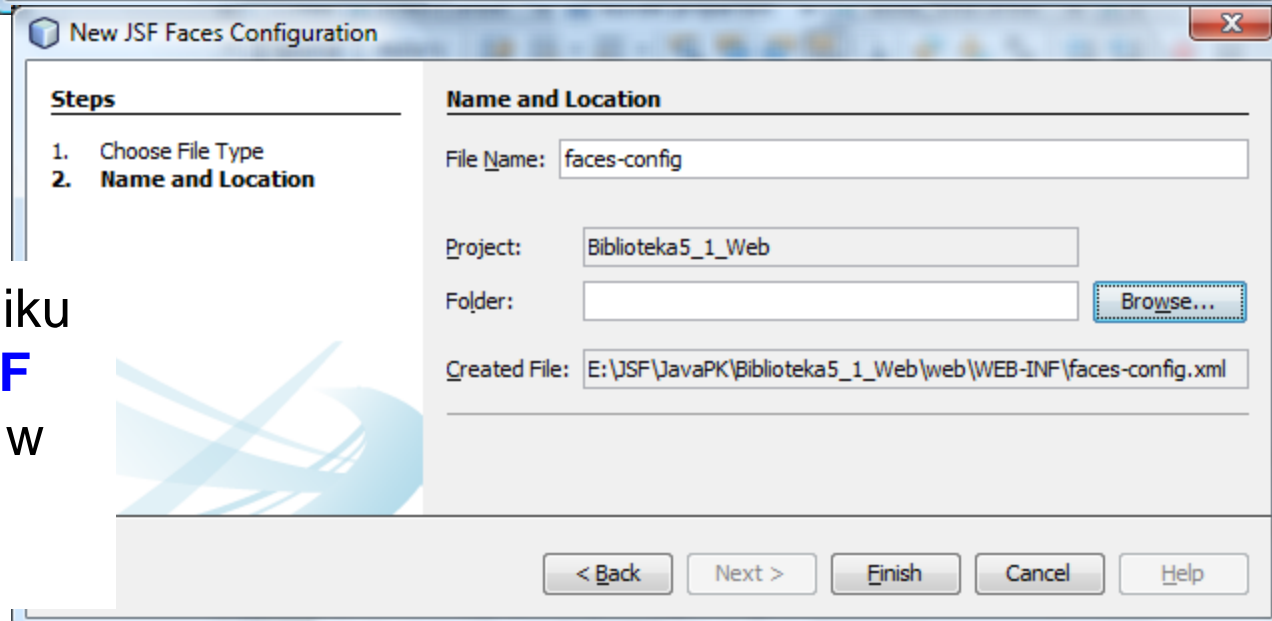
Dodanie pliku konfiguracyjnego JSF faces-config.xml - prawym klawiszem kliknąć na nazwę projektu i wybór **New/Other**



Wybór **JavaServer Faces** i **JSF Faces Configuration**



Zatwierdzenie nazwy pliku
faces-config typu **JSF
Faces Configuration** w
domyślnym katalogu
Web/WEB-INF



Zawartość pliku **faces-config.xml** należy uzupełnić zaznaczoną treścią

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<!-- ===== FULL CONFIGURATION FILE  
===== -->
```

```
<faces-config version="2.1"  
  xmlns="http://java.sun.com/xml/ns/javaee"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  
http://java.sun.com/xml/ns/javaee/web-facesconfig_2_1.xsd">
```

```
<application>
```

```
<resource-bundle>
```

```
<base-name>/Bundle</base-name>
```

```
<var>bundle</var>
```

```
</resource-bundle>
```

```
<locale-config>
```

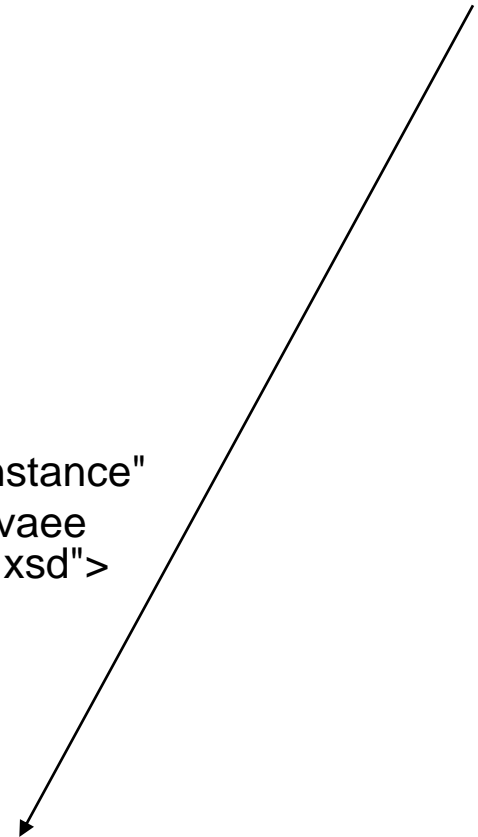
```
<default-locale>pl_PL</default-locale>
```

```
<supported-locale>en_US</supported-locale>
```

```
</locale-config>
```

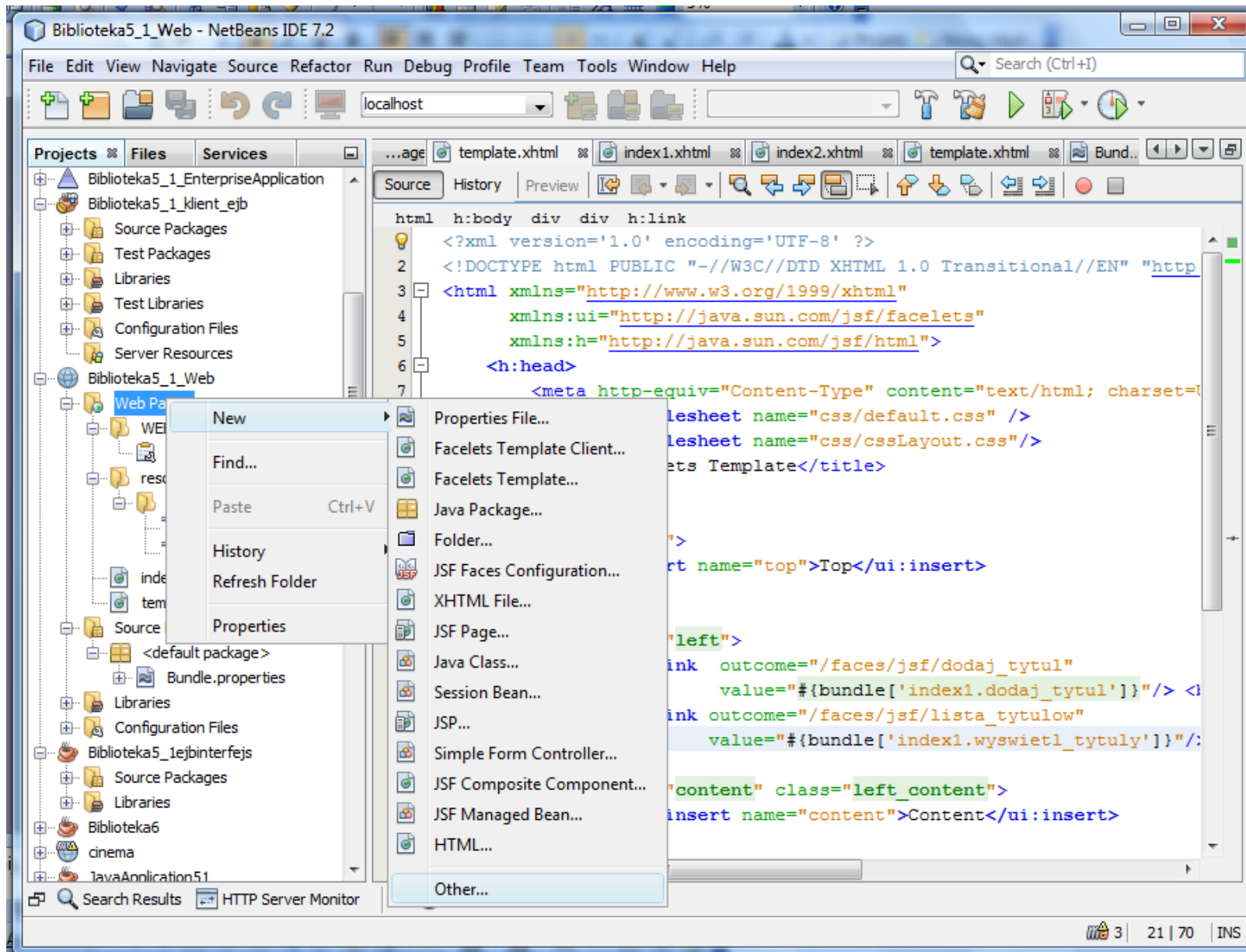
```
</application>
```

```
</faces-config>
```

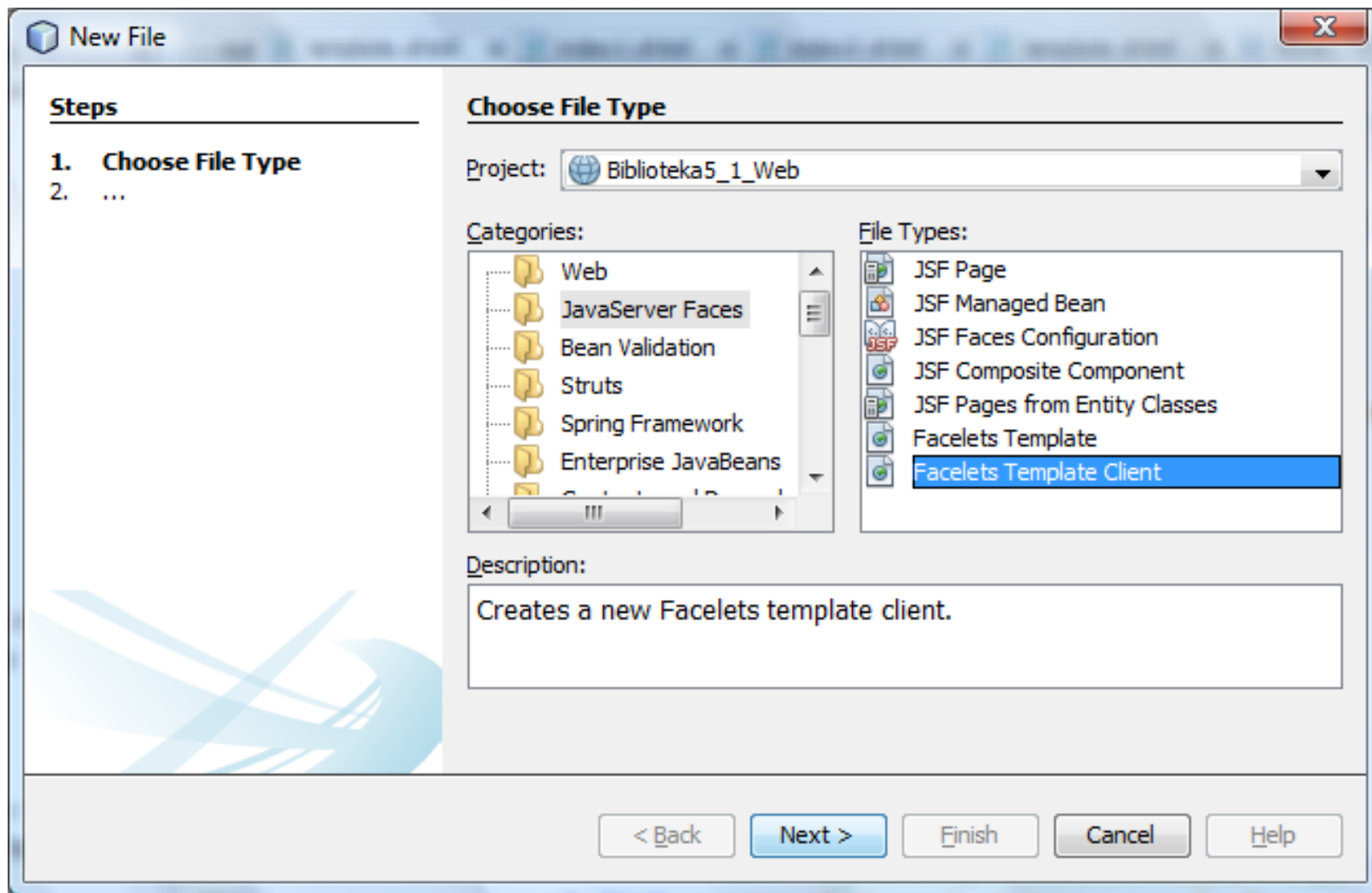


Dodanie dwóch stron obsługujących
podstawowe dwa przypadki użycia aplikacji:
wstawianie tytułów i wyświetlanie tytułów

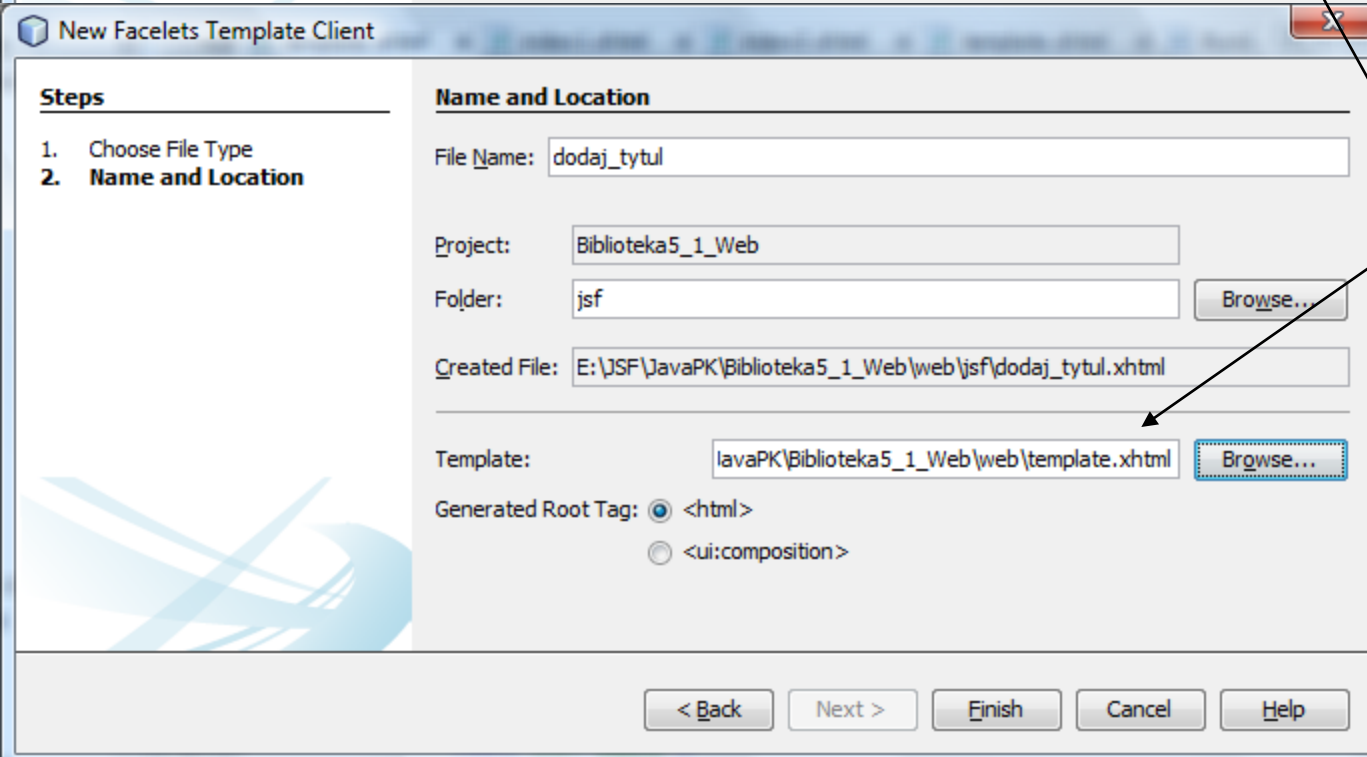
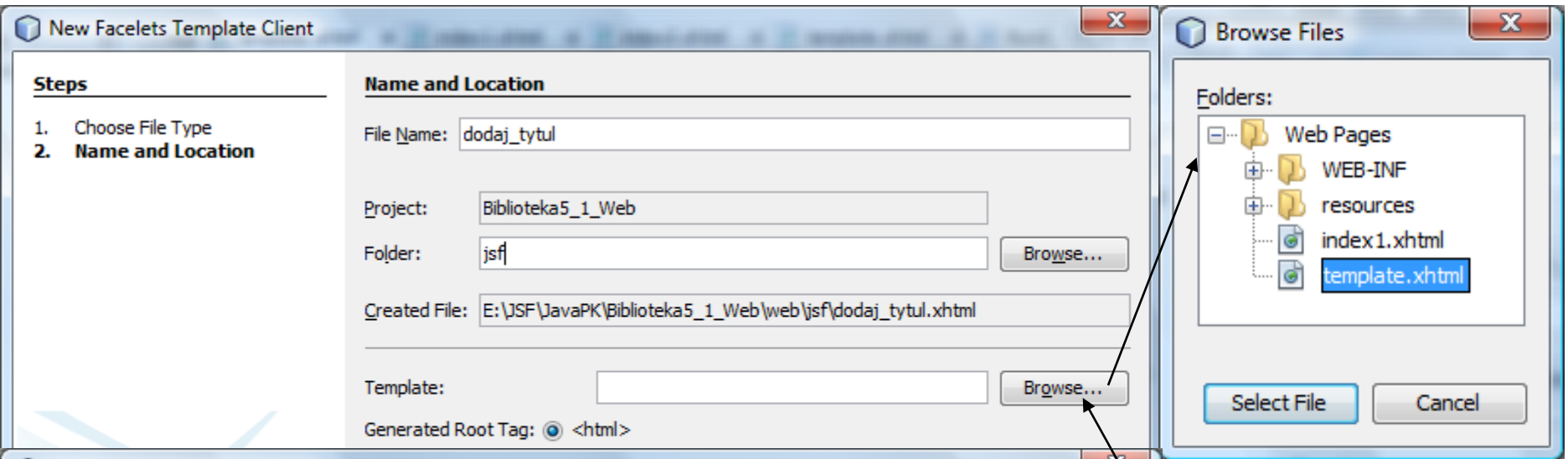
Wstawienie strony JSF typu **Facelets Template Client**, zawierającej formularz do wprowadzania danych tytułu książki - prawym klawiszem kliknąć na nazwę projektu i wybór **New/Other**



Wybór typu strony – **JavaServer Faces/Facelets Template Client**

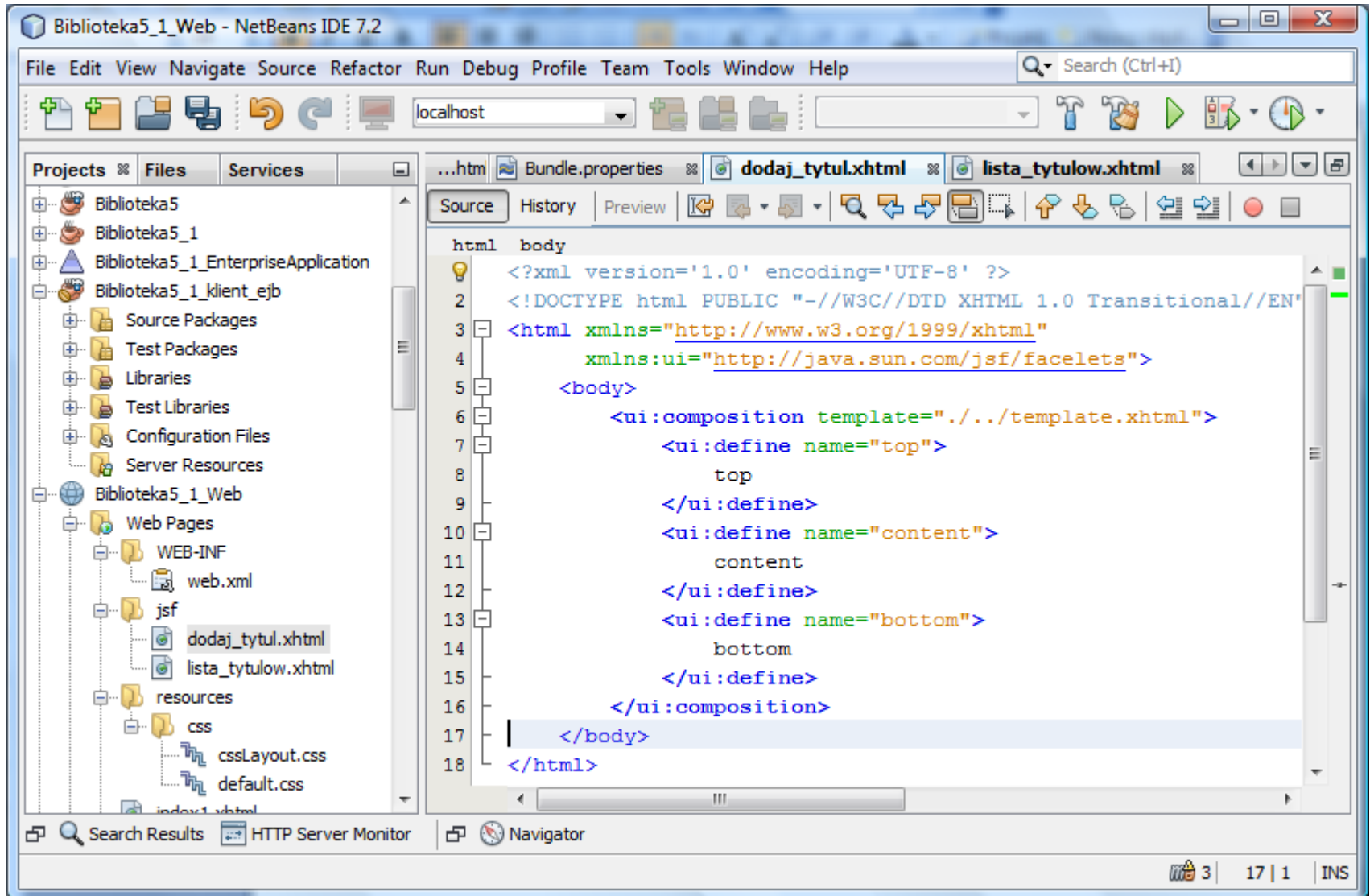


Nadanie nazwy **dodaj_tytul** stronie JSF w polu **File Name**. W polu **Folder** w pisać nazwę **jsf** katalogu stron



Wybór szablonu, wg którego powinna być zbudowana strona **index1**, w polu **Template** za pomocą klawisza **Browse**

Utworzona nowa strona JSF o nazwie **dodaj_tytul.xhtml** jako **Facelets Template Client**



Kod strony **dodaj_tytul.xhtml** - zmodyfikowana zawartość strony w zakresie znacznika **<ui:composition**

```
<?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">
      <h:outputText value="#{bundle['jsf.dodaj_tytul.tytul']}">
      </h:outputText>
    </ui:define>
    <ui:define name="content">
      <h:form><h:panelGrid columns="2">
```

Definicje
własne
strony
wybranych
części
szablonu
(**title**,
content)

```
<h:outputLabel value="#{bundle['jsf.dodaj_tytul.podaj_tytul']}" for="tytul" />
<h:inputText
    id="tytul"
    title="#{bundle['jsf.dodaj_tytul.podaj_tytul']}"
    value="#{managed_tytul.tytul}"
    required="true"
    requiredMessage="#{bundle['jsf.dodaj_tytul.podaj_tytul_blad']}" >
</h:inputText>
<h:outputLabel value="#{bundle['jsf.dodaj_tytul.podaj_nazwisko']}" for="nazwisko" />
<h:inputText
    id="cena"
    title="#{bundle['jsf.dodaj_tytul.podaj_nazwisko']}"
    value="#{managed_tytul.nazwisko}"
    required="true"
    requiredMessage="#{bundle['jsf.dodaj_tytul.podaj_nazwisko_blad']}" >
</h:inputText>
<h:outputLabel value="#{bundle['jsf.dodaj_tytul.podaj_imie']}" for="imie" />
<h:inputText
    id="imie"
    title="#{bundle['jsf.dodaj_tytul.podaj_imie']}"
    value="#{managed_tytul.imie}"
    required="true"
    requiredMessage="#{bundle['jsf.dodaj_tytul.podaj_imie_blad']}" >
</h:inputText>
```

```

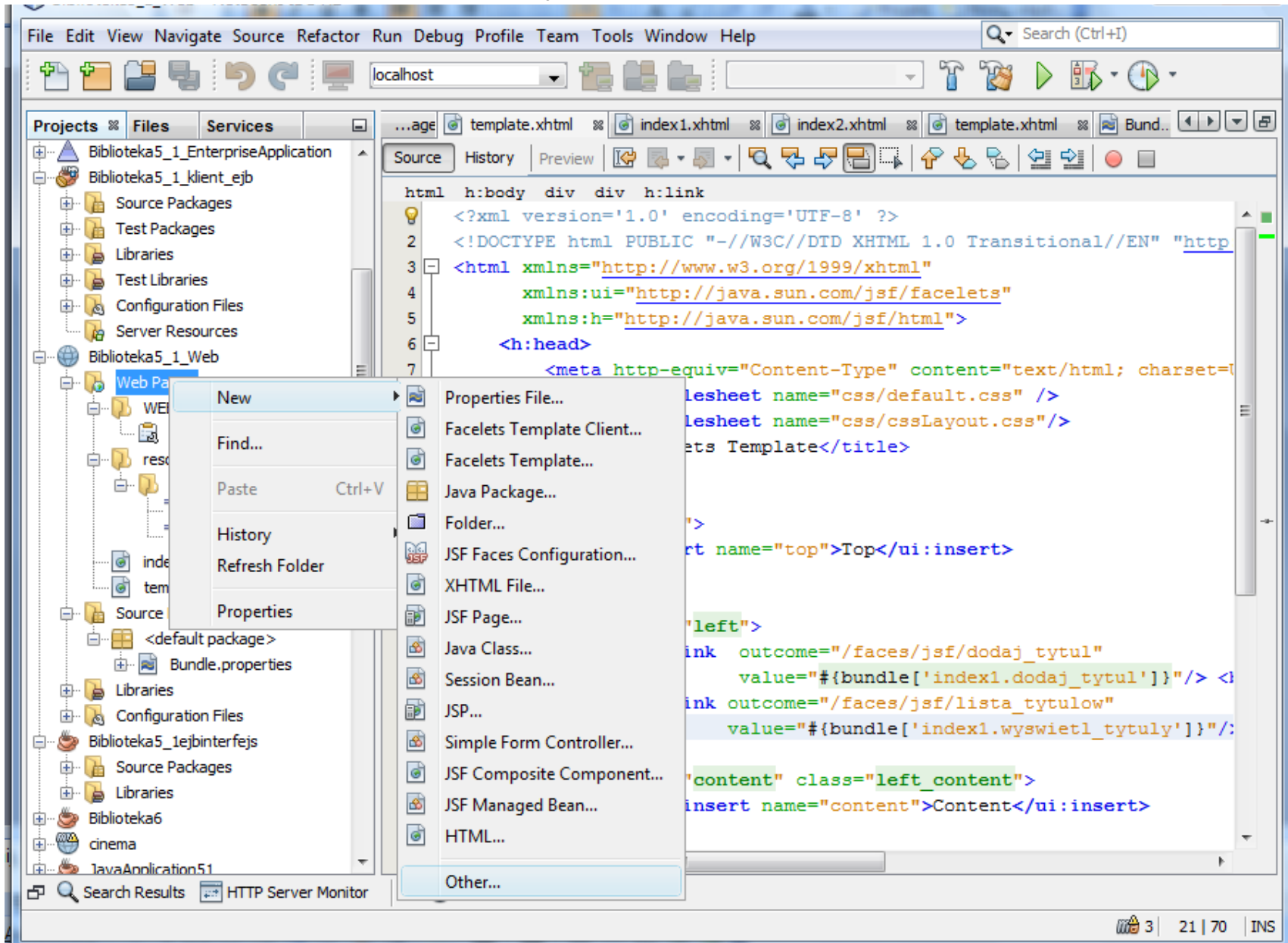
<h:outputLabel value="#{bundle['jsf.dodaj_tytul.podaj_ISBN']}" for="isbn" />
<h:inputText
    id="isbn"
    title="#{bundle['jsf.dodaj_tytul.podaj_ISBN']}"
    value="#{managed_tytul.isbn}"
    required="true"
    requiredMessage="#{bundle['jsf.dodaj_tytul.podaj_ISBN_blad']}" >
</h:inputText>
<h:outputLabel value="#{bundle['jsf.dodaj_tytul.podaj_Wydawnictwo']}" for="Wydawnictwo" />
<h:inputText
    id="ISBN"
    title="#{bundle['jsf.dodaj_tytul.podaj_Wydawnictwo']}"
    value="#{managed_tytul.wydawnictwo}"
    required="true"
    requiredMessage="#{bundle['jsf.dodaj_tytul.podaj_Wydawnictwo_blad']}" >
</h:inputText>
</h:panelGrid>
<h:commandLink action="#{managed_tytul.dodaj_tytul}"
    value="#{bundle['jsf.dodaj_tytul.akcja']}" />

</h:form>
</ui:define>
</ui:composition>
</body>
</html>

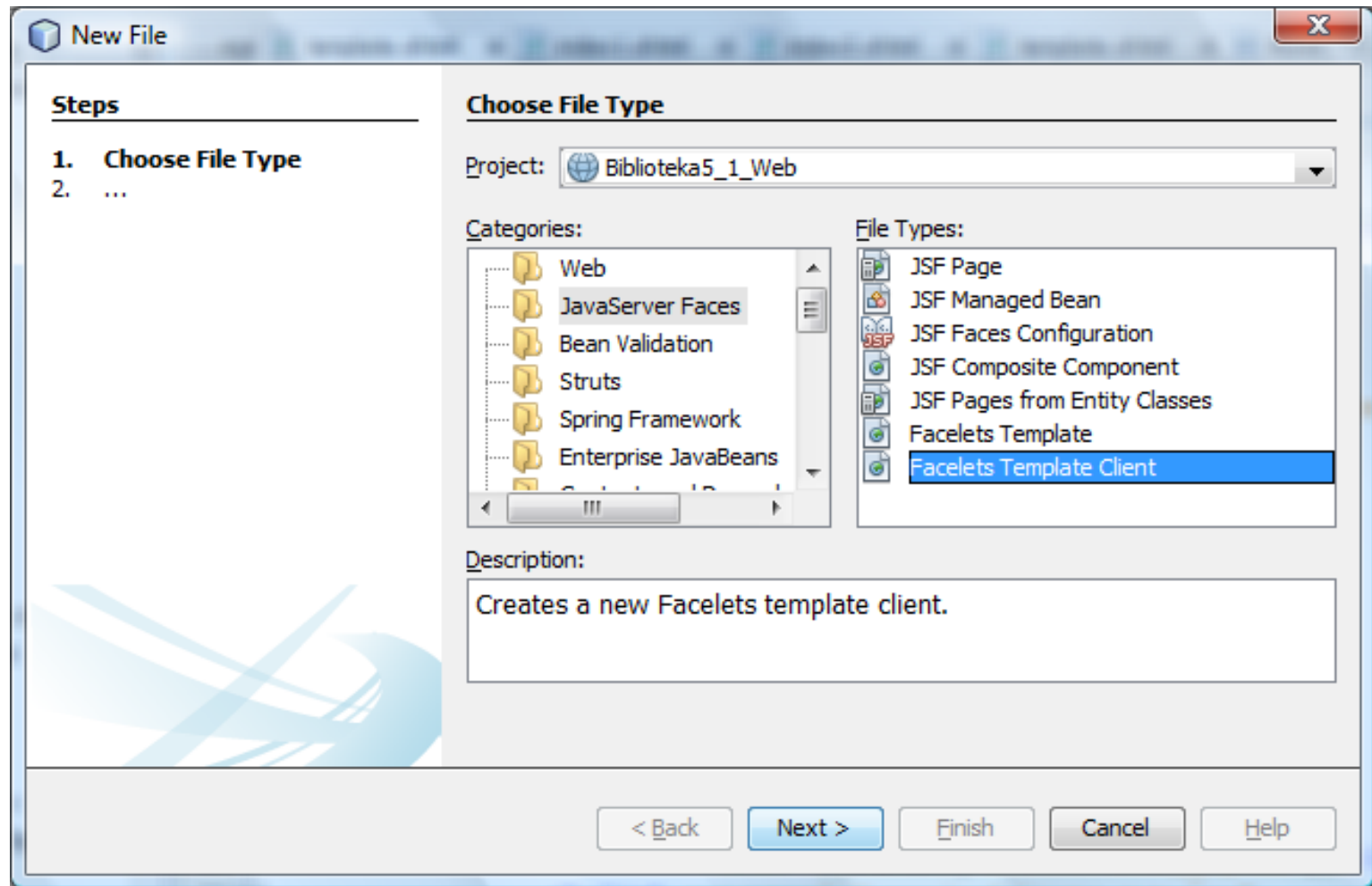
```

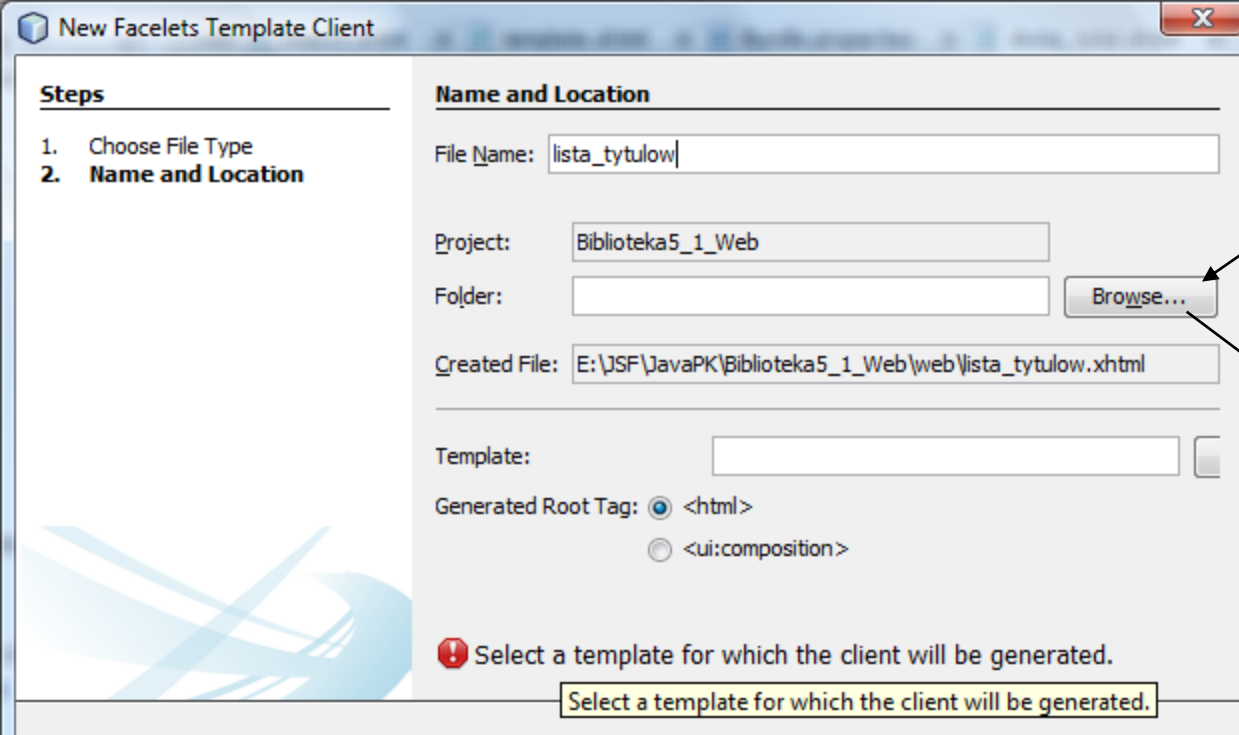
Pozostałe części
strony (**top, left,**
bottom) oparte są na
definicjach zawatych
w szablonie

Wstawienie strony JSF typu **Facelets Template Client**, zawierającej formularz do wyświetlania tabeli z danymi tytułu książki - prawym klawiszem kliknąć na nazwę projektu i wybór **New/Other**

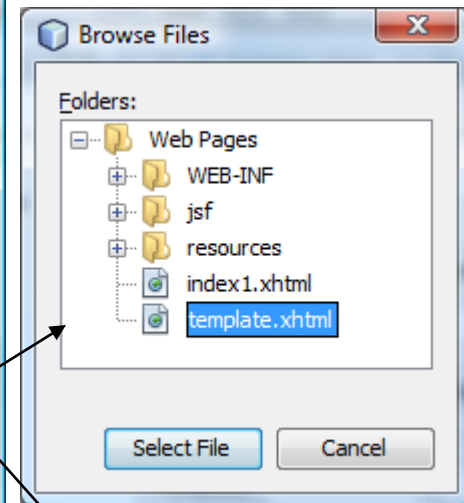
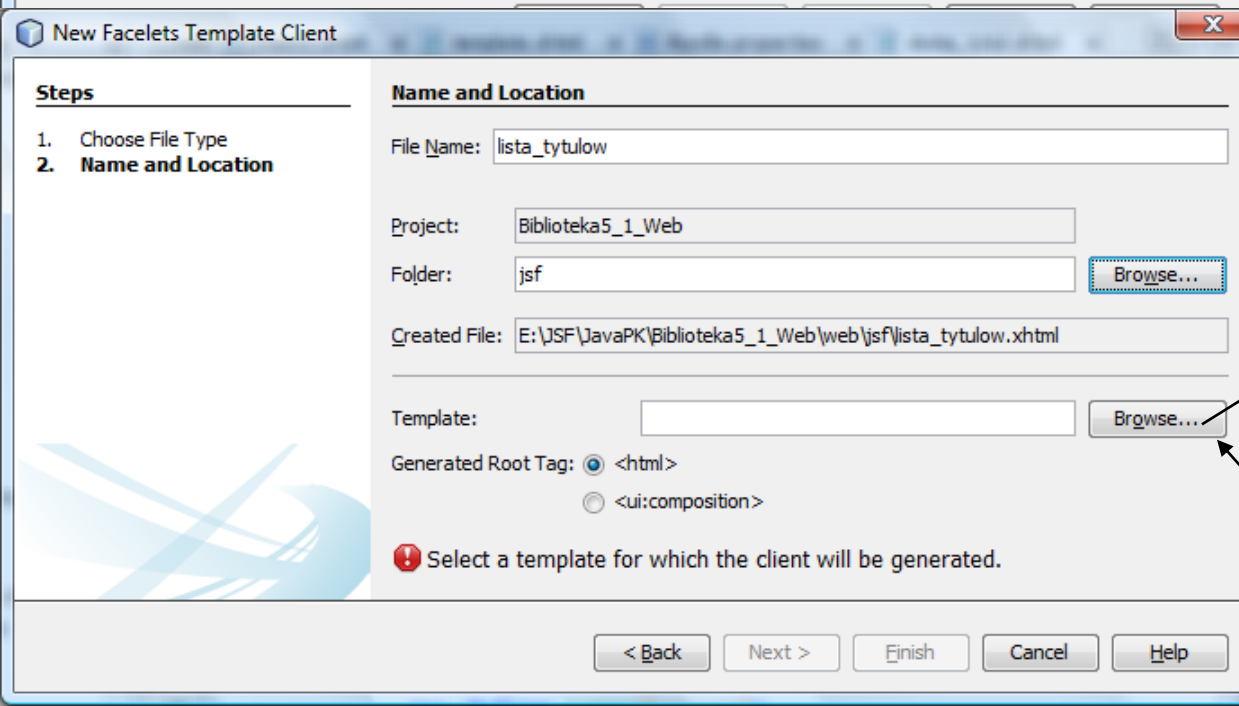
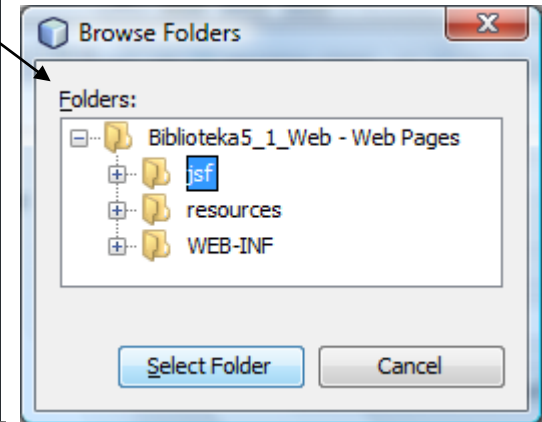


Wybór typu strony – **JavaServer Faces/Facelets Template Client**





Wybór foldera jsf



Wybór szablonu strony

Zakończona definicja nowej strony JSF **lista_tytulow** – nacisnąć klawisz **Finish**

New Facelets Template Client

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: lista_tytulow

Project: Biblioteka5_1_Web

Folder: jsf **Browse...**

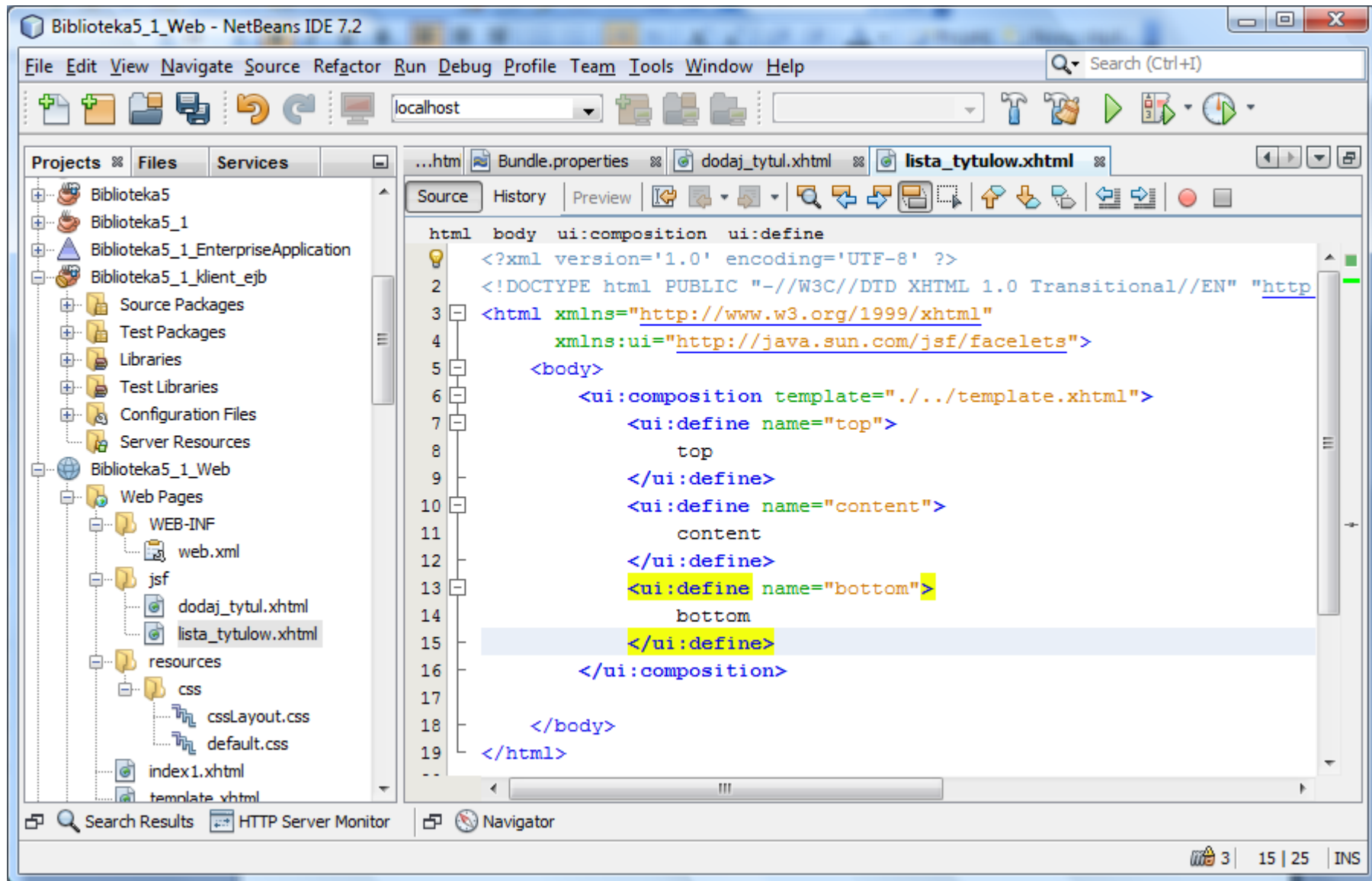
Created File: E:\JSF\JavaPK\Biblioteka5_1_Web\web\jsf\lista_tytulow.xhtml

Template: JSF\JavaPK\Biblioteka5_1_Web\web\template.xhtml **Browse...**

Generated Root Tag: <html>
 <ui:composition>

< Back Next > Finish Cancel Help

Utworzona nowa strona JSF o nazwie **lista_tytulow.xhtml** jako **Facelets Template Client**



Kod strony lista_tytulow.xhtml - zmodyfikowana zawartość strony w zakresie znacznika **<ui:composition**

```
<?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"
  xmlns:f="http://java.sun.com/jsf/core">
```

```
<body>
```

```
<ui:composition template=" ../template.xhtml" >
```

```
<ui:define name="title" >
```

```
  <h:outputText value="#{bundle['jsf.lista_tytulow.tytul']}">
  </h:outputText>
```

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

```
<ui:define name="content" >
```

```
  <h:form styleClass="jsfcrud_list_form">
    <h:panelGroup id="messagePanel" layout="block">
      <h:messages errorStyle="color: red" infoStyle="color: green,,
        layout="table"/>
    </h:panelGroup>
```

```
  </h:panelGroup>
```

Definicje
własne
strony
wybranych
części
szablonu
(**title**,
content)

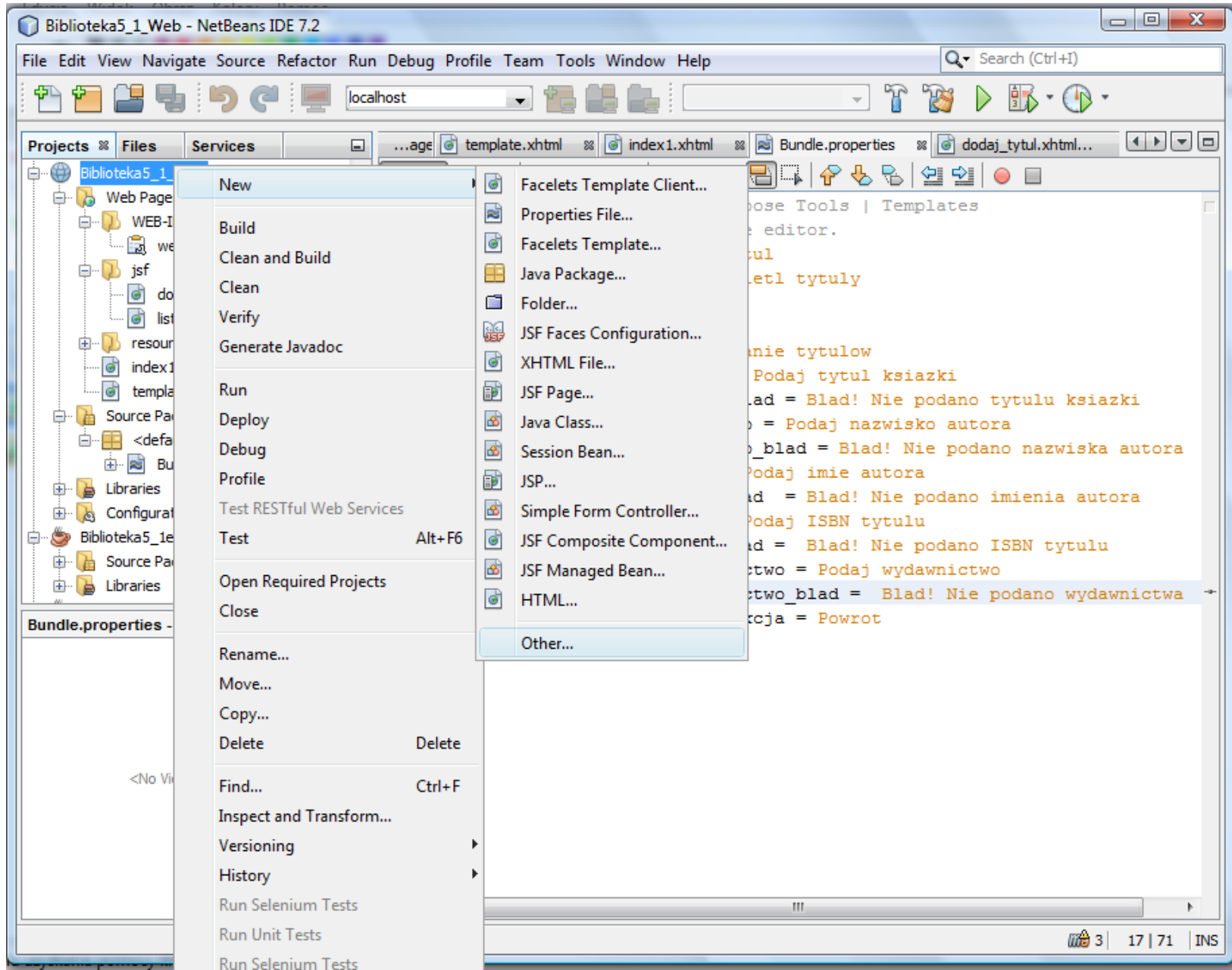
```
<h:dataTable value="#{managed_tytul.items}" var="item"
              border="0" cellpadding="2" cellspacing="0"
              rowClasses="jsfcrud_odd_row,jsfcrud_even_row"
              rules="all" style="border:solid 1px">
  <h:column>
    <f:facet name="header">
      <h:outputText value="#{bundle['jsf.lista_tytulow.tytul']}" />
    </f:facet>
    <h:outputText value="#{item[0]}" />
  </h:column>
  <h:column>
    <f:facet name="header">
      <h:outputText value="#{bundle['jsf.lista_tytulow.nazwisko']}" />
    </f:facet>
    <h:outputText value="#{item[1]}" />
  </h:column>
  <h:column>
    <f:facet name="header">
      <h:outputText value="#{bundle['jsf.lista_tytulow.imie']}" />
    </f:facet>
    <h:outputText value="#{item[2]}" />
  </h:column>
```

```
<h:column>
  <f:facet name="header">
    <h:outputText value="#{bundle['jsf.lista_tytulow.isbn']}" />
  </f:facet>
  <h:outputText value="#{item[3]}" />
</h:column>
<h:column>
  <f:facet name="header">
    <h:outputText value="#{bundle['jsf.lista_tytulow.wydawnictwo']}" />
  </f:facet>
  <h:outputText value="#{item[4]}" />
</h:column>
</h:dataTable>
</h:panelGroup>
</h:form>
</ui:define>
</ui:composition>
</body>
</html>
```

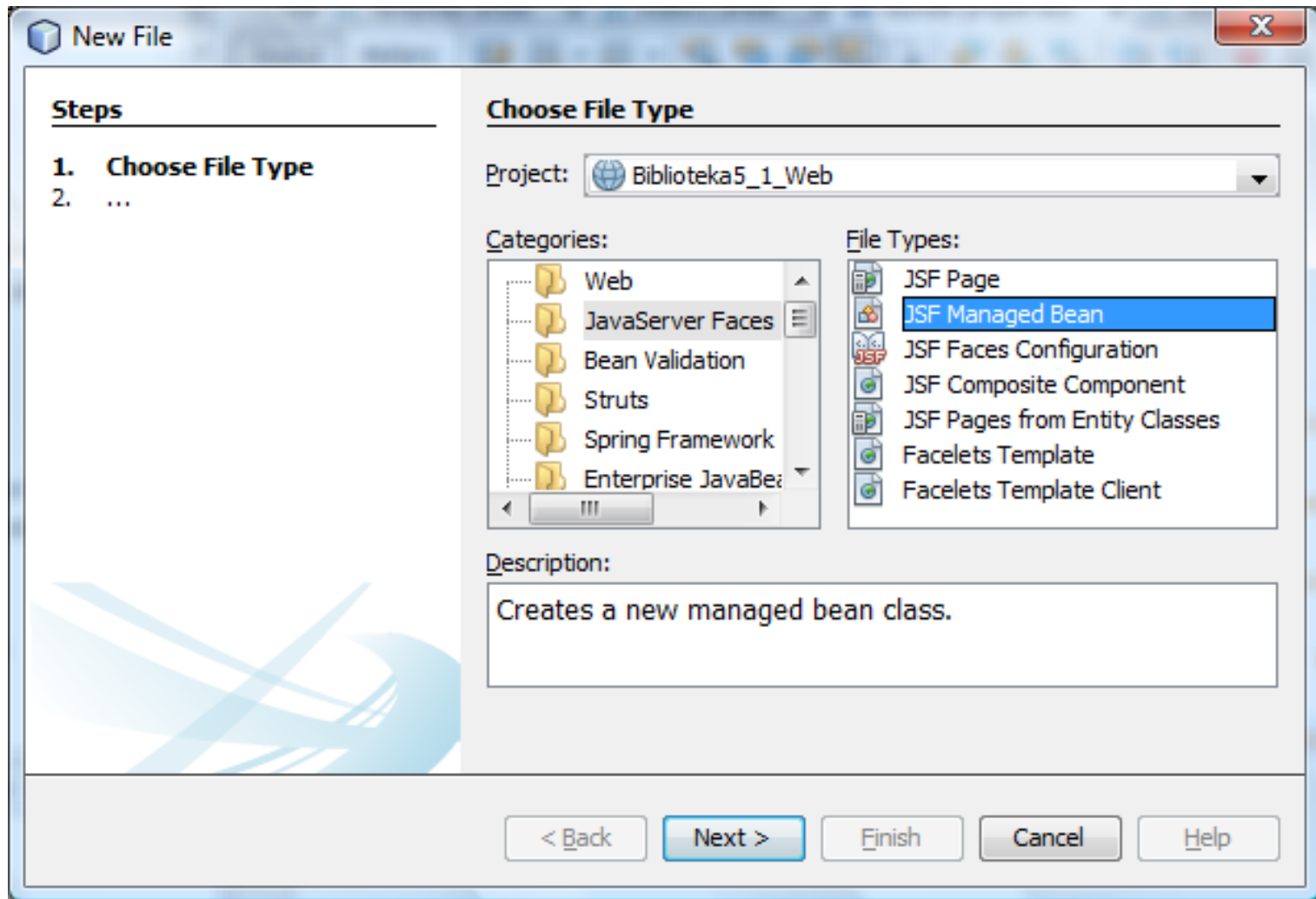
Pozostałe części
strony (**top**, **left**,
bottom) oparte są na
definicjach zawatych
w szablonie

Dodanie klasy typu **Managed Bean** do obsługi programowej utworzonych stron – pełniącej rolę kontrolera w modelu MVC

Dodanie klasy typu **Managed Bean** - prawym klawiszem kliknąć na nazwę projektu i wybór **New/Other**



Wybór **JavaServer Faces / JSF Managed Bean**



Nadanie klasie nazwy **Managed_tytul** w polu **Class Name** oraz wstawienie jej do nowego pakietu **jsf** typu **Java Package**. Nazwa obiektu **managed_tytul** jest nadana w polu **Name**. Czas życia obiektu jest równy **request** w polu **Scope** - obiekt jest tworzony podczas obsługi zadania (**request**) i usuwany w fazie odpowiedzi (**response**). Można wybrać dłuższy czas życia obiektu – **session, application, view** lub **none**.

New JSF Managed Bean

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

Add data to configuration file

Configuration File:

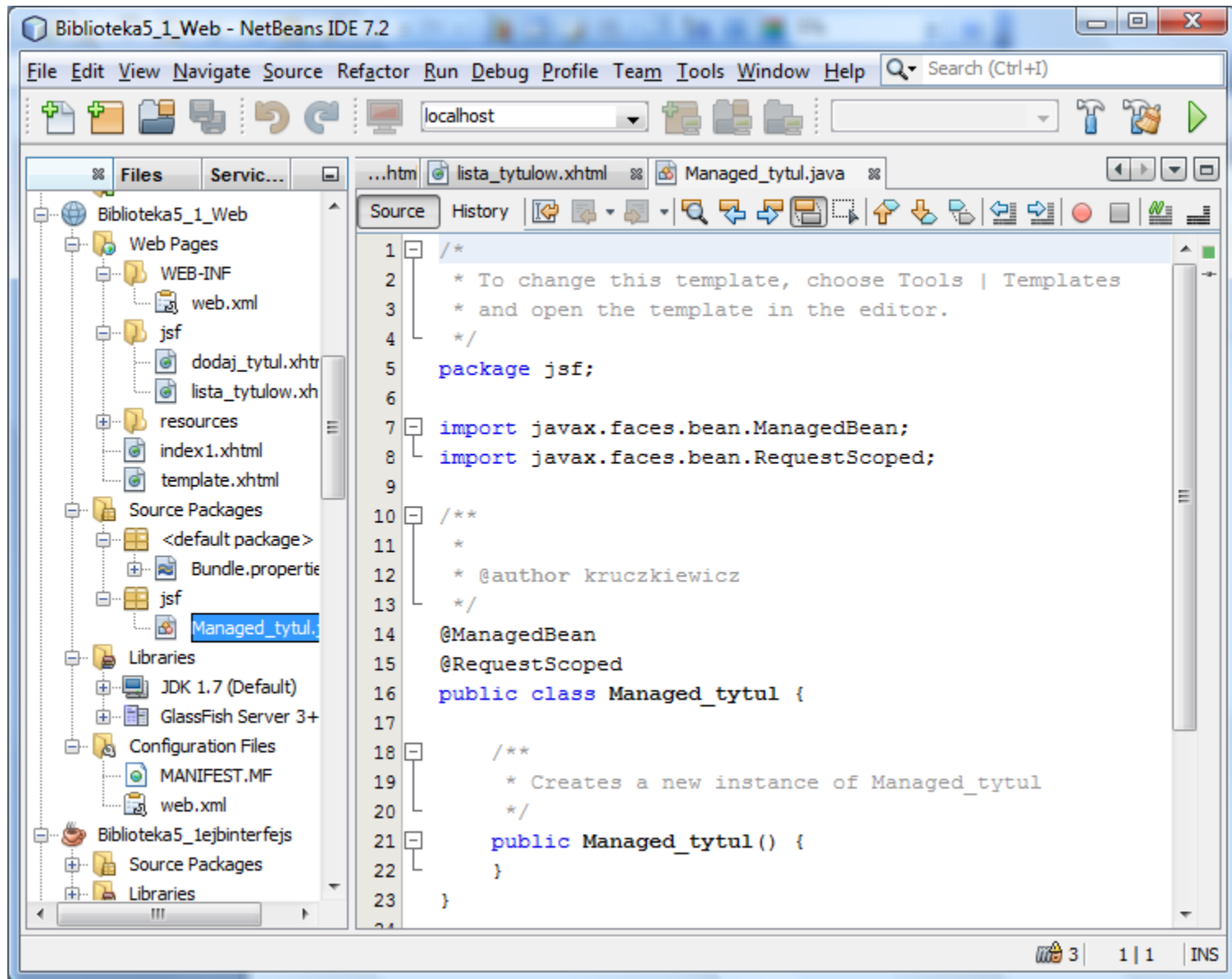
Name:

Scope:

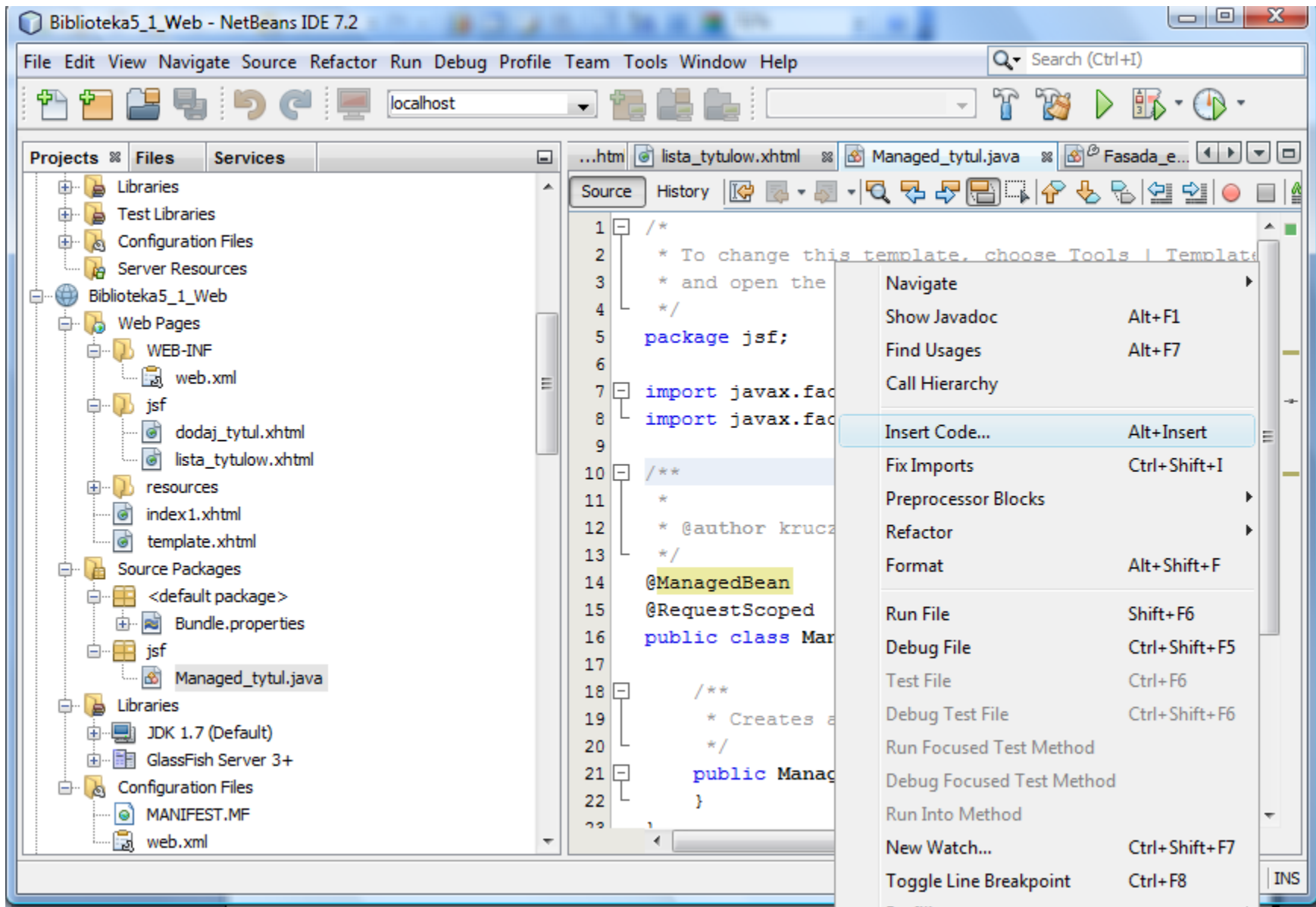
Bean Description:

< Back Next > **Finish** Cancel Help

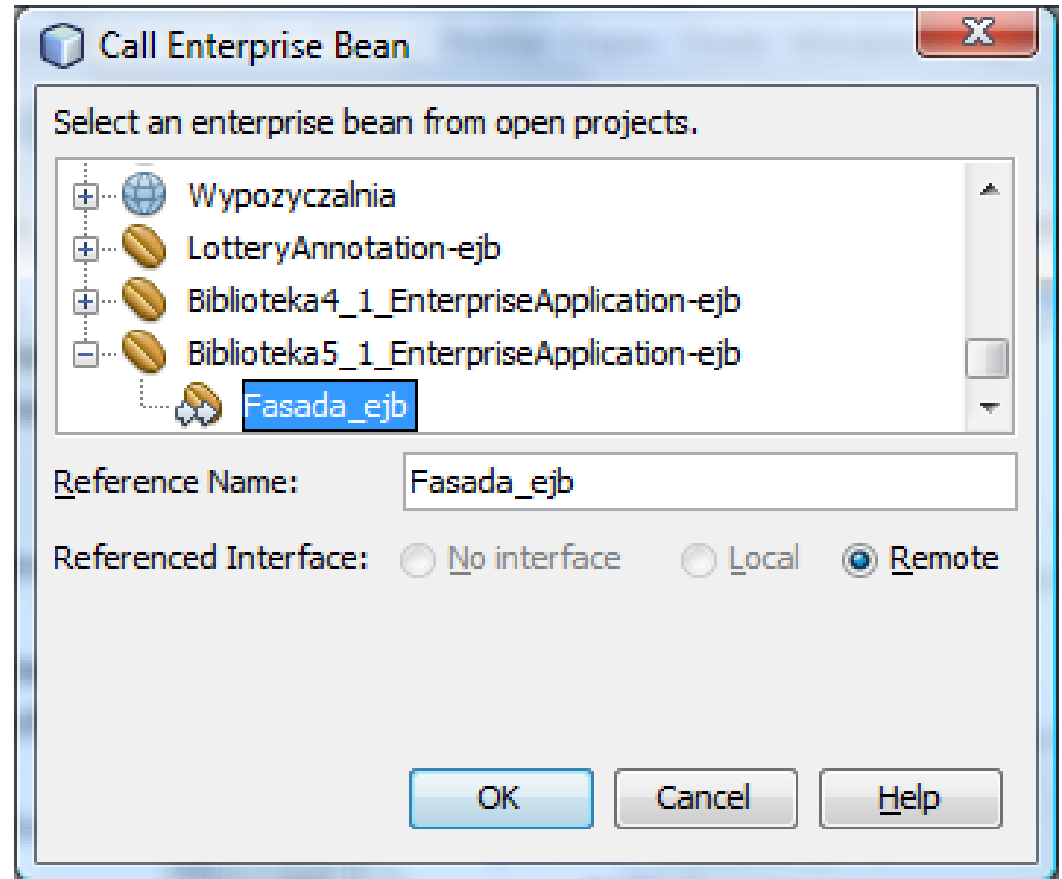
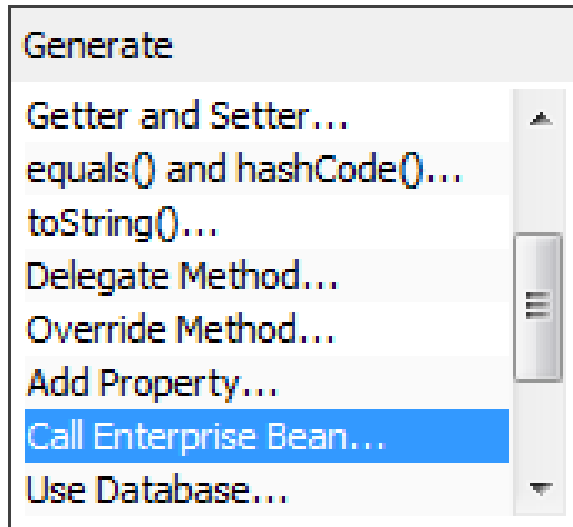
Utworzona pusta klasa **Managed_tytul**



Dodanie referencji do zdalnego obiektu typu EJB, czyli **Fasada_ejb** – po kliknięciu prawym klawiszem myszy na powierzchnię edytora należy wybrać pozycję **Insert Code...**



Wybór z listy pozycję **Call Enterprise Bean...** i następnie wybór projektu **Biblioteka5_1-EnterpriseApplication** z listy projektów, i następnie po naciśnięciu na pole+ przy projekcie wybór ziarna **Fasada_ejb**.




Kod klasy **Managed_tytul** – zaznaczono utworzoną referencję do obiektu zdalnego typu EJB

```
package jsf;  
  
import.ejb.Fasada_ejbRemote;  
import.java.util.ArrayList;  
import.java.xml.ejb.EJB;  
import.java.xml.faces.bean.ManagedBean;  
import.java.xml.faces.bean.RequestScoped;  
import.java.xml.faces.model.DataModel;  
import.java.xml.faces.model.ListDataModel;
```

```
@ManagedBean  
@RequestScoped  
public class Managed_tytul {
```

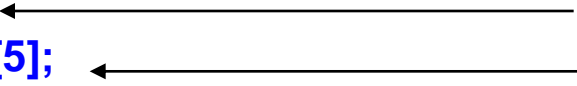
```
@EJB  
private Fasada_ejbRemote fasada;
```

Dostęp do logiki
biznesowej za
pomocą ziarna typu
EJB



```
private DataModel items;  
String[] tytuly = new String[5];
```

Modele widoków
oparte na danych
tworzonych przez
przez warstwę
biznesową



```
public Managed_tytul() {  
}
```

```
public Fasada_ejbRemote getFasada_ejb() {  
    return fasada;  
}
```

```
public void setFasada_ejb(Fasada_ejbRemote fasada_ejb) {  
    this.fasada = fasada_ejb;  
}
```

```
public String[] getTytuly() {  
    return tytuly;  
}
```

```
public void setTytuly(String[] tytuly) {  
    this.tytuly = tytuly;  
}
```

```
public String getTytul() {
    return tytuly[0];
}
public void setTytul(String val) {
    this.tytuly[0] = val;
}
public String getImie() {
    return tytuly[1];
}
public void setImie(String val) {
    this.tytuly[1] = val;
}
public String getNazwisko() {
    return tytuly[2];
}
public void setNazwisko(String val) {
    this.tytuly[2] = val;
}
public String getIlsbn() {
    return tytuly[3];
}
public void setIlsbn(String val) {
    this.tytuly[3] = val;
}
public String getWydawnictwo() {
    return tytuly[4];
}
public void setWydawnictwo(String val) {
    this.tytuly[4] = val;
}
```

Właściwości modelu
wykorzystane do atrybutów
value znaczników typu
<h:inputText
renderowanych jako pola
wejściowe
(**<input type="text">**)

```
public String dodaj_tytul() {
    fasada.dodaj_tytul(tytuly);
    return "/faces/index1";
}
```

Obsługa zdarzenia kliknięcia obsługiwane
przez komponent znacznika
<h:commandLink (atrybut **action**)

```
public DataModel utworz_DataModel() {
    return new ListDataModel(tytuly(fasada.tytuly()));
}
```

```
public ArrayList<String[]> tytuly(ArrayList<String> lista)
{
    ArrayList<String[]> pom =new ArrayList();
    String [] dane;
    for (String s: lista)
    {
        dane=s.split(" ");
        String[] dane1={dane[1], dane[3], dane[4], dane[6], dane[8]};
        pom.add(dane1); }
    return pom;
}
```

Metody pomocnicze

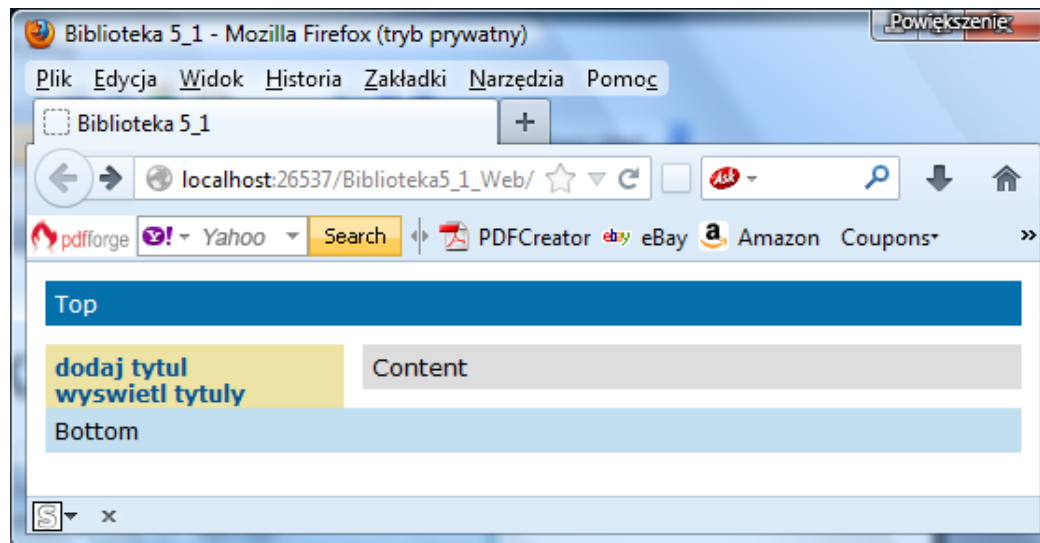
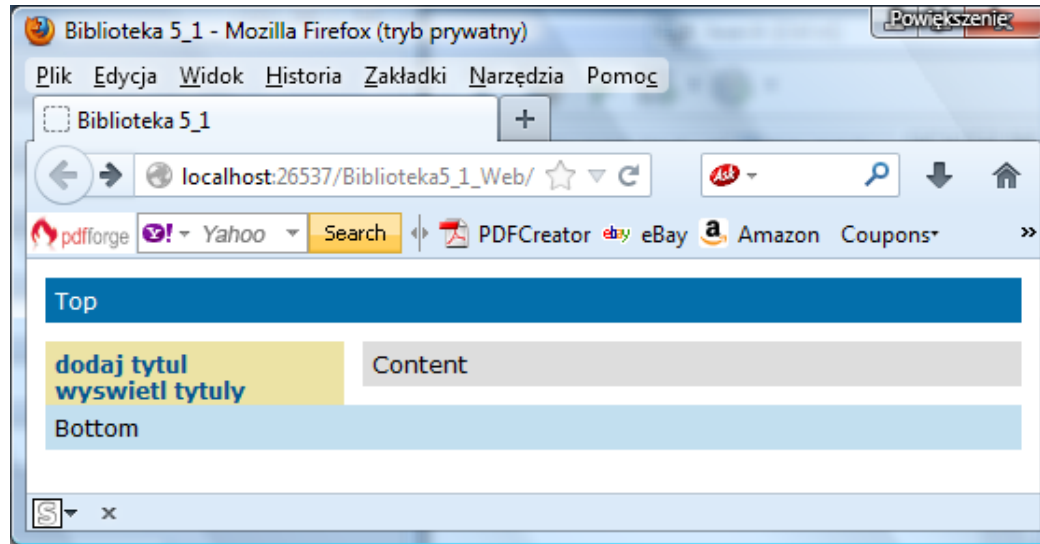
```
public DataModel getItems() {
    if (items == null) {
        items = utworz_DataModel(); }
    return items;
}
```

```
public void setItems(DataModel items) {
    this.items = items;
}
}
```

Właściwości modelu
wykorzystane do
atrybutów **value** i **var**
komponentu typu
<h:dataTable
renderowanych jako
tabela (**<table>**)

Uruchomienie aplikacji z warstwą klienta zawierającej dwa typy aplikacji klientów:
klienta przeglądarki oraz klienta EE

Uruchomienie dwóch klientów przeglądarki – widok strony **index1.xhtml** po renderingu. W celu uruchomienia należy wpisać adres strony:
http://localhost:8080/Biblioteka5_1_Web/



Widok formularza klienta typu Java EE 6 – widok danych pobranych za pośrednictwem ziarna **Fasada_ejb** (w warstwie biznesowej pobrano dane z bazy danych i zbuforowano w aplikacji)

Tytuł książki: 3

Nazwisko autora książki: 3

Imię autora książki: 3

ISBN tytułu książki: 3

Wydawnictwo książki: 3

Numer książki:

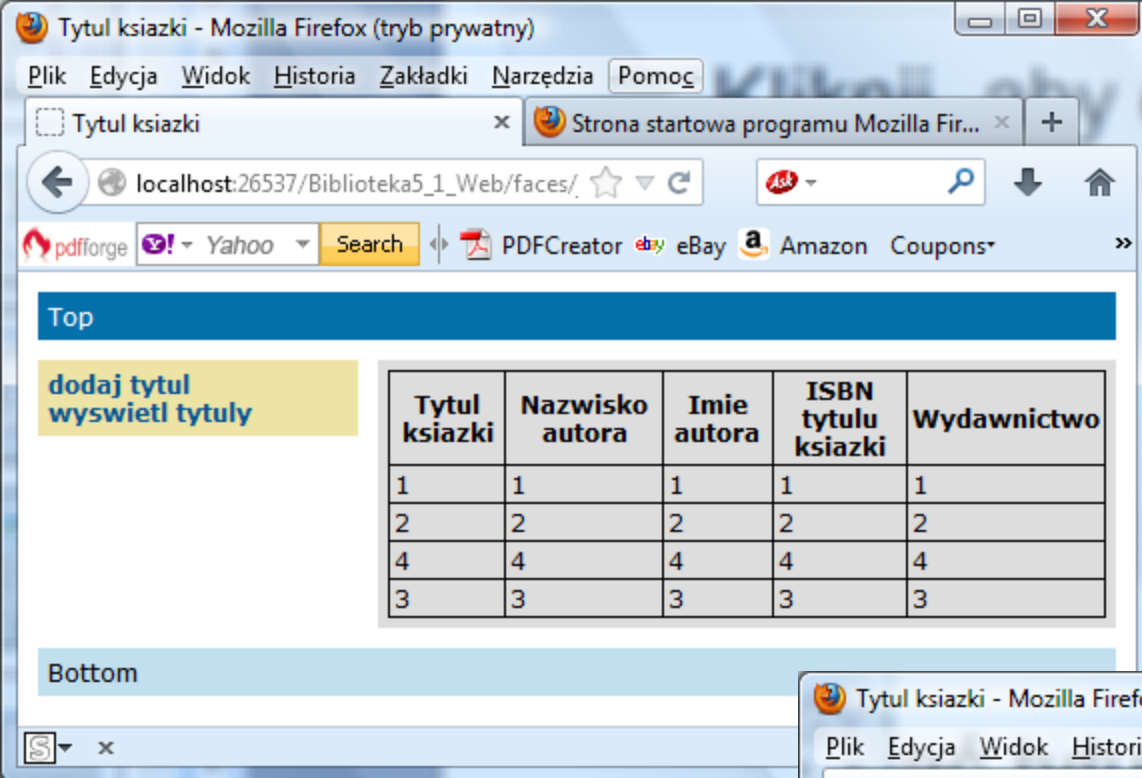
Zapisz tytuł Zapisz książkę Wyświetl tytuły Wyświetl książki

Tytuły do bazy Książki do bazy Tytuły z bazy Książki z bazy Dane z bazy

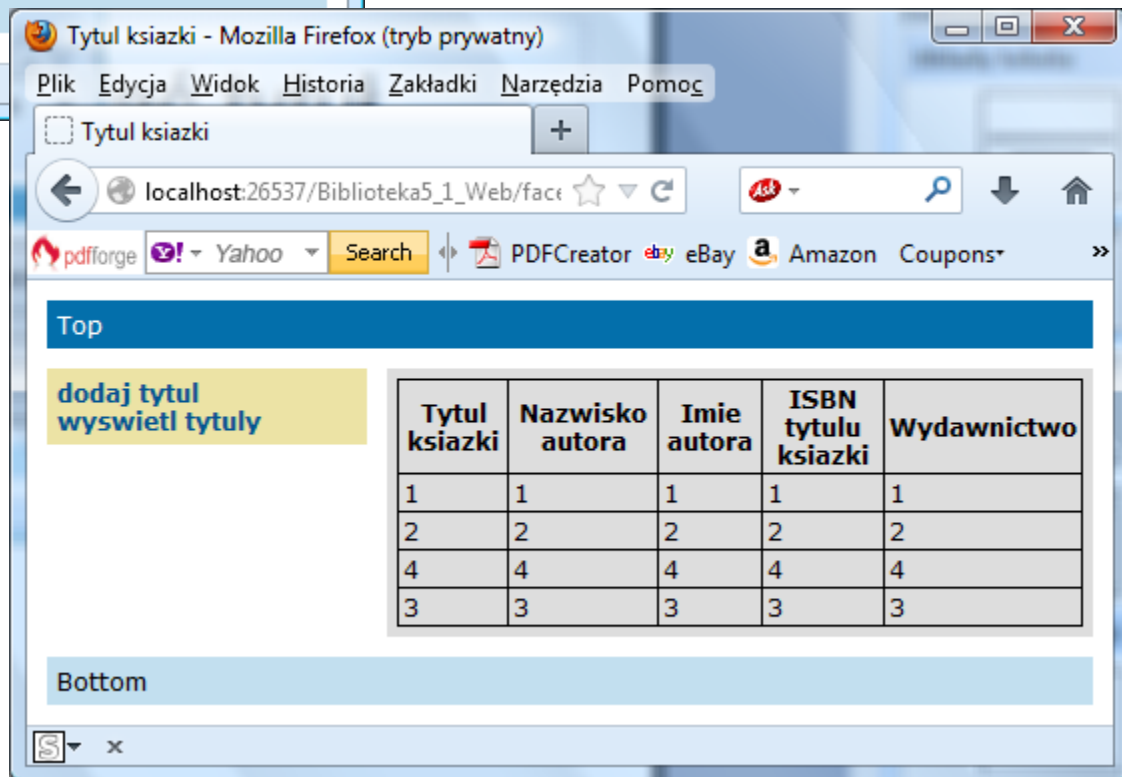
Tytuły książek: Tytuł: 1 Autor: 1 1 ISBN: 1 Wydawnictwo: 1

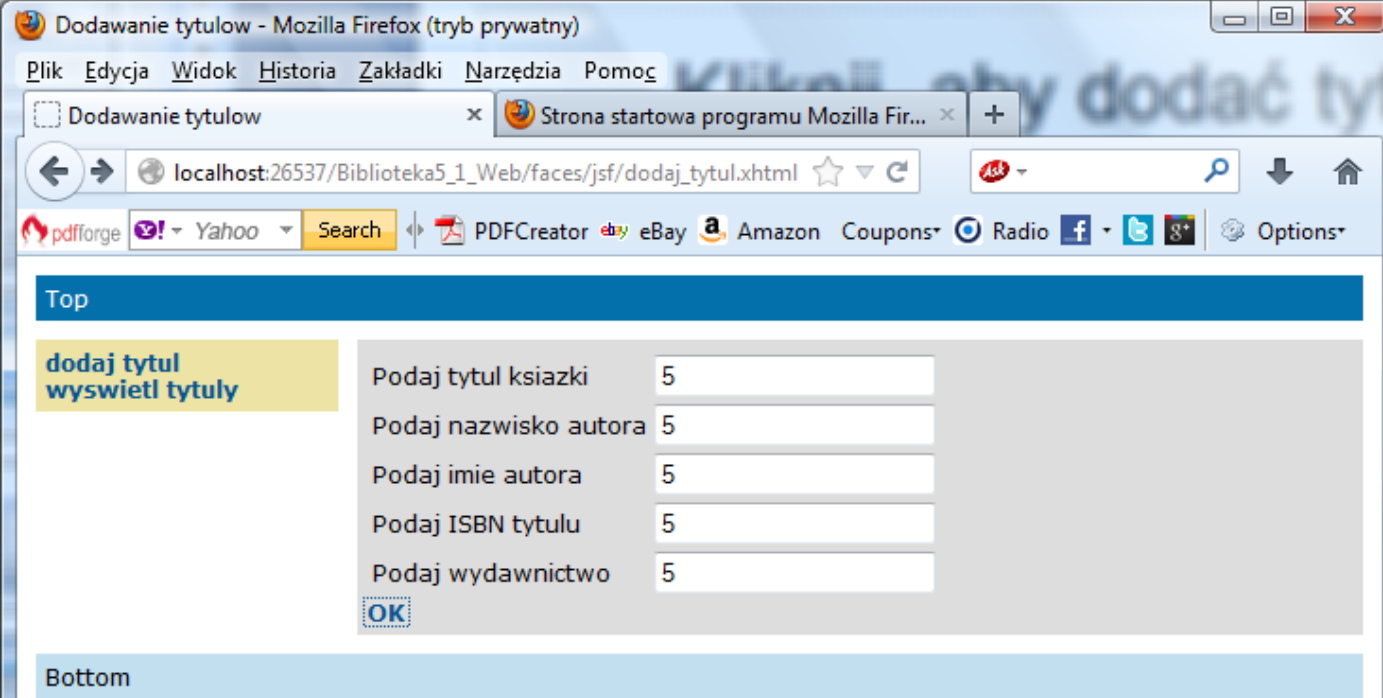
Książki:

- Tytuł: 1 Autor: 1 1 ISBN: 1 Wydawnictwo: 1
- Tytuł: 2 Autor: 2 2 ISBN: 2 Wydawnictwo: 2
- Tytuł: 4 Autor: 4 4 ISBN: 4 Wydawnictwo: 4
- Tytuł: 3 Autor: 3 3 ISBN: 3 Wydawnictwo: 3

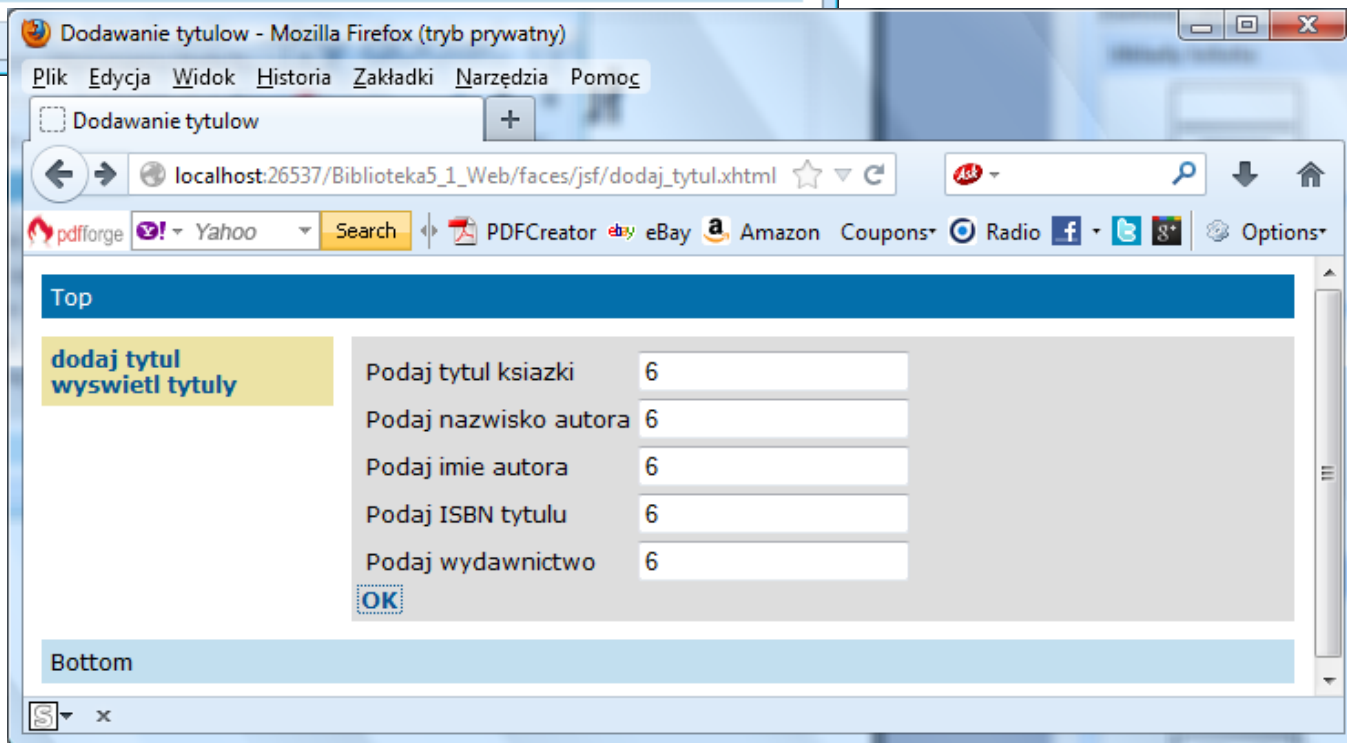


Widok danych pobranych za pomocą współużywanego ziarna **Fasada_ejb** z klientem Java EE 6.

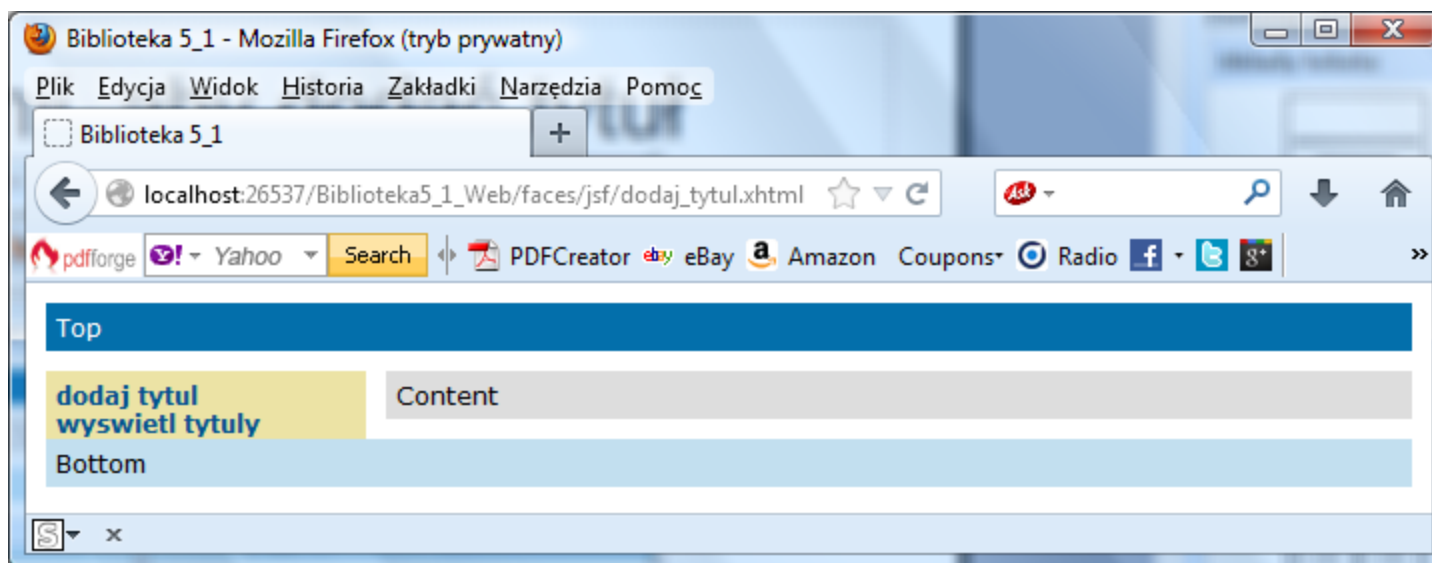
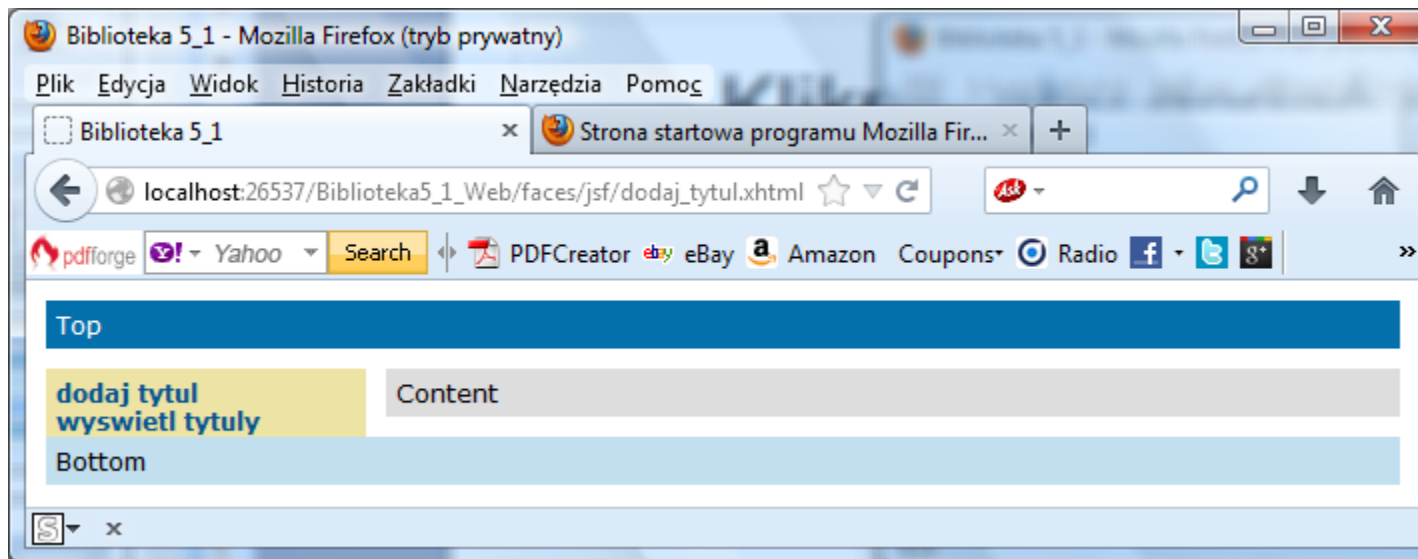




Wprowadzenie dwóch różnych danych tytułów do warstwy biznesowej za pomocą stron `dodaj_tytul.xhtml` (wywołanie metody `dodaj_tytul()` ziarna `Managed_tytul` po kliknięciu na **OK**, co powoduje wywołanie metody `fasada.dodaj_tytul(tytyly)`; ziarna `EJB (Fasada_ejb)`



Po kliknięciu na **OK** przekierowanie na stronę główną **index1.xhtml**



Tytuł książki - Mozilla Firefox (tryb prywatny)

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Tytuł książki x Strona startowa programu Mozilla Fir... x +

localhost:26537/Biblioteka5_1_Web/faces/jsf/lista_tytulow.xhtml

pdfforge Yahoo Search PDFCreator eBay Amazon Coupons Radio f t g+

Top

dodaj tytuł
wyświetl tytuły

Tytuł książki	Nazwisko autora	Imie autora	ISBN tytułu książki	Wydawnictwo
1	1	1	1	1
2	2	2	2	2
4	4	4	4	4
3	3	3	3	3
5	5	5	5	5
6	6	6	6	6

Bottom

Wyświetlenie na stronach **lista_tytulow.xhtml** danych pobranych od warstwy biznesowej za pomocą metody **fasada.tytuly()** – dane są przechowywane w warstwie biznesowej

Tytuł książki - Mozilla Firefox (tryb prywatny)

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Tytuł książki x +

localhost:26537/Biblioteka5_1_Web/faces/jsf/lista_tytulow.xhtml

pdfforge Yahoo Search PDFCreator eBay Amazon Coupons Radio f t g+

Top

dodaj tytuł
wyświetl tytuły

Tytuł książki	Nazwisko autora	Imie autora	ISBN tytułu książki	Wydawnictwo
1	1	1	1	1
2	2	2	2	2
4	4	4	4	4
3	3	3	3	3
5	5	5	5	5
6	6	6	6	6

Bottom

Widok formularza klienta typu Java EE 6 po wprowadzenie przez **klientów przeglądarki** – widok danych pobranych za pośrednictwem ziarna Fasada_ejb (w warstwie biznesowej pobrano dane z bazy danych i zbuforowano w aplikacji)

The screenshot shows a Java Swing window with a light blue title bar and standard OS window controls (minimize, maximize, close). The window contains a form for entering book details and a list of retrieved data.

Form fields:

- Tytuł książki: 3
- Nazwisko autora książki: 3
- Imię autora książki: 3
- ISBN tytułu książki: 3
- Wydawnictwo książki: 3
- Numer książki: (empty)

Buttons:

- Zapisz tytuł
- Zapisz książkę
- Wyświetl tytuły
- Wyświetl książki
- Tytuły do bazy
- Książki do bazy
- Tytuły z bazy
- Książki z bazy
- Dane z bazy

Data List:

Tytuły książek: [Dropdown menu showing selected item: Tytuł: 1 Autor: 1 1 ISBN: 1 Wydawnictwo: 1]

Książki:

- Tytuł: 1 Autor: 1 1 ISBN: 1 Wydawnictwo: 1
- Tytuł: 2 Autor: 2 2 ISBN: 2 Wydawnictwo: 2
- Tytuł: 4 Autor: 4 4 ISBN: 4 Wydawnictwo: 4
- Tytuł: 3 Autor: 3 3 ISBN: 3 Wydawnictwo: 3
- Tytuł: 5 Autor: 5 5 ISBN: 5 Wydawnictwo: 5
- Tytuł: 6 Autor: 6 6 ISBN: 6 Wydawnictwo: 6

Widok formularza klienta typu Java EE 6 po wprowadzeniu przez **klientów przeglądarki** – widok danych pobranych z bazy danych za pośrednictwem ziarna Fasada_ejb

The screenshot shows a web application window with the following elements:

- Input Fields:**
 - Tytuł książki: 3
 - Nazwisko autora książki: 3
 - Imię autora książki: 3
 - ISBN tytułu książki: 3
 - Wydawnictwo książki: 3
 - Numer książki: (empty)
- Buttons:**
 - Zapisz tytuł
 - Zapisz książkę
 - Wyświetl tytuły
 - Wyświetl książki
 - Tytuły do bazy
 - Książki do bazy
 - Tytuły z bazy
 - Książki z bazy
 - Dane z bazy
- Data Display:**
 - Tytuły książek:** A dropdown menu showing a list of titles and their associated metadata (id_tytuł, tytuł, autor_nazwisko, autor_imie, ISBN, wydawnictwo).
 - Książki:** A list of books with the same metadata as the titles.