

Wstęp - Prosta aplikacja internetowa w technologii Java EE 5

**Programowanie
komponentowe 1**

Przykład 1- Wykonanie prostej aplikacji internetowej w technologii JavaEE w środowisku Netbeans 5.5 z wykorzystaniem narzędzia typu RAPID – Visual Web Pack

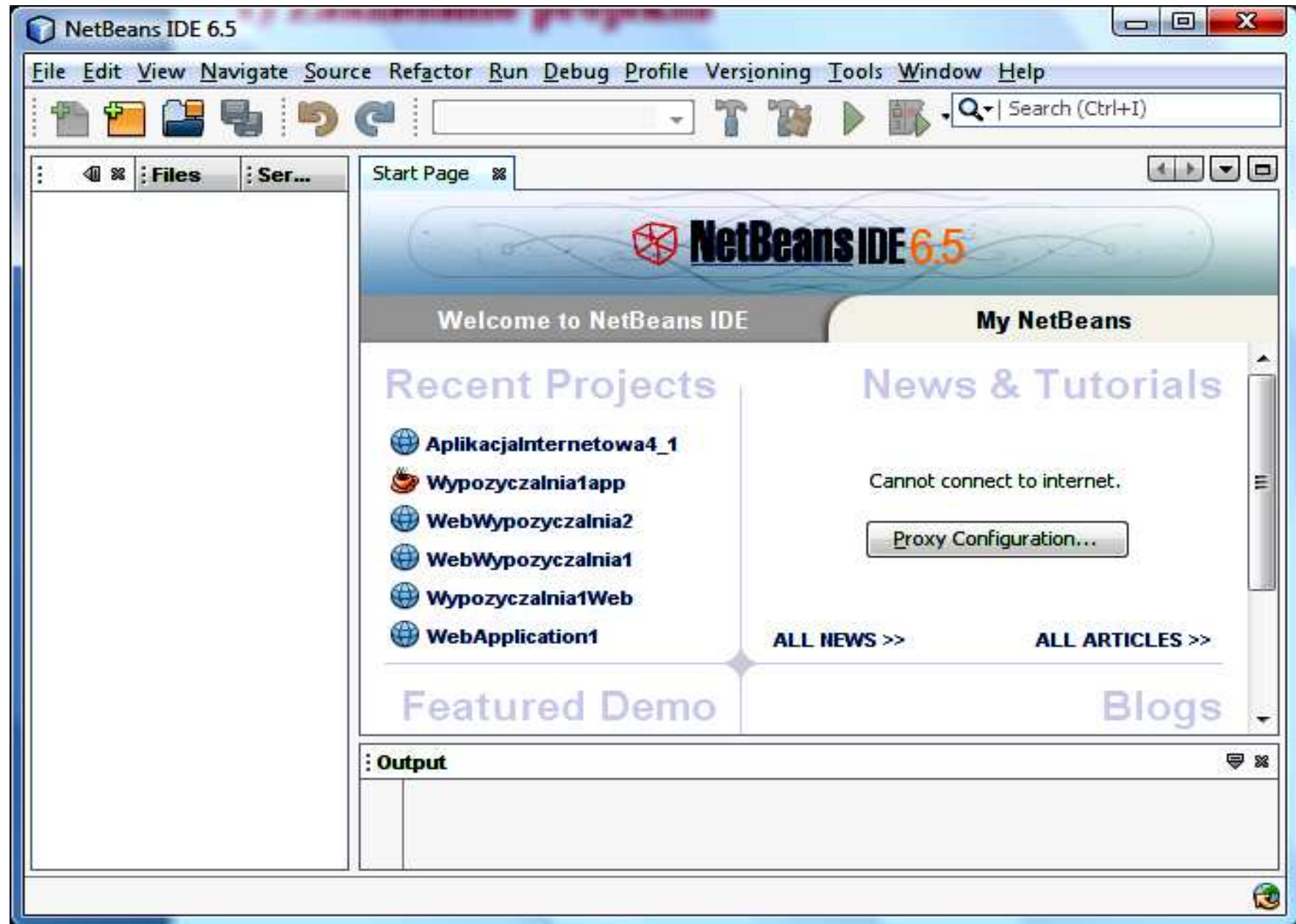
Czynności podstawowe – pierwsza iteracja

- 1) Zakładanie projektu**
- 2) Zaprojektowanie strony: komponenty Label, TextField, Button, StaticText, MessageGroup**
- 3) Dodanie nowych właściwości elementom strony**
- 4) Uruchomienie aplikacji**

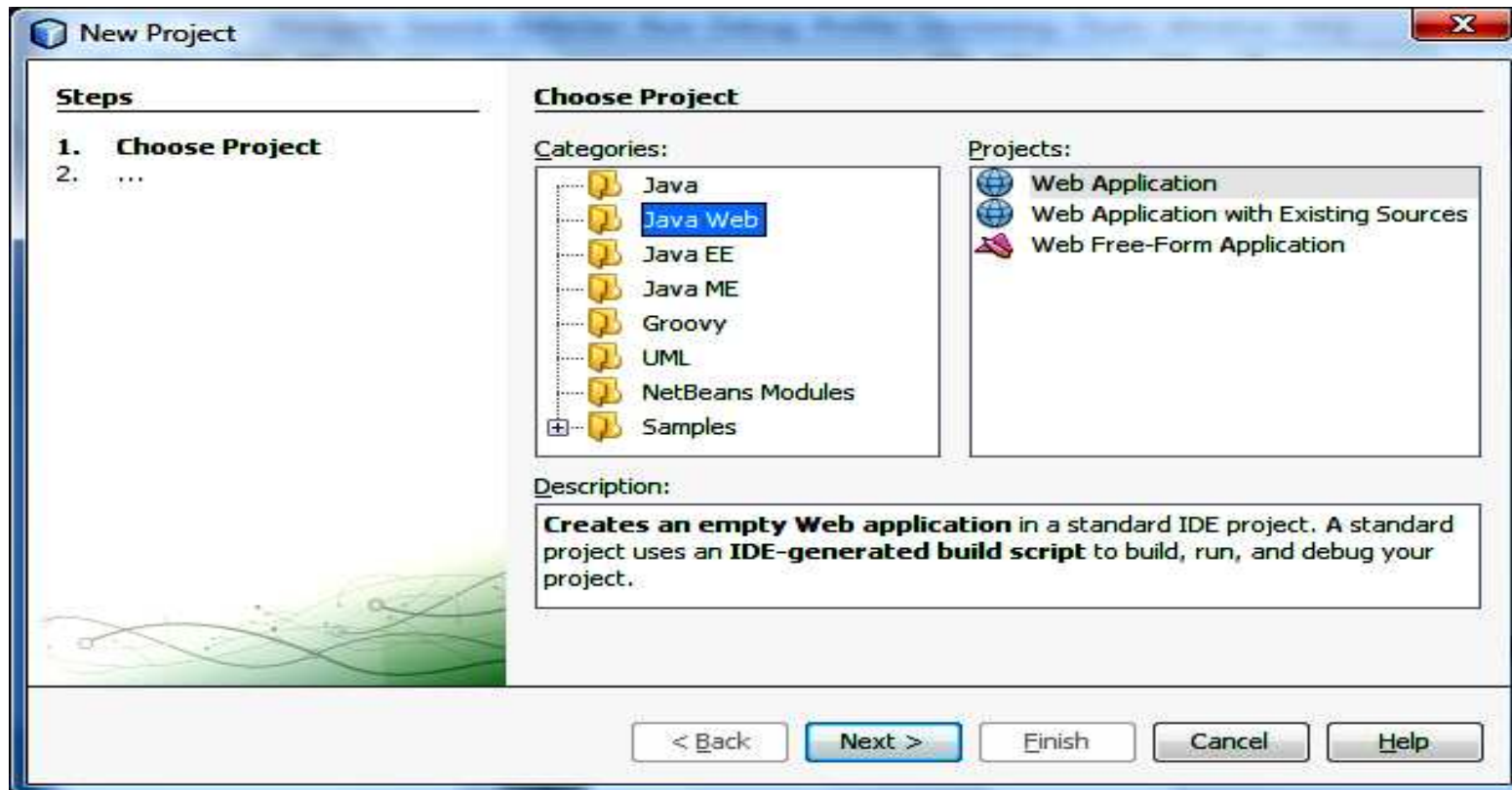
Czynności dodatkowe – druga iteracja

- 5) Zastąpienie elementu typu TextField elementem typu DropDownList**
- 6) Powiązanie elementu typu DropDownList z tabelą bazy danych**
- 7) Dodanie nowych właściwości elementom strony**
- 8) Uruchomienie aplikacji**
- 9) Uzupełnienia aplikacji**

1) Zakładanie projektu



1. Z głównego menu wybierz opcję [File > New Project](#).
2. W [New Project](#), wybierz Java Web listy kategorii ([Categories](#)) i wybierz [Web Application](#) z listy projektów ([Projects](#)).
3. Kliknij Next.
4. Nazwij projekt [AplikacjaInternetowa1 \(Project Name\)](#) i wybierz katalog ([Browse dla Project Location](#)) i kliknij [Next](#).
5. Wybierz [Visual Web JavaServer Faces](#) i kliknij [Finish](#)
6. Pojawi się strona www ([Page1](#)) w oknie [Visual Designer](#).



New Web Application



Steps

1. Choose Project
- 2. Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name: AplikacjaInternetowa1

Project Location: E:\Dydaktyka\d1\Komponenty_2\pkw1

Browse...

Project Folder: E:\Dydaktyka\d1\Komponenty_2\pkw1\AplikacjaInternetowa1

Use Dedicated Folder for Storing Libraries

Libraries Folder:

Browse...

Different users and projects can share the same compilation libraries (see Help for details).

Set as Main Project

< Back

Next >

Finish

Cancel

Help

New Web Application



Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**
4. Frameworks

Server and Settings

Add to Enterprise Application: <None>

Server:

GlassFish V2

Add...

Use dedicated library folder for server JAR files

Java EE Version: Java EE 5

Context Path: /AplikacjaInternetowa1

< Back

Next >

Finish

Cancel

Help

New Web Application



Steps

1. Choose Project
2. Name and Location
3. Server and Settings
4. **Frameworks**

Frameworks

Select the frameworks you want to use in your web application.

- Visual Web JavaServer Faces
- Spring Web MVC 2.5
- JavaServer Faces
- Struts 1.2.9
- Hibernate 3.2.5

Visual Web JavaServer Faces Configuration

Default Java Package:

JSF Servlet Name:

Servlet URL Mapping:

Validate XML

Verify Objects

< Back

Next >

Finish

Cancel

Help



Files | **Services**

- ApplikacjaInternetowa1
 - Web Pages
 - WEB-INF
 - resources
 - Page1.jsp
 - Configuration Files
 - Server Resources
 - Source Packages
 - Test Packages
 - Libraries
 - Test Libraries
 - Project Woodstock Themes
 - Component Libraries
 - Data Source References

Start Page | Page1

Design | JSP | Java | Any Size

Design your page by dragging components from the Palette.

Define a component's behavior by adding code: Either double-click component or right-click the component and choose its event handler.

Page layout is currently set to Grid Layout, which positions components at specific x and y coordinates. If you would like to use flow layout, which positions components left to right and top to bottom, change the Page Layout property to Flow Layout.

Palette

- Woodstock Basic
 - Label
 - Static Text
 - Text Field
 - Text Area
 - Button
 - Hyperlink

Page1 - Properties

- General
 - id: Page1
- Appearance

Page1

- Page1 (request)

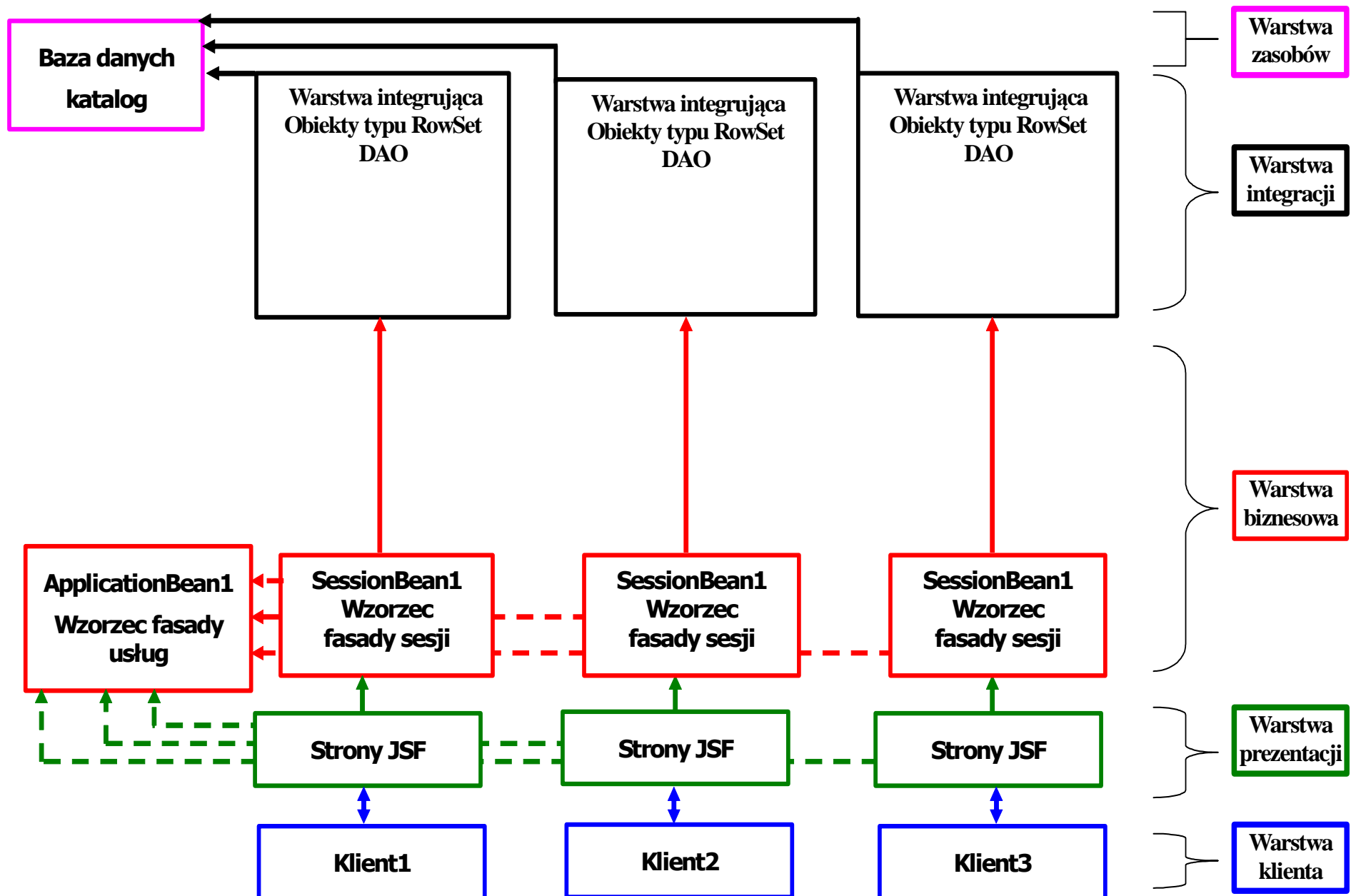
Page1 - Navigator

- Page1
- RequestBean1
- SessionBean1
- ApplicationBean1

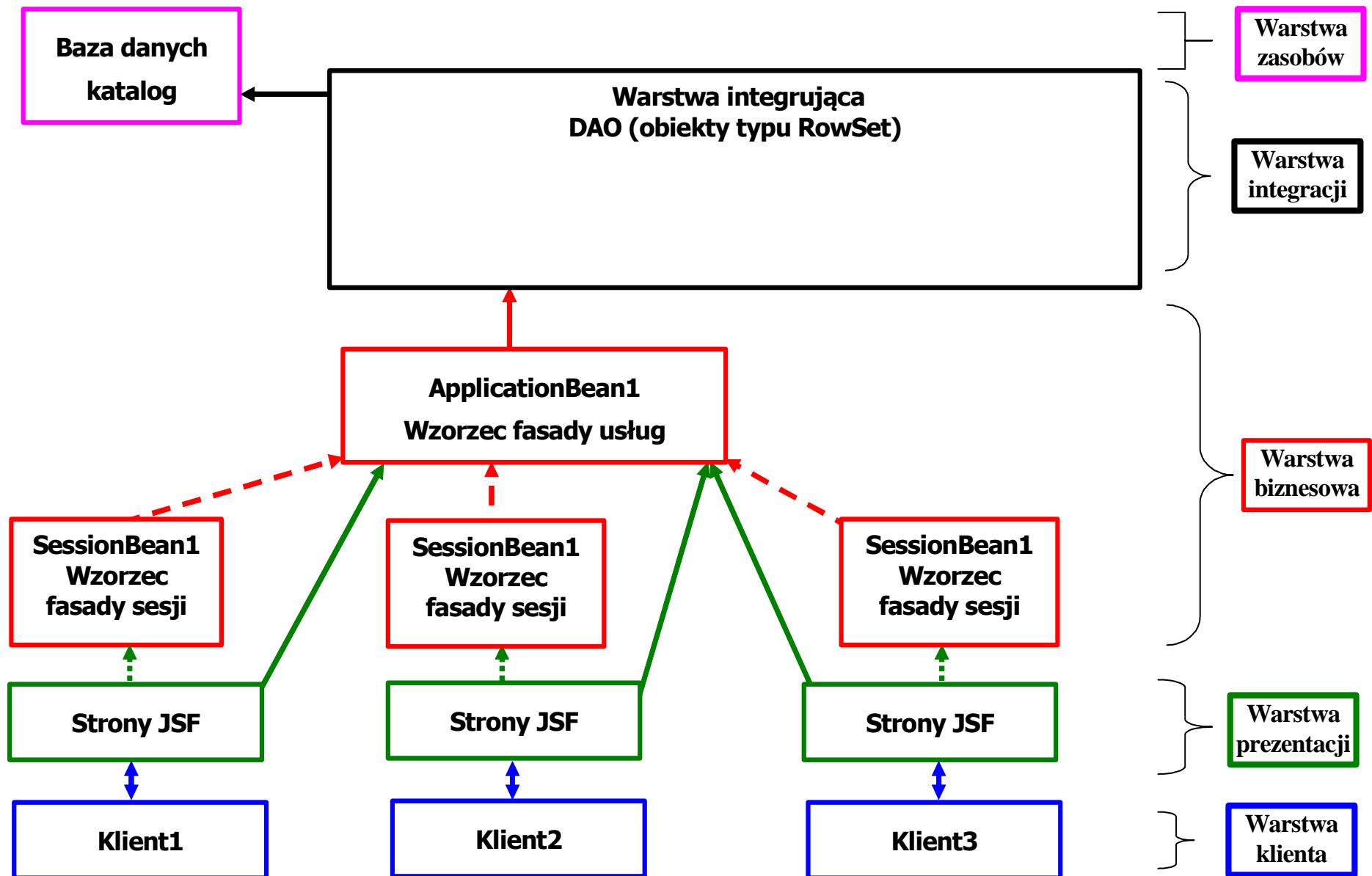
Output | **Tasks**



Architektura aplikacji pięciowarstwowej (przykład z wykładu 5 - punkt 4)- linie przerywane oznaczają powiązania nie wykorzystane w aplikacji

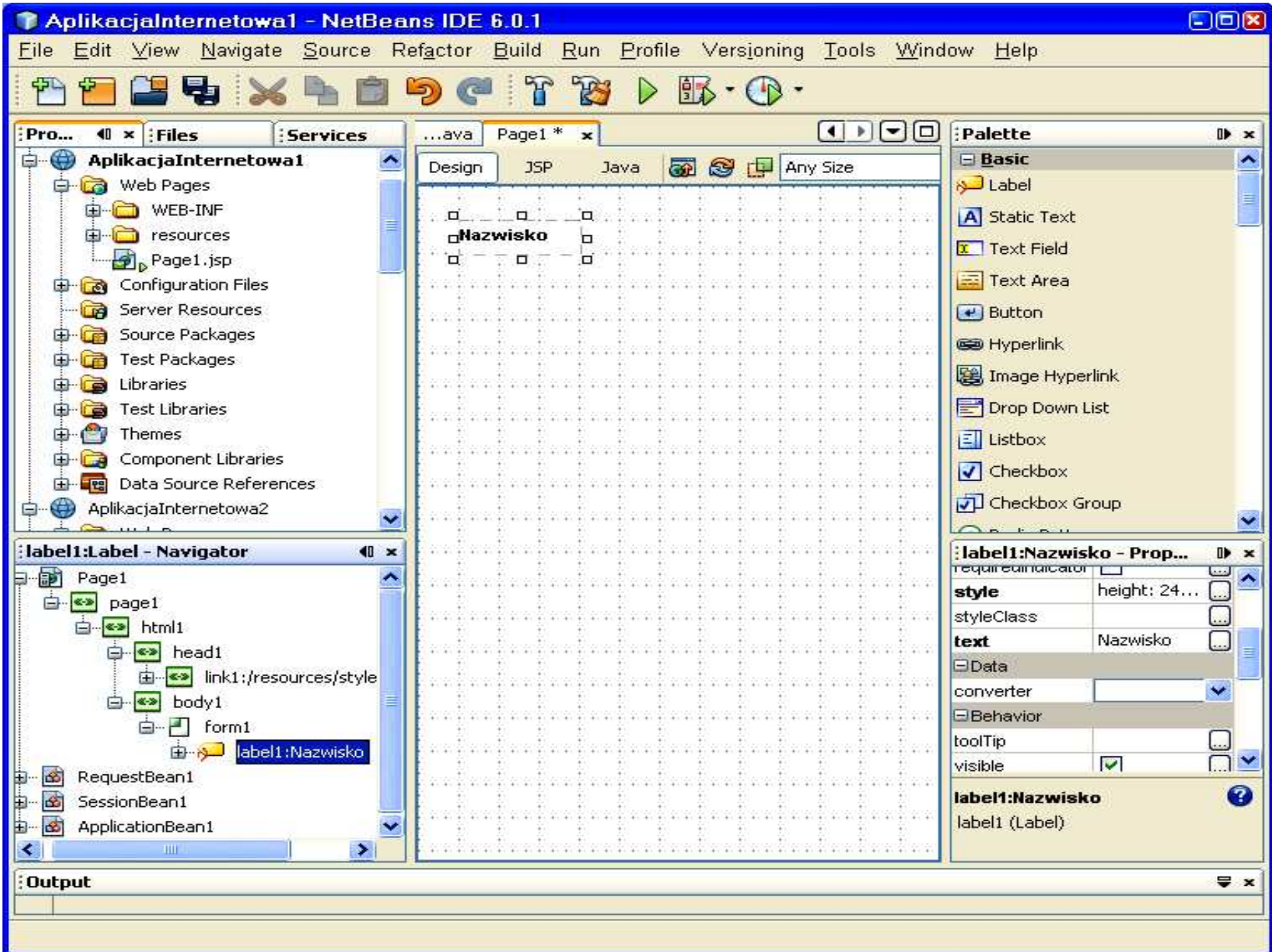


Architektura aplikacji pięciowarstwowej (przykład z wykładu 6 – punkt 4)- linie przerywane oznaczają powiązania nie wykorzystane w aplikacji



2) Projektowanie strony w sposób wizualny

1. Ustaw Paletę komponentów – wybierz **Window > Palette** i przewiń na obszar komponentów typu **Woodstock Basic**
2. Przeciągnij komponent typu **Label** na stronę **Page1** ustawionej w trybie **Visual Design**
3. W **Properties** wpisz we właściwości **text** słowo **Nazwisko:**, to samo można zrobić bezpośrednio na elemencie klikając prawym klawiszem na obszarze elementu i wybierając z wyskakującego menu opcję edycji
4. Przeciągnij z **Palety** komponent typu **Text Field** i połóż go z prawej strony komponentu typu **Label** i wpisz słowa **Podaj nazwisko** wg sposobu z p.2. Naciśnij **Ctrl-Shift-Drag** na komponentcie **Label** i przeciągnij na komponent typu **Text Field** w celu powiązania tych komponentów
5. Przeciągnij z **Palety** komponent typu **Button** i połóż go z prawej strony komponentu typu **TextField**, wpisz na nim słowa: **Dzień dobry** wg sposobu z p.2
6. Przeciągnij z Palety komponent typu **StaticText** i połóż go poniżej komponentu typu **Label**
7. Przeciągnij z **Palety** komponent typu **Message Group** i połóż go poniżej komponentu typu **TextField** – służy on przeglądarce do wyświetlania komunikatów o błędach
8. Na belce narzędziowej okna edycji strony kliknij na JSP – pojawi okno edycji pliku typu **JavaServer Pages (JSP)**
9. Ustaw dla strony **Page1** właściwość **Title** na **Dzień dobry, Web** w oknie **Properties**. Będzie to tytuł umieszczony w oknie przeglądarki
10. Należy na każdy z komponentów kliknąć prawym klawiszem myszy i na wyskakującym menu kliknąć na pozycję **Add Binding Attribute** – spowoduje to wygenerowanie kodu Javy w pliku Page1.java, reprezentujący te komponenty



label1:Label - Navigator

- Page1
 - page1
 - html1
 - head1
 - link1:/resources/style
 - body1
 - form1
 - label1:Nazwisko
- RequestBean1
- SessionBean1
- ApplicationBean1

Palette

Basic

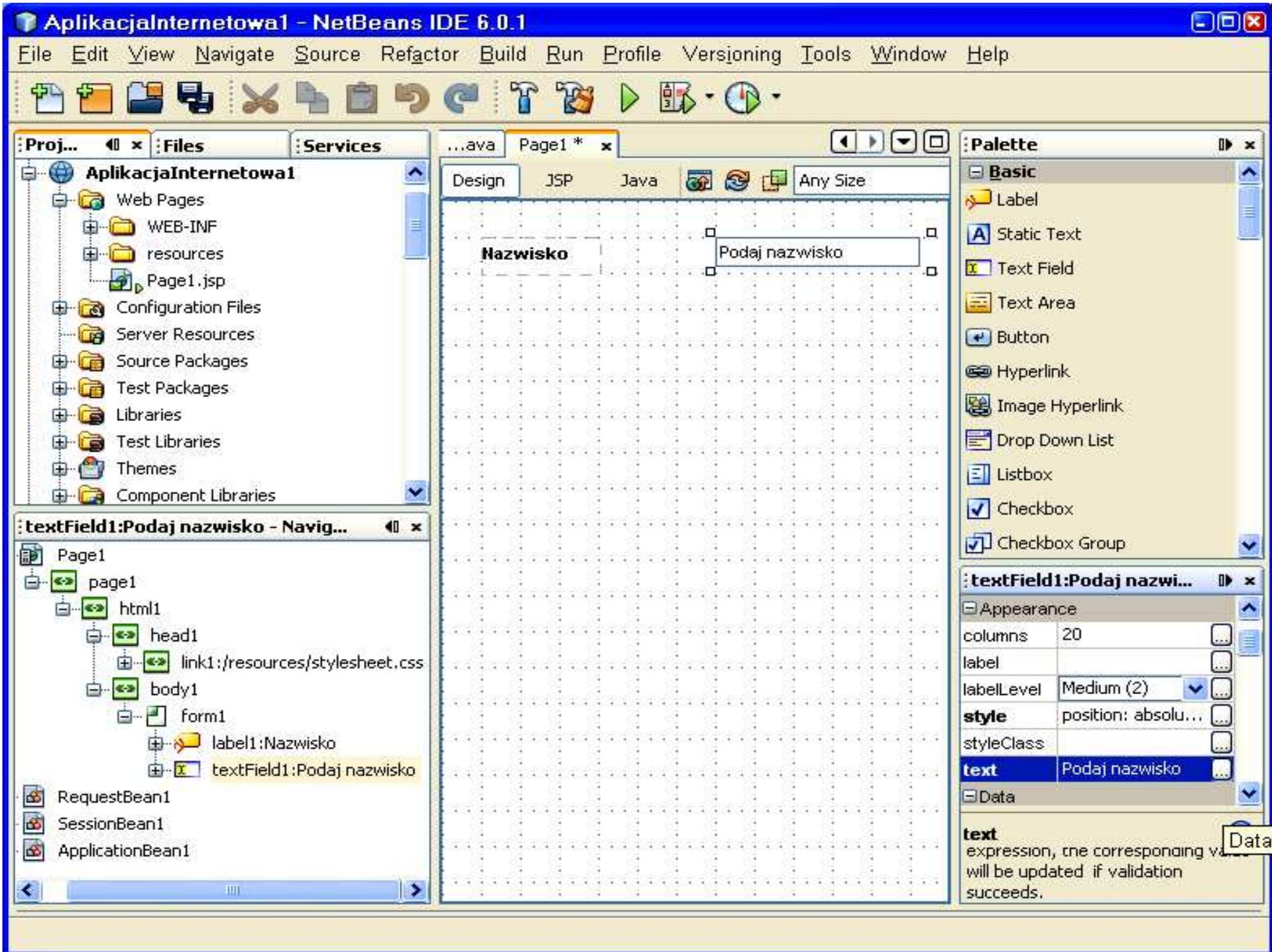
- Label
- Static Text
- Text Field
- Text Area
- Button
- Hyperlink
- Image Hyperlink
- Drop Down List
- Listbox
- Checkbox
- Checkbox Group

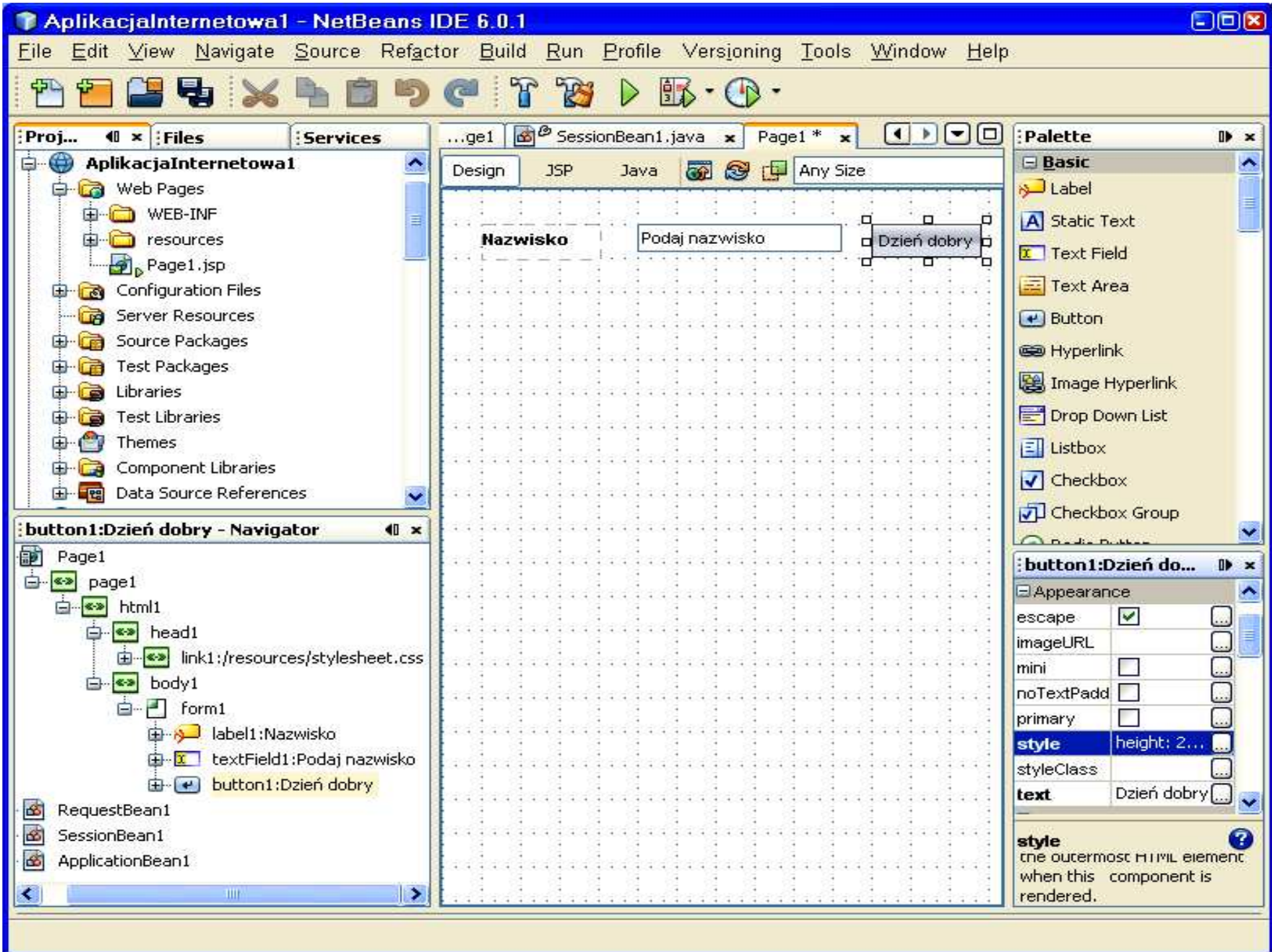
label1:Nazwisko - Prop...

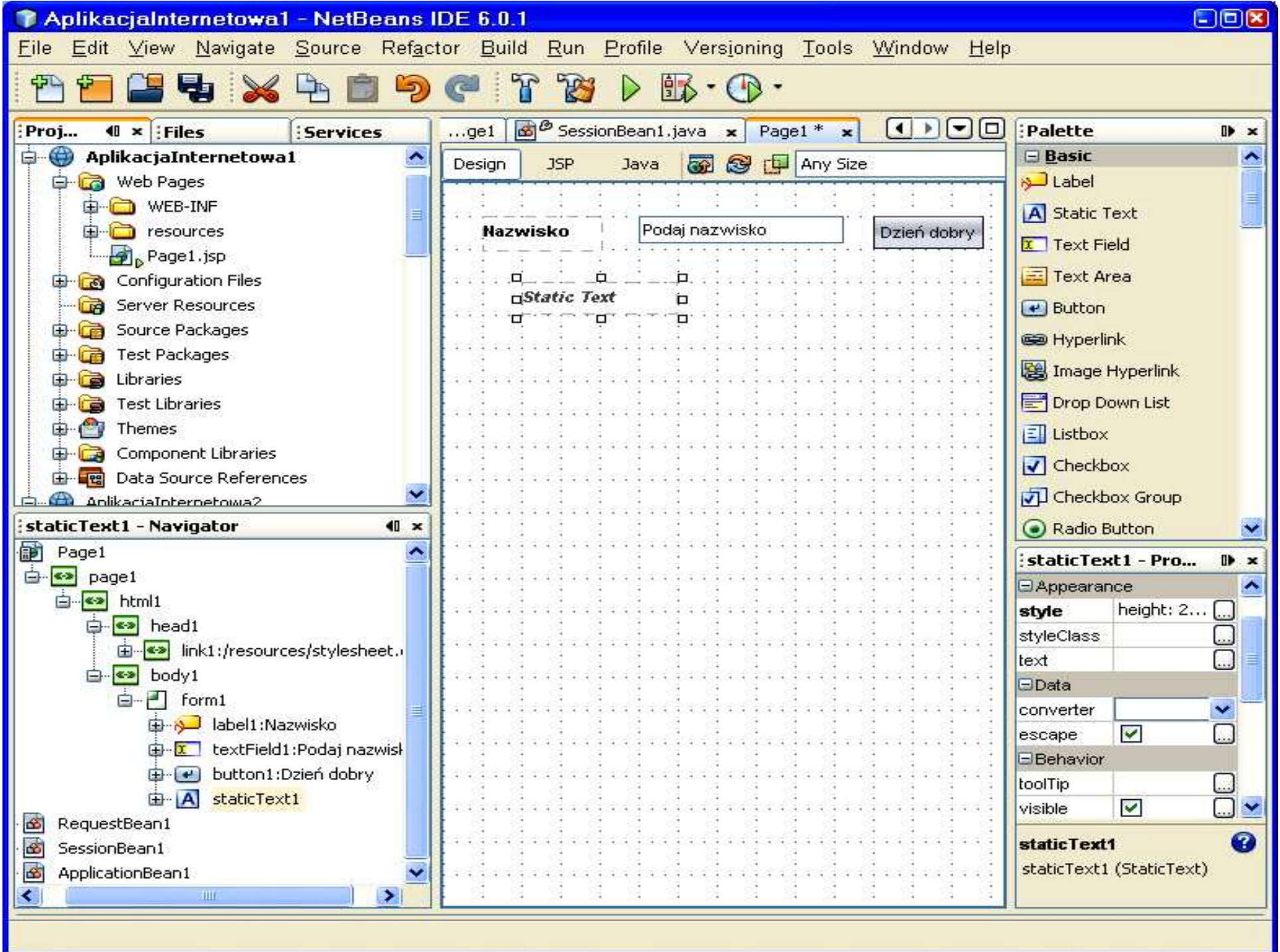
requiredIndicator	<input type="checkbox"/>
style	height: 24...
styleClass	
text	Nazwisko
Data	
converter	
Behavior	
toolTip	
visible	<input checked="" type="checkbox"/>

label1:Nazwisko
label1 (Label)

Output









Proj... **Files** **Services**

- AplikacjaInternetowa1
 - Web Pages
 - WEB-INF
 - resources
 - Page1.jsp

button1:Dzień dobry - Navigator

- Page1
 - page1
 - html1
 - head1
 - body1
 - form1
 - label1:Nazwisko
 - textField1:Podaj nazwisko
 - button1:Dzień dobry
 - staticText1
 - messageGroup1

RequestBean1
SessionBean1
ApplicationBean1

Start Page Page1 *
Design JSP Java Any Size

Nazwisko Podaj nazwisko Dzień dobry

Static Text

System Messages
List of all message summaries

Palette

- Woodstock B...
- Label
- Static Text
- Text Field
- Text Area
- Button

button1:Dzień d...

General

id bu...

Appearance

escape

image IRI

button1 (Button)

Output **Tasks**



Search (Ctrl+I)

Project Explorer showing the structure of 'AplikacjaInternetowa1' with folders like 'Web Pages', 'WEB-INF', 'resources', and 'Page1.jsp'.

Navigator showing the 'label1:Nazwisko' component with sub-elements like 'Page1', 'RequestBean1', 'SessionBean1', and 'ApplicationBean1'.

Design view tabs for 'Page1 *' with sub-tabs for 'Design', 'JSP', and 'Java'. A dropdown menu shows 'Any Size'.

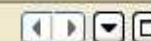
Main design canvas showing a grid with a label component. A context menu is open over the label, listing actions such as 'Edit Java Source', 'Add Binding Attribute', 'Edit Label Text', and 'Cut'.

Output

Palette showing various UI components categorized into 'Woodstock Basic', 'Woodstock Layout', 'Woodstock Composite', 'Validators', and 'Converters'. Components include 'Label', 'Text Field', 'Table', and 'Form'.

Properties window for 'label1:Nazwisko' showing the 'General' tab with the 'id' property set to 'label1'.

Tasks



Start Page x Page1 x Page1 x @ SessionBean1.java x Page1 * x

Design

JSP

Java



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3     Document    : Page1
4     Created on  : 2008-02-29, 20:56:35
5     Author     : ZK
6 -->
7 <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsp/core" xmlns:h="http://java.sun.com/jsp/html" xmlns:
8     <jsp:directive.page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"/>
9     <f:view>
10         <webuijsf:page binding="#{Page1.page1}" id="page1">
11             <webuijsf:html binding="#{Page1.html1}" id="html1">
12                 <webuijsf:head binding="#{Page1.head1}" id="head1">
13                     <webuijsf:link binding="#{Page1.link1}" id="link1" url="/resources/stylesheet.css"/>
14                 </webuijsf:head>
15                 <webuijsf:body binding="#{Page1.body1}" id="body1" style="-rave-layout: grid">
16                     <webuijsf:form binding="#{Page1.form1}" id="form1">
17                         <webuijsf:label binding="#{Page1.label1}" id="label1" style="height: 24px; left: 24px;
18                         <webuijsf:textField binding="#{Page1.textField1}" id="textField1" style="left: 120px
19                         <webuijsf:button binding="#{Page1.button1}" id="button1" style="height: 24px; left:
20                         <webuijsf:staticText binding="#{Page1.staticText1}" id="staticText1" style="height:
21                         <webuijsf:messageGroup binding="#{Page1.messageGroup1}" id="messageGroup1" style="po
22                     </webuijsf:form>
23                 </webuijsf:body>
24             </webuijsf:html>
25         </webuijsf:page>
26     </f:view>
27 </jsp:root>
```

1:1

INS

3) Dodanie nowych właściwości elementom strony

Obsługa zdarzeń

1. Z belki narzędziowej edytora strony **Page1** należy wybrać opcje **Design** w celu przełączenia się do trybu **Visual Designer**.
2. Należy podwójnie kliknąć na komponent typu **Button**.
3. Po kliknięciu pojawi się okno edycji dla pliku Java z klasa dziedziczącą od klasy **AbstractPageBean** w miejscu metody **button1_action()**
4. Należy wpisać do jej ciała następujące instrukcje

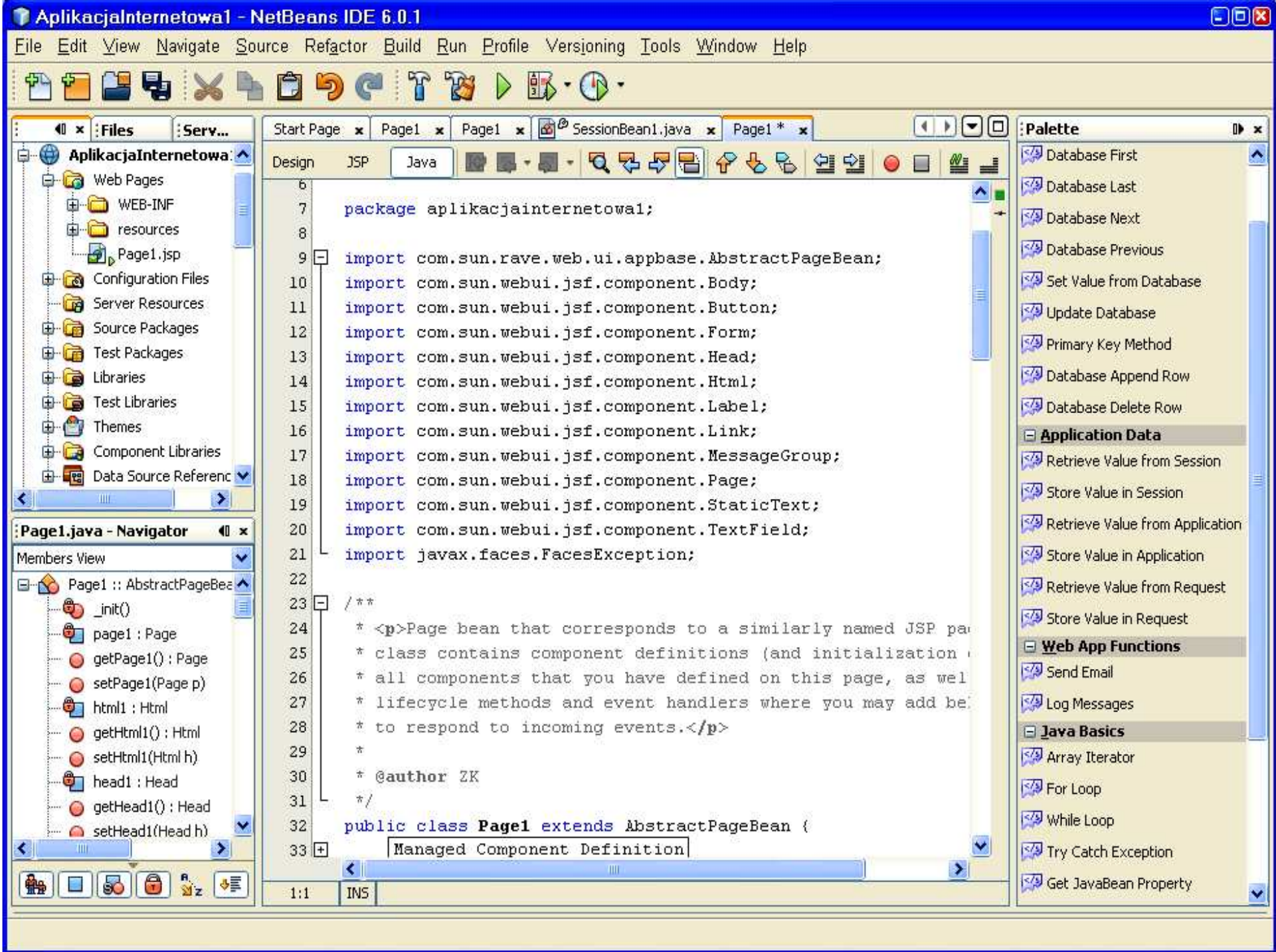
```
String name = (String)textField1.getText();
```

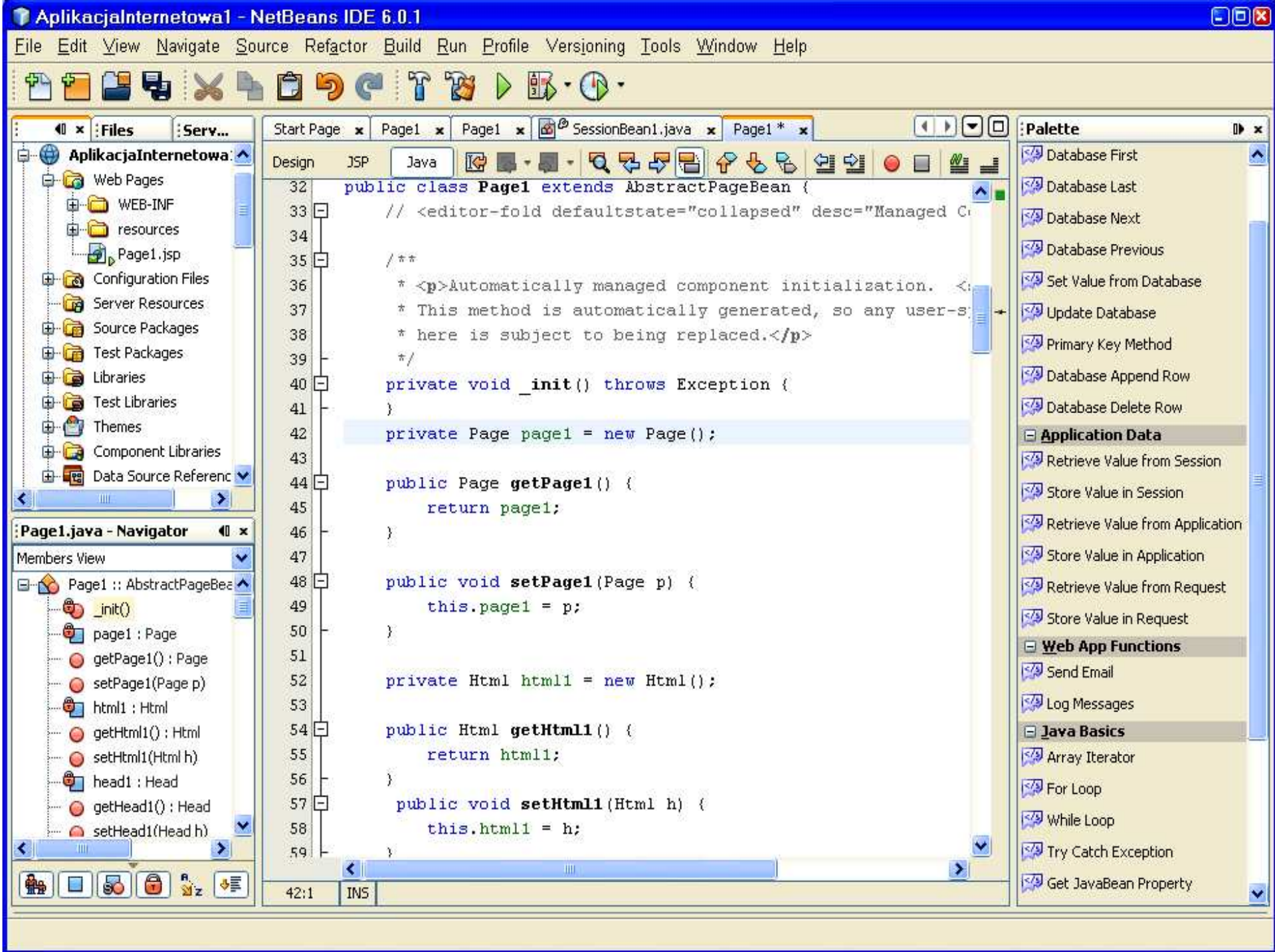
```
//pobranie z pola wejściowego tekstowego typu TextField wpisane słowa
```

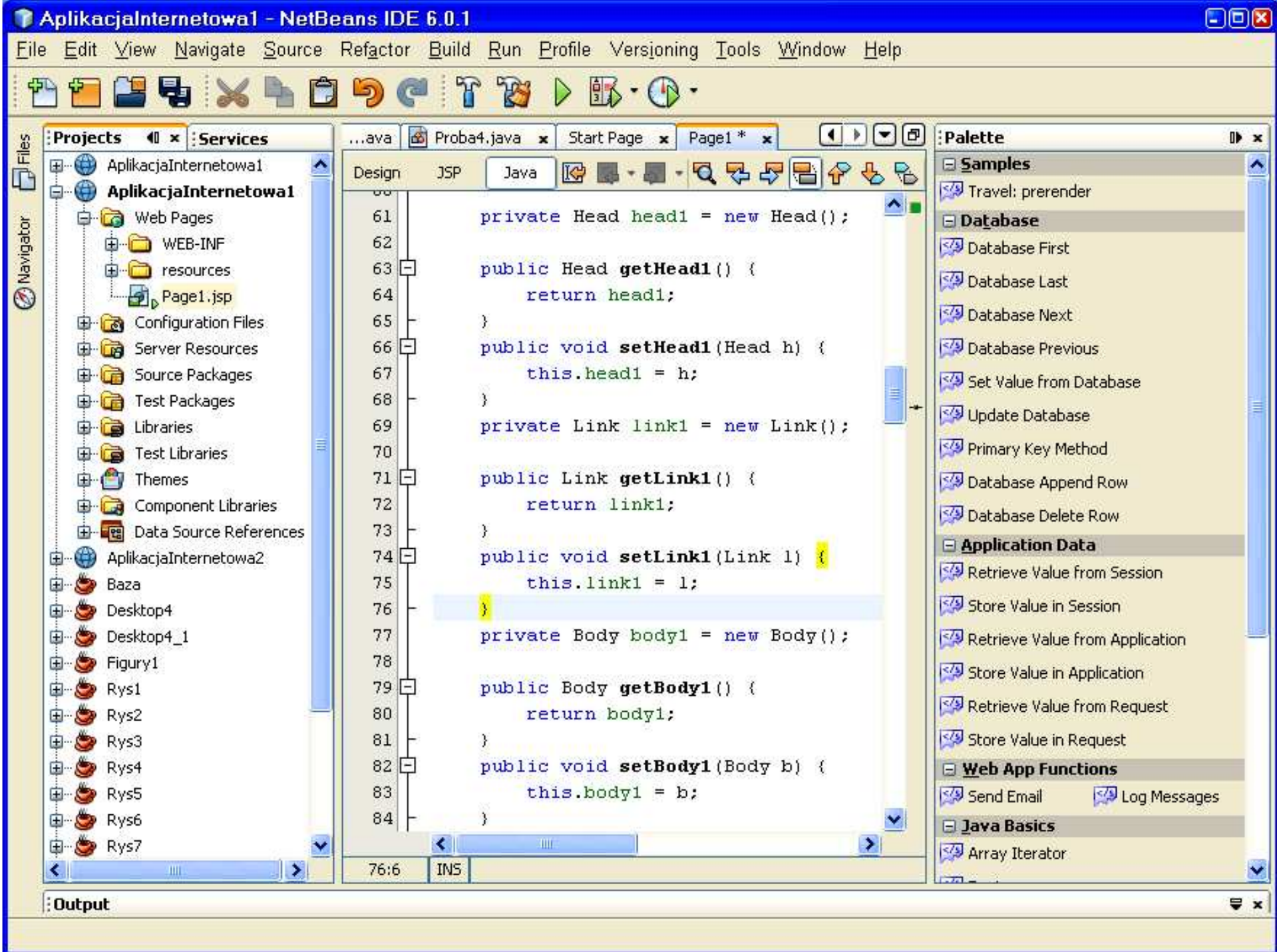
```
staticText1.setText("Dzień dobry, " + name + "!");
```

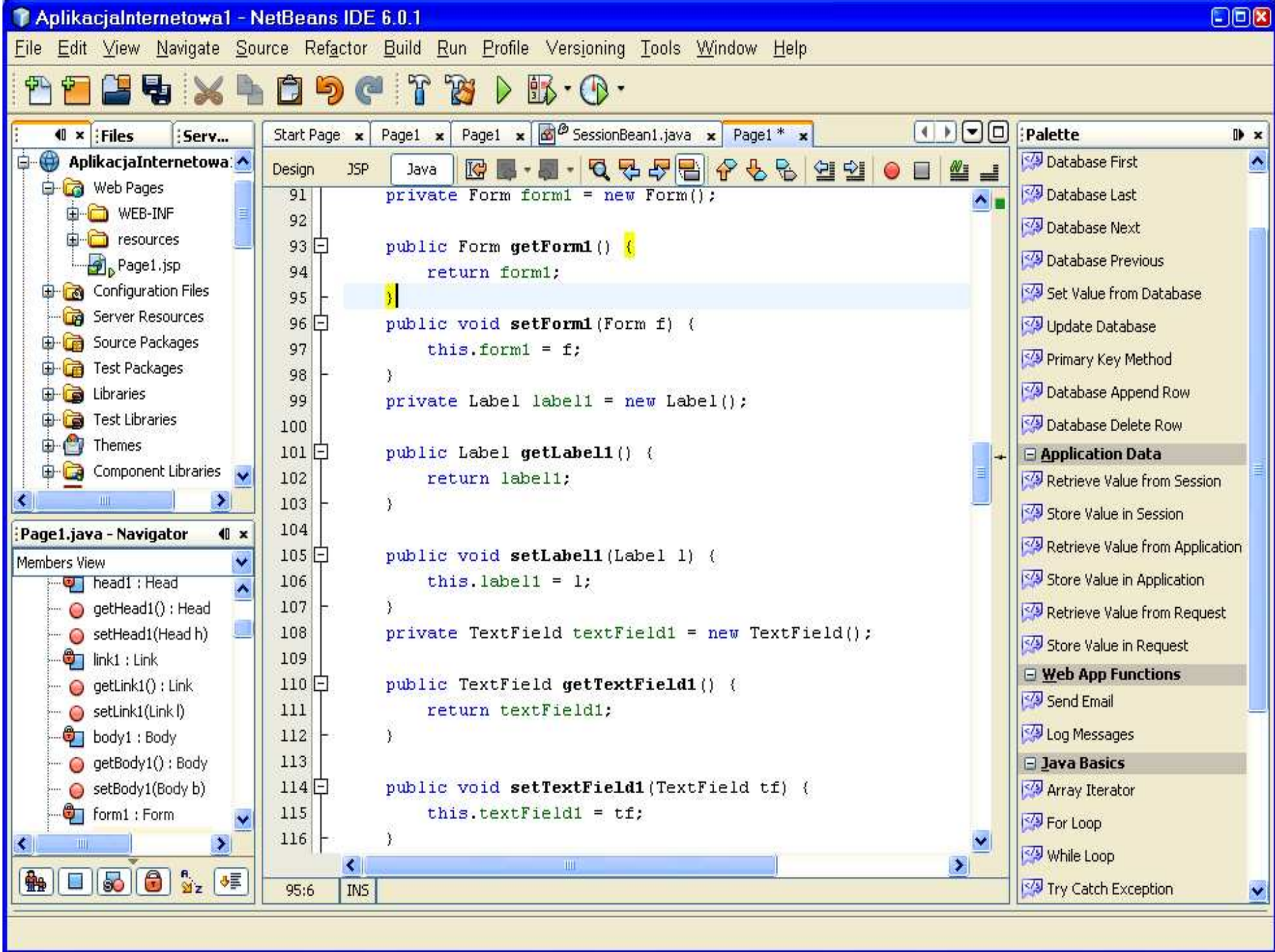
```
// i wyświetlenie ich w polu typu StaticText po naciśnięciu klawisza typu Button
```

```
i naciśnij Alt+Shift+F w celu uaktualnienia kodu
```









AplikacjaInternetowa1 - NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

The image shows the NetBeans IDE 6.0.1 interface. The main window is titled "AplikacjaInternetowa1 - NetBeans IDE 6.0.1". The menu bar includes "File", "Edit", "View", "Navigate", "Source", "Refactor", "Build", "Run", "Profile", "Versioning", "Tools", "Window", and "Help". The toolbar contains various icons for file operations, editing, and running. The main editor area shows a Java file named "Page1.java" with the following code:

```
117 private Button button1 = new Button();
118
119 public Button getButton1() {
120     return button1;
121 }
122
123 public void setButton1(Button b) {
124     this.button1 = b;
125 }
126 private StaticText staticText1 = new StaticText();
127
128 public StaticText getStaticText1() {
129     return staticText1;
130 }
131
132 public void setStaticText1(StaticText st) {
133     this.staticText1 = st;
134 }
135 private MessageGroup messageGroup1 = new MessageGroup();
136
137 public MessageGroup getMessageGroup1() {
138     return messageGroup1;
139 }
140 public void setMessageGroup1(MessageGroup mg) {
141     this.messageGroup1 = mg;
142 }
```

The code is displayed in a "Design" view. The "Page1.java - Navigator" on the left shows the following members:

- getTextField1(): TextField
- setTextField1(TextField tf)
- button1: Button
- getButton1(): Button
- setButton1(Button b)
- staticText1: StaticText
- getStaticText1(): StaticText
- setStaticText1(StaticText st)
- messageGroup1: MessageGrc
- getMessageGroup1(): Messag

The "Palette" on the right contains the following items:

- Database First
- Database Last
- Database Next
- Database Previous
- Set Value from Database
- Update Database
- Primary Key Method
- Database Append Row
- Database Delete Row
- Application Data
 - Retrieve Value from Session
 - Store Value in Session
 - Retrieve Value from Application
 - Store Value in Application
 - Retrieve Value from Request
 - Store Value in Request
- Web App Functions
 - Send Email
 - Log Messages
- Java Basics
 - Array Iterator
 - Iterates through an ArrayList
 - While Loop
 - Try Catch Exception

AplikacjaInternetowa1 - NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

The image shows the NetBeans IDE 6.0.1 interface. The main editor window displays the code for `Page1.java`. The code includes the `init()` method, which performs initializations inherited from the superclass and application-specific initializations. The code is as follows:

```
149 public Page1() {
150 }
151 /**...*/
152 @Override
153 public void init() {
154     // Perform initializations inherited from our superclass
155     super.init();
156     // Perform application initialization that must complete
157     // *before* managed components are initialized
158     // TODO - add your own initialization code here
159
160     // <editor-fold defaultstate="collapsed" desc="Managed Co
161     // Initialize automatically managed components
162     // *Note* - this logic should NOT be modified
163     try {
164         _init();
165     } catch (Exception e) {
166         log("Page1 Initialization Failure", e);
167         throw e instanceof FacesException ? (FacesException)
168     }
169     // </editor-fold>
170     // Perform application initialization that must complete
171     // *after* managed components are initialized
172     // TODO - add your own initialization code here
173 }
174
```

The left sidebar shows the project structure for `AplikacjaInternetowa1`, including `Web Pages`, `WEB-INF`, `resources`, and `Page1.jsp`. The `Page1.java - Navigator` shows the class members, including `getButton1()`, `setButton1()`, `staticText1`, `getMessageGroup1()`, `setMessageGroup1()`, `Page1()`, and `init()`.

The right sidebar shows the `Palette` of web application functions, including `Database First`, `Database Last`, `Database Next`, `Database Previous`, `Set Value from Datas`, `Update Database`, `Primary Key Method`, `Database Append Row`, `Database Delete Row`, `Application Data`, `Retrieve Value from Ses`, `Store Value in Session`, `Retrieve Value from App`, `Store Value in Applicatio`, `Retrieve Value from Rec`, `Store Value in Request`, `Web App Functions`, `Send Email`, `Log Messages`, `Java Basics`, `Array Iterator`, `For Loop`, `While Loop`, and `Try Catch Exception`.



Start Page x Page1 x Page1 x SessionBean1.java x Page1 * x

Design JSP Java

```

186  /**
187   * <p>Callback method that is called after the component tree has been
188   * restored, but before any event processing takes place. This method
189   * will <strong>only</strong> be called on a postback request that
190   * is processing a form submit. Customize this method to allocate
191   * resources that will be required in your event handlers.</p>
192   */
193   @Override
194   public void preprocess() {
195   }
196   /**
197   * <p>Callback method that is called just before rendering takes place.
198   * This method will <strong>only</strong> be called for the page that
199   * will actually be rendered (and not, for example, on a page that
200   * handled a postback and then navigated to a different page). Customiz
201   * this method to allocate resources that will be required for rendering
202   * this page.</p>
203   */
204   @Override
205   public void prerender() {
206   }
207   /**
208   * <p>Callback method that is called after rendering is completed for
209   * this request, if <code>init()</code> was called (regardless of whethe
210   * or not this was the page that was actually rendered). Customize this
211   * method to release resources acquired in the <code>init()</code>,
212   * <code>preprocess()</code>, or <code>prerender()</code> methods (or
213   * acquired during execution of an event handler).</p>
214   */
215   @Override
216   public void destroy() { }

```

216:37 INS

ApplikacjaInternetowa

- Web Pages
 - WEB-INF
 - resources
 - Page1.jsp
- Configuration Files
- Server Resources
- Source Packages
- Test Packages
- Libraries
- Test Libraries
- Themes
- Component Libraries
- Data Source Referenc
- ApplikacjaInternetowa2
 - Web Pages

Page1.java - Navigator

Members View

- getMessageGroup1(): MessageG
- setMessageGroup1(MessageG
- Page1()
- init()
- preprocess()
- prerender()
- destroy()
- getSessionBean1(): SessionBe
- getRequestBean1(): Request
- getApplicationBean1(): Applic
- afterApplyRequestValues()
- afterInvokeApplication()
- afterPhase/PhaseEvent event

Datab

Datab

Datab

Datab

Set Va

Updat

Primar

Datab

Datab

Ap...

Retrie

Store

Retrie

Store

Retrie

Store

We...

Send I

Log M

Ja...

Array

For Lc

While

Try Co

Get Ja

If-The

Switch

Conce

AplikacjaInternetowa1 - NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Start Page x Page1 x Page1 x SessionBean1.java x Page1 * x

Design JSP Java

```
217 /**
218  * <p>Return a reference to the scoped data bean.</p>
219  *
220  * @return reference to the scoped data bean
221  */
222 protected SessionBean1 getSessionBean1() {
223     return (SessionBean1) getBean("SessionBean1");
224 }
225 /**
226  * <p>Return a reference to the scoped data bean.</p>
227  *
228  * @return reference to the scoped data bean
229  */
230 protected RequestBean1 getRequestBean1() {
231     return (RequestBean1) getBean("RequestBean1");
232 }
233 /**
234  * <p>Return a reference to the scoped data bean.</p>
235  *
236  * @return reference to the scoped data bean
237  */
238 protected ApplicationBean1 getApplicationBean1() {
239     return (ApplicationBean1) getBean("ApplicationBean1");
240 }
241 }
```

216:37 INS

Members View

- getMessageGroup1() : MessageG
- setMessageGroup1(MessageG
- Page1()
- init()
- preprocess()
- prerender()
- destroy()
- getSessionBean1() : SessionBe
- getRequestBean1() : Request
- getApplicationBean1() : Applic

Datab

Datab

Datab

Datab

Set Va

Updat

Primar

Datab

Datab

Ap...

Retrie

Store

Retrie

Store

Retrie

Store

We...

Send I

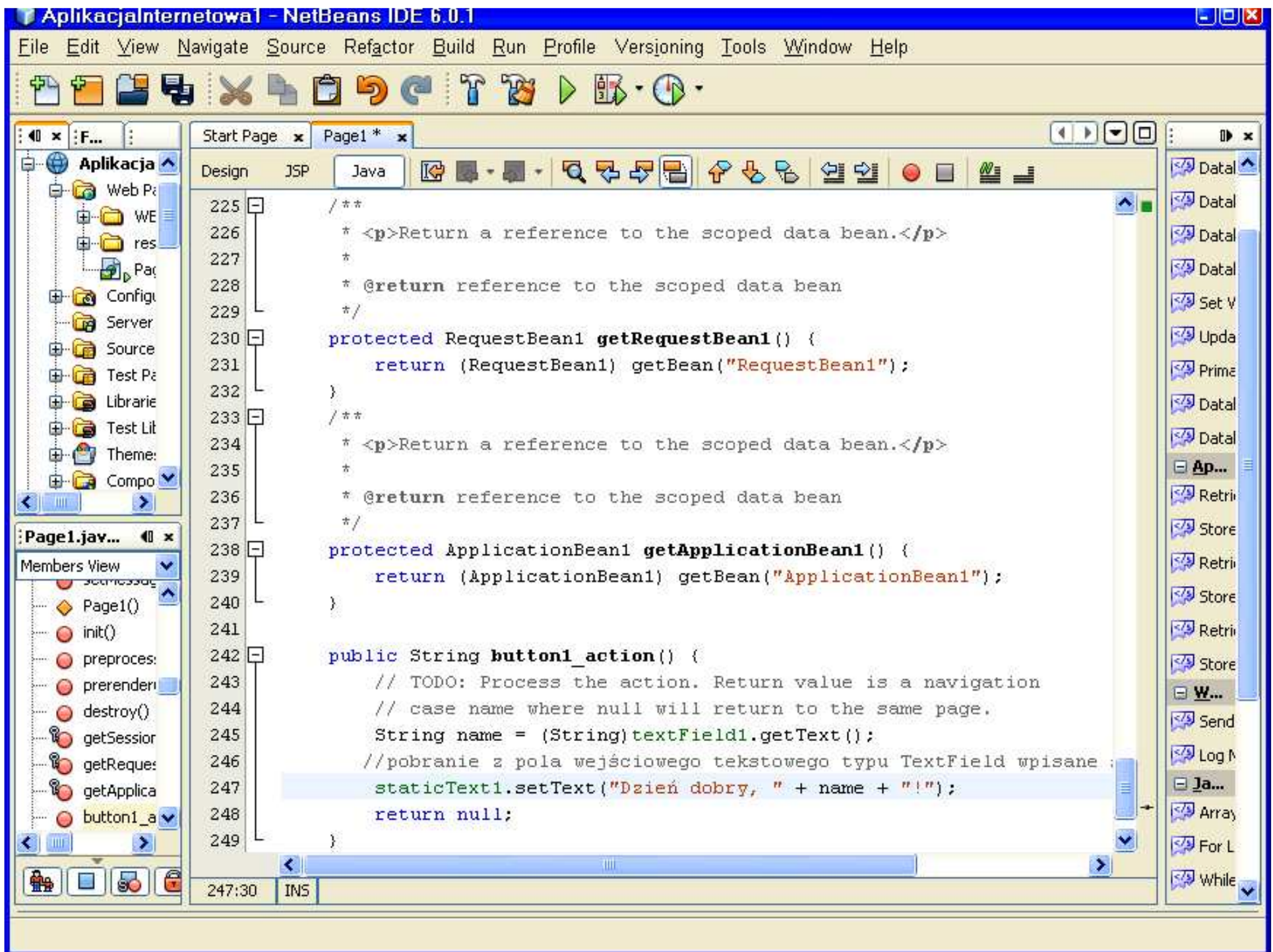
Log M

Ja...

Array

For Lc

While



4) Definiowanie metody `button1_action()`

The screenshot shows the NetBeans IDE 6.0.1 interface. The main editor window displays the source code of a JSP page. A context menu is open over the code, with the 'Run' option highlighted. The code defines several methods: `getSessionBean1()`, `getRequestBean1()`, `getApplicationBean1()`, and `button1_action()`. The `button1_action()` method is highlighted with a red box and contains the following code:

```
public String button1_action() {  
    // TODO: Process the action. Return value is a navigation  
    // case name where null will return to the same page.  
    String name = (String)textField1.getText();  
    //pobranie z pola wejściowego tekstowego typu TextField wpisane słowa  
    staticText1.setText("Dzień dobry, " + name + "!");  
    return null;  
}
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Build, Run, Profile, Versioning, Tools, Window, Help), a toolbar, and a project explorer on the left showing the project structure. The status bar at the bottom indicates the current cursor position at line 237, column 8, in the 'INS' mode.



: Output

▶▶ AplikacjaInternetowa1 (run) x Java DB Database Process x GlassFish V2 x

```
Copying 1 file to C:\Documents and Settings\ZK\My Documents\NetBeansProjects\AplikacjaInternetowa1\build\web\WEB-INF\classes
Copying 1 file to C:\Documents and Settings\ZK\My Documents\NetBeansProjects\AplikacjaInternetowa1\build\web\WEB-INF\classes
Copying 1 file to C:\Documents and Settings\ZK\My Documents\NetBeansProjects\AplikacjaInternetowa1\build\web\WEB-INF\classes
Copying 1 file to C:\Documents and Settings\ZK\My Documents\NetBeansProjects\AplikacjaInternetowa1\build\web\WEB-INF\classes
Copying 1 file to C:\Documents and Settings\ZK\My Documents\NetBeansProjects\AplikacjaInternetowa1\build\web\WEB-INF\classes
Copying 1 file to C:\Documents and Settings\ZK\My Documents\NetBeansProjects\AplikacjaInternetowa1\build\web\WEB-INF\classes
Copying 1 file to C:\Documents and Settings\ZK\My Documents\NetBeansProjects\AplikacjaInternetowa1\build\web\WEB-INF\classes
library-inclusion-in-manifest:
Compiling 4 source files to C:\Documents and Settings\ZK\My Documents\NetBeansProjects\AplikacjaInternetowa1\build\web\WEB-INF\classes
Copying 1 file to C:\Documents and Settings\ZK\My Documents\NetBeansProjects\AplikacjaInternetowa1\build\web\WEB-INF\classes
compile:
compile-jsps:
Undeploying ...
While undeploying, trying to stop application in target server completed successfully
While undeploying, trying to remove reference for application in target server completed successfully
Undeploying the application
Trying to undeploy application from domain completed successfully
```

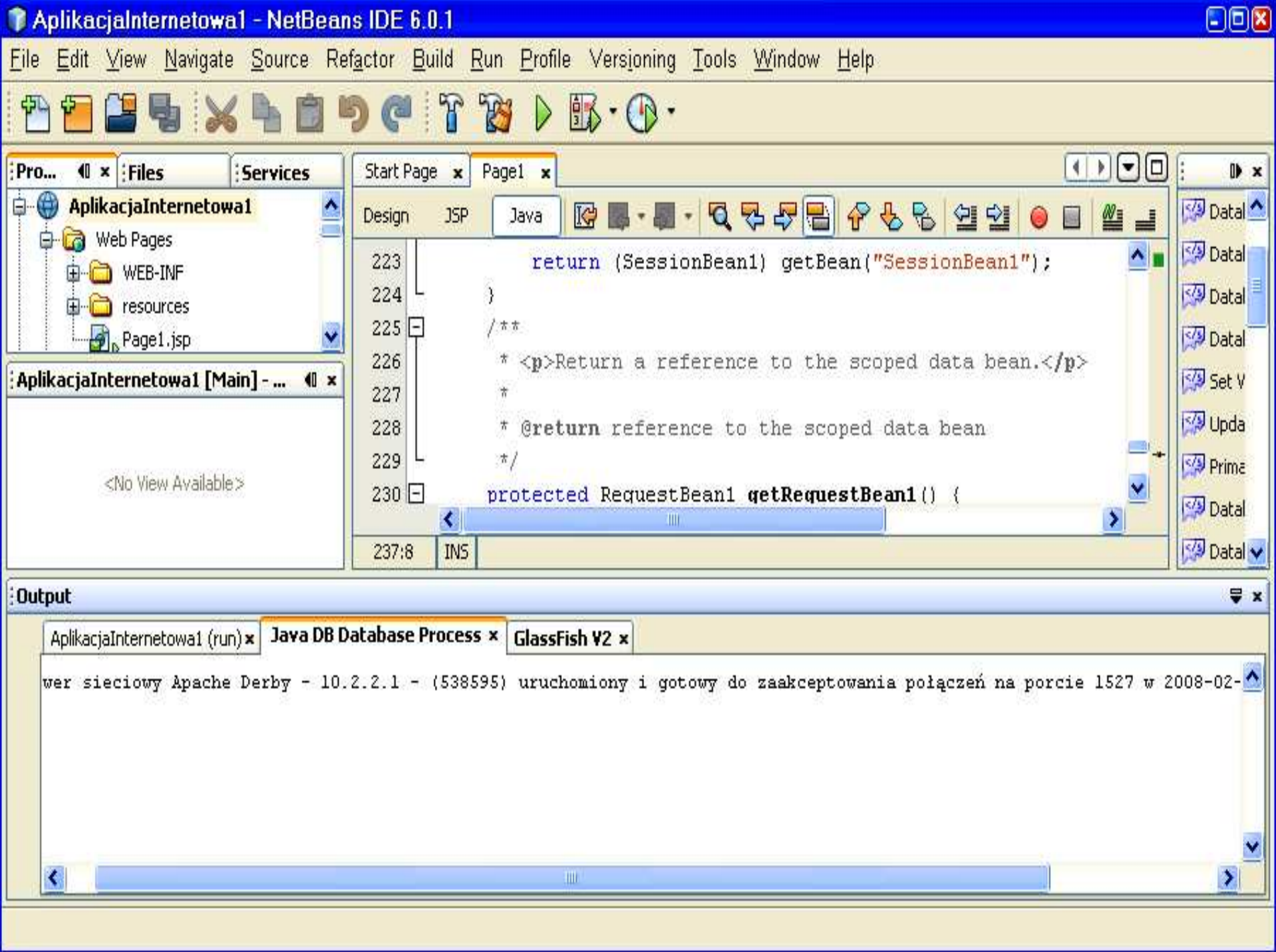


Project: AplikacjaInternetowa1
 Start Page x Page1 x
 Design JSP Java
 237:8 INS

Output

```

AplikacjaInternetowa1 (run) x Java DB Database Process x GlassFish V2 x
Undeployment of application AplikacjaInternetowa1 completed successfully
All operations completed successfully
In-place deployment at C:\Documents and Settings\ZK\My Documents\NetBeansProjects\AplikacjaInternetowa1\build\w
Start registering the project's server resources
Finished registering server resources
moduleID=AplikacjaInternetowa1
deployment started : 0%
deployment finished : 100%
Deploying application in domain completed successfully
Trying to create reference for application in target server completed successfully
Trying to start application in target server completed successfully
Deployment of application AplikacjaInternetowa1 completed successfully
Enable of AplikacjaInternetowa1 in target server completed successfully
Enable of application in all targets completed successfully
All operations completed successfully
run-deploy:
Browsing: http://localhost:8090/AplikacjaInternetowa1/
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 4 minutes 0 seconds)
    
```



Output

AplikacjaInternetowa1 (run) x Java DB Database Process x GlassFish V2 x

```
2008-02-29 22:03:36 com.sun.enterprise.admin.servermgmt.launch.ASLauncher buildCommand
INFO:
C:/Program Files/Java/jdk1.6.0_02/bin/java
-Dcom.sun.aas.instanceRoot=C:/Program Files/glassfish-v2url/domains/domain1
-Dcom.sun.aas.ClassPathPrefix=
-Dcom.sun.aas.ClassPathSuffix=
-Dcom.sun.aas.ServerClassPath=
-Dcom.sun.aas.classloader.appserverChainJars.ee=
-Dcom.sun.aas.classloader.appserverChainJars=admin-cli.jar,admin-cli-ee.jar,j2ee-svc.jar
-Dcom.sun.aas.classloader.excludesList=admin-cli.jar,appserv-upgrade.jar,sun-appserv-ant.jar
-Dcom.sun.aas.classloader.optionalOverrideableChain.ee=
-Dcom.sun.aas.classloader.optionalOverrideableChain=webservices-rt.jar,webservices-tools.jar
-Dcom.sun.aas.classloader.serverClassPath.ee=/lib/hadbjdbc4.jar,C:/Program Files/glassfish-v2url/lib/SUNWjdmk/5
-Dcom.sun.aas.classloader.serverClassPath=C:/Program Files/glassfish-v2url/lib/install/applications/jmsra/imqjr
-Dcom.sun.aas.classloader.sharedChainJars.ee=appserv-se.jar,appserv-ee.jar,jesmf-plugin.jar,/lib/dbstate.jar,/l
-Dcom.sun.aas.classloader.sharedChainJars=javaee.jar,C:/Program Files/Java/jdk1.6.0_02/lib/tools.jar,install/ap
-Dcom.sun.aas.configName=server-config
-Dcom.sun.aas.configRoot=C:/Program Files/glassfish-v2url/config
-Dcom.sun.aas.defaultLogFile=C:/Program Files/glassfish-v2url/domains/domain1/logs/server.log
-Dcom.sun.aas.domainName=domain1
-Dcom.sun.aas.installRoot=C:/Program Files/glassfish-v2url
```



Output

```
AplikacjaInternetowa1 (run) x Java DB Database Process x GlassFish V2 x
-Dcom.sun.aas.instanceName=server
-Dcom.sun.aas.processLauncher=SE
-Dcom.sun.aas.promptForIdentity=true
-Dcom.sun.enterprise.config.config_environment_factory_class=com.sun.enterprise.config.serverbeans.AppserverCore
-Dcom.sun.enterprise.overrideablejavaxpackages=javax.help,javax.portlet
-Dcom.sun.enterprise.taglibs=appserv-jstl.jar,jspf-impl.jar
-Dcom.sun.enterprise.taglisteners=jspf-impl.jar
-Dcom.sun.updatecenter.home=C:/Program Files/glassfish-v2url/updatecenter
-Ddomain.name=domain1
-Djava.endorsed.dirs=C:/Program Files/glassfish-v2url/lib/endorsed
-Djava.ext.dirs=C:/Program Files/Java/jdk1.6.0_02/lib/ext;C:/Program Files/Java/jdk1.6.0_02/jre/lib/ext;C:/Program Files/Java/jdk1.6.0_02/lib/ext
-Djava.library.path=C:\Program Files\glassfish-v2url\lib;C:\Program Files\glassfish-v2url\lib;C:\Program Files\Java\jdk1.6.0_02\bin
-Djava.security.auth.login.config=C:/Program Files/glassfish-v2url/domains/domain1/config/login.conf
-Djava.security.policy=C:/Program Files/glassfish-v2url/domains/domain1/config/server.policy
-Djava.util.logging.manager=com.sun.enterprise.server.logging.ServerLogManager
-Djavax.management.builder.initial=com.sun.enterprise.admin.server.core.jmx.AppServerMBeanServerBuilder
-Djavax.net.ssl.keyStore=C:/Program Files/glassfish-v2url/domains/domain1/config/keystore.jks
-Djavax.net.ssl.trustStore=C:/Program Files/glassfish-v2url/domains/domain1/config/cacerts.jks
-Djdbc.drivers=org.apache.derby.jdbc.ClientDriver
-Djmx.invoke.getters=true
-Dsun.rmi.dgc.client.gcInterval=3600000
```



Output

```
AplikacjaInternetowa1 (run) x Java DB Database Process x GlassFish V2 x
- Dsun.rmi.dgc.server.gcInterval=3600000
- client
-XX:+UnlockDiagnosticVMOptions
-XX:MaxPermSize=192m
-Xmx512m
-XX:NewRatio=2
-XX:+LogVMOutput
-XX:LogFile=C:/Program Files/glassfish-v2url/domains/domain1/logs/jvm.log
-cp
C:/Program Files/glassfish-v2url/lib/jhall.jar;C:\Program Files\glassfish-v2url\lib\appserv-launch.jar
com.sun.enterprise.server.PELaunch
start
Starting Sun Java System Application Server 9.1_01 (build b09d-fcs) ...
MBeanServer started: com.sun.enterprise.interceptor.DynamicInterceptor
CORE5098: AS Socket Service Initialization has been completed.
CORE5076: Using [Java HotSpot(TM) Client VM, Version 1.6.0_02] from [Sun Microsystems Inc.]
SEC1002: Security Manager is OFF.
C:/Program Files/glassfish-v2url/domains/domain1/config/._com_sun_appsrv_pid
ADM0001: SunoneInterceptor is now enabled
SEC1143: Loading policy provider com.sun.enterprise.security.provider.PolicyWrapper.
WEB0114: SSO is disabled in virtual server [server]
```



Output

AplikacjaInternetowa1 (run) x Java DB Database Process x GlassFish V2 x

SEC1002: Security Manager is OFF.

C:/Program Files/glassfish-v2url/domains/domain1/config/._com_sun_appserv_pid

ADM0001: SunoneInterceptor is now enabled

SEC1143: Loading policy provider com.sun.enterprise.security.provider.PolicyWrapper.

WEB0114: SSO is disabled in virtual server [server]

WEB0114: SSO is disabled in virtual server [_asadmin]

ADM1079: Initialization of AMX MBeans started

ADM1504: Here is the JMXServiceURL for the Standard JMXConnectorServer: [service:jmx:rmi:///jndi/rmi:///localhost

ADM1506: Status of Standard JMX Connector: Active = [true]

JBIFW0010: JBI framework ready to accept requests.

WEB0302: Starting Sun-Java-System/Application-Server.

WEB0712: Starting Sun-Java-System/Application-Server HTTP/1.1 on 8090

WEB0712: Starting Sun-Java-System/Application-Server HTTP/1.1 on 8182

WEB0712: Starting Sun-Java-System/Application-Server HTTP/1.1 on 4849

Initializing Sun's JavaServer Faces implementation (1.2_04-b20-p03) for context '/AplikacjaInternetowa2'

Initializing Sun's JavaServer Faces implementation (1.2_04-b20-p03) for context '/AplikacjaInternetowa1'

Initializing Sun's JavaServer Faces implementation (1.2_04-b20-p03) for context '/WebApl'

SMGT0007: Self Management Rules service is enabled

Application server startup complete.

deployed with moduleid = AplikacjaInternetowa1

Initializing Sun's JavaServer Faces implementation (1.2_04-b20-p03) for context '/AplikacjaInternetowa1'



Project and file tabs: Pro..., Files, Services, Start Page, Page1. Below are tabs for Design, JSP, and Java. A Data window is also visible on the right.

HTTP Monitor

Request Cookies Session Context Client and Server Headers

All Records

- Current Records
- GET AplikacjaInternetowa1 22:06 29.02.08]
- GET AplikacjaInternetowa1 22:06 29.02.08]
- GET suntheme.js 22:07 29.02.08]
- GET suntheme.js 22:07 29.02.08]
- GET suntheme.js 22:07 29.02.08]
- GET stylesheet.css 22:07 29.02.08]
- GET stylesheet.css 22:07 29.02.08]
- GET bg_gradient.gif 22:07 29.02.08]
- GET bg_gradient.gif 22:07 29.02.08]
- Saved Records

Request

Request URI	/AplikacjaInternetowa1/
Method	GET
Query string	
Protocol	HTTP/1.1
Client IP address	127.0.0.1
Scheme	http
HTTP exit status (as set by servlet)	Could not be determined

There was no query string

Request attributes before the request

com.sun.enterprise.http.sessionTracker	org.apache.coyote.tomcat5.SessionTracker@c56236
--	---

Request attributes after the request

javax.faces.webapp.PAGECONTEXT_COUNTER	1
com.sun.faces.extensions.avatar.LINKED/com.sun.faces.ajax.js	true
com.sun.faces.taglib.jsf_core.VIEWTAG_STACK	[]
com.sun.faces.renderKitImplForRequest	com.sun.faces.renderkit.RenderKitFactoryImpl@19e2380
com.sun.faces.managedBeanStack	[]
javax.servlet.jsp.jstl.fmt.locale.request	pl_PL
com.sun.faces.INVOCATION_PATH	/faces
javax.faces.webapp.GLOBAL_ID_VIEW	{html1=com.sun.webui.jsf.component.HtmlTaq@9a94b3, page1=com.sun....
com.sun.enterprise.http.sessionTracker	org.apache.coyote.tomcat5.SessionTracker@c56236
com.sun.faces.ExternalContextImpl	com.sun.faces.context.ExternalContextImpl@dcbf25
com.sun.faces.extensions.avatar.LINKED/prototype.js	true
com.sun.webui.jsf.theme.THEME	com.sun.webui.theme.ResourceBundleTheme@188f0f7
com.sun.faces.taglib.jsf_core.PRE_VIEW_LOADBUNDLES_LIST	[]

Output

AplikacjaInternetowa1 (run) x Java DB Database Process x GlassFish V2 x

```
Undeploying ...
While undeploying, trying to stop application in target server completed successfully
While undeploying, trying to remove reference for application in target server completed successfully
```





6) Zastąpienie elementu typu **TextField** elementem typu **DropDownList**

Należy wykonać kopię projektu za pomocą opcji **Copy** wybranej z wyskakującego menu, po kliknięciu prawym klawiszem myszy na nazwę projektu w zakładce **Project**.

1. Po przejściu do trybu **Design** dla strony **Page1** należy usunąć komponent typu **TextField** (kliknąć na elemencie prawym klawiszem myszy i wybrać z wyskakującego menu opcję **Delete**)

2. Przeciągnąć z Palety komponent typu **DropDownList** i położyć na miejscu usuniętego elementu typu **TextField**.

3. W oknie **Navigator** pojawił się komponent **dropDown1** oraz obiekt typu **dropDown1DefaultOptions** jako właściwość **items** komponentu typu **DropDownList**

4. Naciśnij **Ctrl-Shift-Drag** na komponencie **Label** i przeciągnij na komponent typu **DropDownList** w celu powiązania tych komponentów (pojawi się w okienku **Properties** dla komponentu typu **Label** właściwość **for** ustawiona na **dropDown1**)



Project Explorer showing the structure of 'AplikacjaInternetowa1' with folders like Web Pages, WEB-INF, resources, and Page1.jsp.

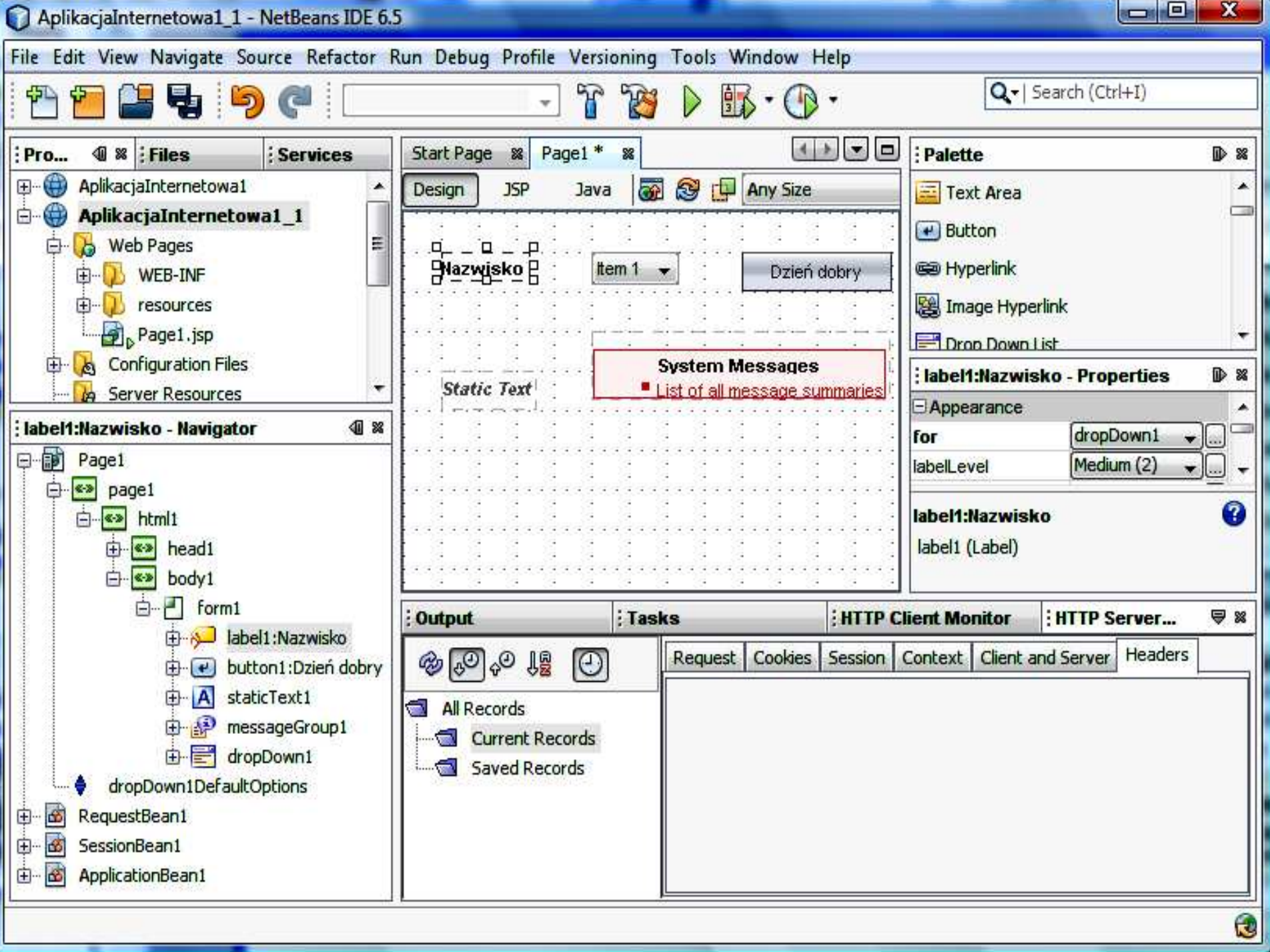
Design view of 'Page1' showing a text field labeled 'Podaj nazwisko' and a static text element. A context menu is open over the text field.

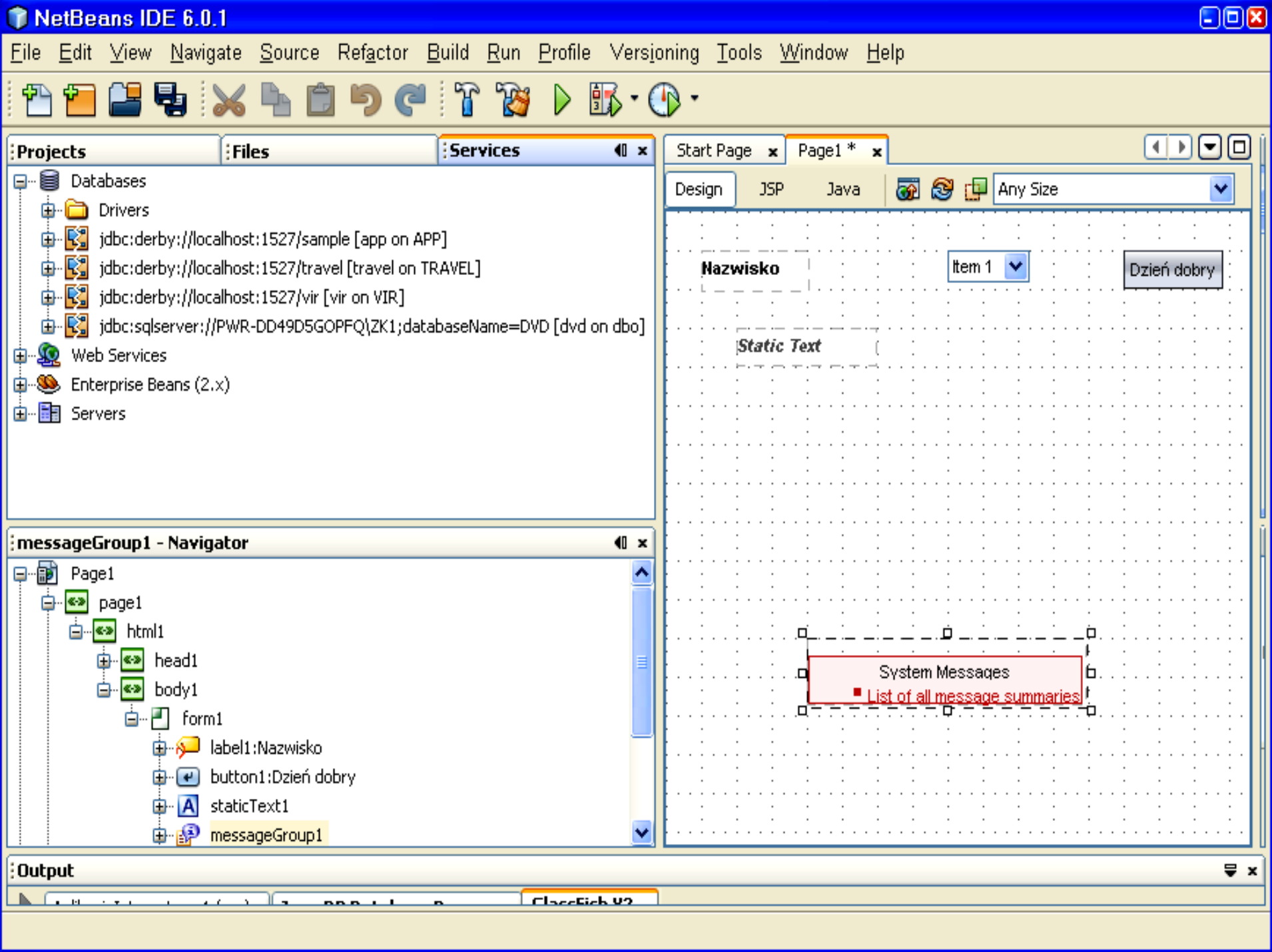
- Edit Java Source
- Auto-Submit on Change
- Bind to Data...
- Add
- Property Bindings...
- Edit JSP Source
- Edit Inline
- Select Parent
- Edit Event Handler
- Set Initial Focus
- Configure Virtual Forms...
- Snap to Grid
- Bring to Front
- Send to Back
- Customize
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Delete
- Preview in Browser...
- Refresh
- Target Browser Window
- Page Navigation

Structure view showing the DOM tree for 'Page1', including elements like page1, html1, head1, body1, form1, and textField1:Podaj nazwisko.

Output window showing log messages.

Property Inspector showing various properties for the selected 'textField1:Podaj nazwisko' component, such as style, text, and data.





7) Powiązania elementu typu DropDownList z tabelą bazy danych

1. W oknie **Services** należy wybrać **Databases** oraz bazę danych **Travel** i połączyć się z nią: należy kliknąć prawym klawiszem myszy, wybrać z wyskakującego menu **Connect**
2. Pojawi się okno logowania dla loginu **travel** i hasła **travel**, należy je wprowadzić do formularza logowania i zatwierdzić
3. Po połączeniu pojawią elementy bazy danych **Travel**: tabele, procedury wbudowane itp.
4. Przeciągnij tabelę **Person** na komponent typu **DropDownList** w trybie **Design**
5. W oknie **Navigator** pojawią się nowe elementy niewizualne: komponent **personDataProvider** jako element strony **Page1** do obsługi wybranej tabeli oraz właściwość **personRowSet**, która należy do **SessionBean1** i służy do obsługi zapytań SQL
6. Kliknij prawym klawiszem myszy komponent typu **DropDownList** i wybierz opcję **Bind to Data** z wyskakującego menu. Pojawi się okno dialogowe **Bind to Data**
7. Należy ustawić pole **Value** na **PERSON.NAME** (używane jako wartość domyślna odpowiadająca kolumnie wybranej tabeli) oraz pole **Display PERSON.NAME** do wyświetlania w liście rozwijanej komponentu **DropDownList** – zatwierdzić wybór klikając na przycisk **OK**



Projects | **Files** | **Services**

- Databases
 - Drivers
 - jdbc:derby://localhost:1527/sample [app on APP]
 - jdbc:derby://localhost:1527/travel [travel on TRAVEL]
 - jdbc:derby://localhost:1527/... [Connect...]
 - jdbc:derby://localhost:1527/... [Disconnect]
 - jdbc:sqlserver://PWR... [Execute Command...]
 - ... [Refresh]
 - ... [Delete] [Delete]
 - ... [Properties]
- Web Services
- Enterprise Beans (2.x)
- Servers

Start Page | Page1 * | Design | JSP | Java | Any Size

Nazwisko | Item 1 | Dzień dobry

Static Text

System Messages
List of all message summaries

messageGroup1 - Navigator

- Page1
 - page1
 - html
 - head1
 - body1
 - form1
 - label1:Nazwisko
 - button1:Dzień dobry
 - staticText1
 - messageGroup1
 - dropDown1
- dropDown1DefaultOptions
- RequestBean1
- SessionBean1

Output

Connect [Close]

Basic setting | Advanced

User name:

Password:

Remember password during this session

OK Cancel

Connect [Close]

Basic setting | Advanced

User name:

Password:

Remember password during this session

Connecting to database ...

OK Cancel



Projects

- Databases
 - Drivers
 - jdbc:derby://localhost:1527/sample [app on APP]
 - jdbc:derby://localhost:1527/travel [travel on TRAVEL]
 - Tables
 - CARRENTAL
 - FLIGHT
 - HOTEL
 - PERSON
 - TRIP
 - TRIPTYPE
 - VALIDATION_TABLE
 - Views
 - Procedures
 - jdbc:derby://localhost:1527/vir [vir on VIR]
 - jdbc:sqlserver://PWR-DD49D5G0PFQ\ZK1;databaseName=DVD [dvd on dbo]
- Web Services
- Enterprise Beans (2.x)
- Servers

Start Page x Page1 * x

Design JSP Java Any Size

Nazwisko Item 1

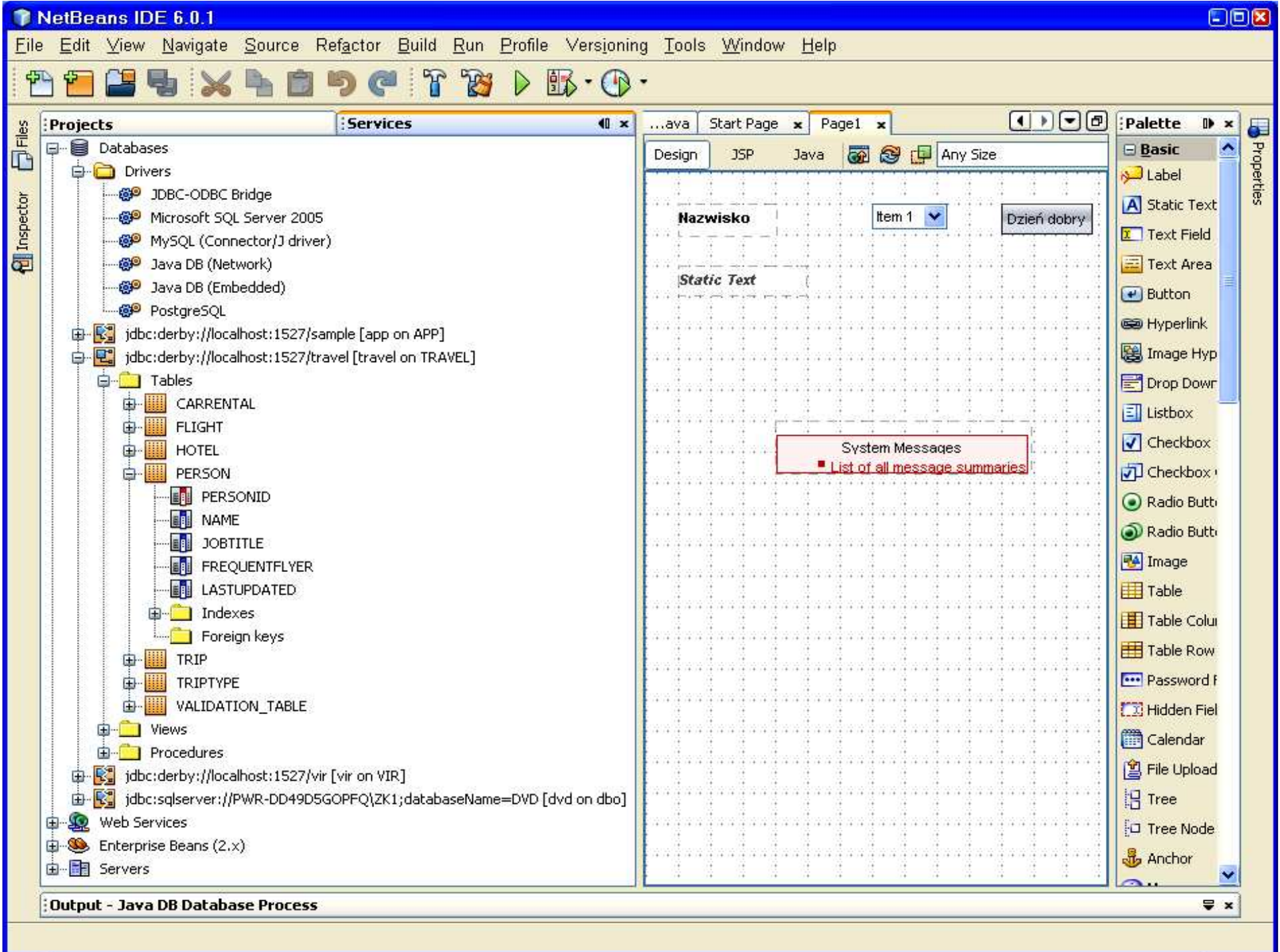
Static Text

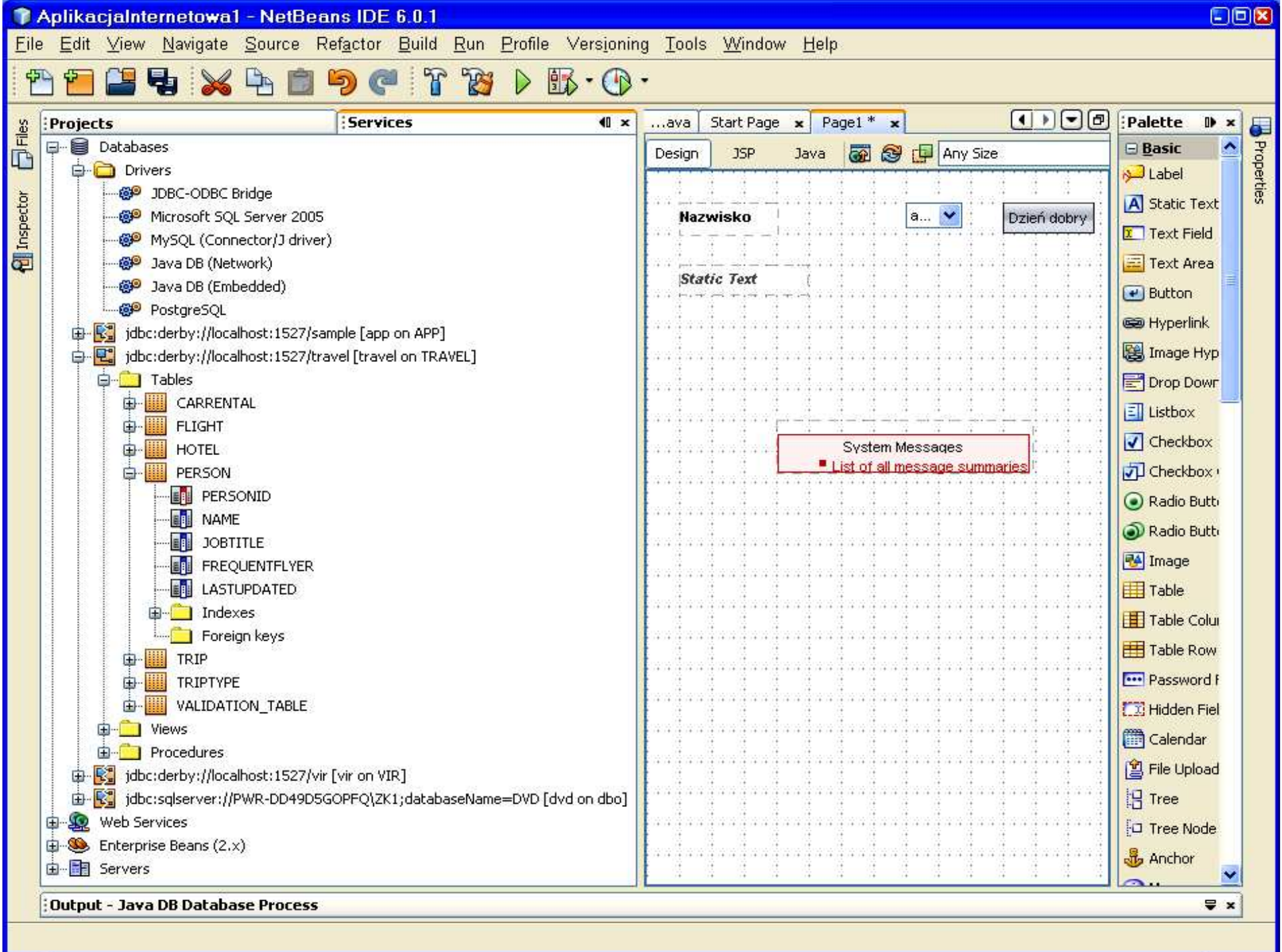
System Messages
List of all message summaries

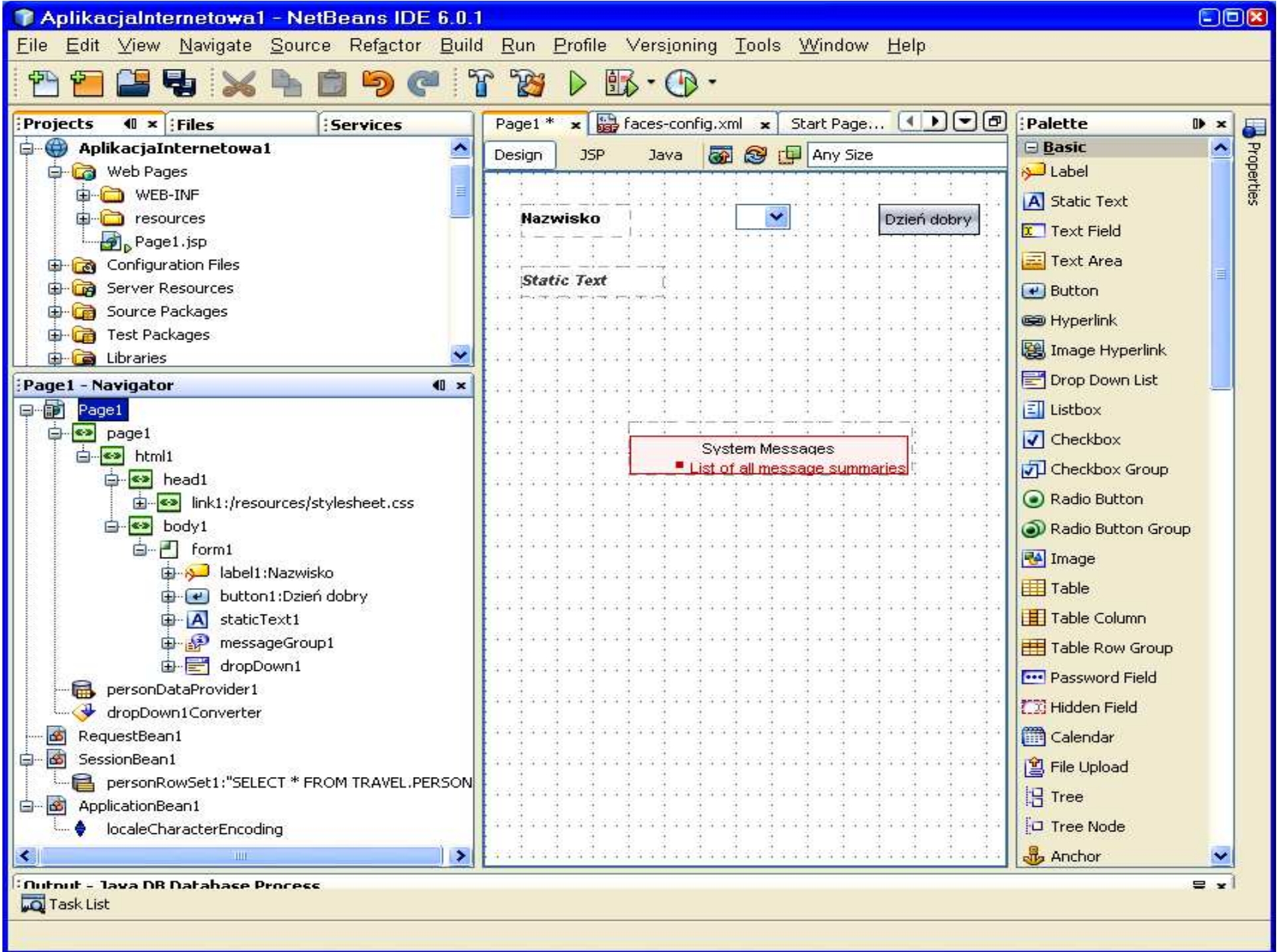
messageGroup1 - Navigator

- Page1
 - page1
 - html1
 - head1
 - body1
 - form1
 - label1:Nazwisko
 - button1:Dzień dobry
 - staticText1
 - messageGroup1
 - dropDown1

Output









Projects | **Files** | **Services**

- AplikacjaInternetowa1
 - Web Pages
 - WEB-INF
 - resources
 - Page1.jsp
 - Configuration Files
 - Server Resources
 - Source Packages
 - Test Packages
 - Libraries

dropDown1 - Navigator

- Page1
 - page1
 - html1
 - head1
 - link1:/resources/stylesheet.css
 - body1
 - form1
 - label1:Nazwisko
 - button1:Dzień dobry
 - staticText1
 - messageGroup1
 - dropDown1
 - personDataProvider1
 - dropDown1Converter
 - RequestBean1
 - SessionBean1
 - personRowSet1:"SELECT * FROM TRAVEL.PERSON"
 - ApplicationBean1
 - localeCharacterEncoding

Page1 * x faces-config.xml x

Design JSP Java

Nazwisko

Static Text

System M

List of all m

- Edit Java Source
- Auto-Submit on Change
- Bind to Data...**
- Add
- Property Bindings...
- Edit JSP Source
- Edit Label
- Select Parent
- Edit Event Handler
- Set Initial Focus
- Configure Virtual Forms...
- Snap to Grid
- Bring to Front
- Send to Back
- Customize
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Delete
- Preview in Browser...
- Refresh
- Target Browser Window
- Page Navigation

Palette

- Basic
 - Label
 - Static Text
 - Text Field
 - Text Area
 - Button
 - Hyperlink
 - Image Hyperlink
 - Drop Down List
 - Listbox
 - Checkbox
 - Checkbox Group
 - Radio Button
 - Radio Button Group
 - Image
 - Table
 - Table Column
 - Table Row Group
 - Password Field
 - Hidden Field
 - Calendar
 - File Upload
 - Tree
 - Tree Node
 - Anchor

Properties

Output - Java DB Database Process

Task List

Bind to Data - dropDown1

Current Items property setting

```
#{Page1.personDataProvider1.options['PERSON.NAME,PERSON.NAME']}
```

Bind to Data Provider

Bind to an Object

Choose a Data Provider to bind to dropDown1:

personDataProvider1 (Page1)

Value field:

<none>
PERSON.PERSONID Integer
PERSON.NAME String
PERSON.JOBTITLE String
PERSON.FREQUENTFLYER Integer
PERSON.LASTUPDATED Timestamp

Display field:

<use value field>
PERSON.PERSONID Integer
PERSON.NAME String
PERSON.JOBTITLE String
PERSON.FREQUENTFLYER Integer
PERSON.LASTUPDATED Timestamp

OK

Cancel

Apply

Help

8) Dodanie nowych właściwości elementom strony

1. Obsługa zdarzenia kliknięcia na komponent typu **Button** – w trybie **Design** należy dwukrotnie kliknąć na przycisk Pojawi się wtedy plik typu Java dla strony **Page1** z klasą dziedziczącą po klasie **AbstractPageBean** na metodzie **button1_action()**.

Ciało metody button1_action():

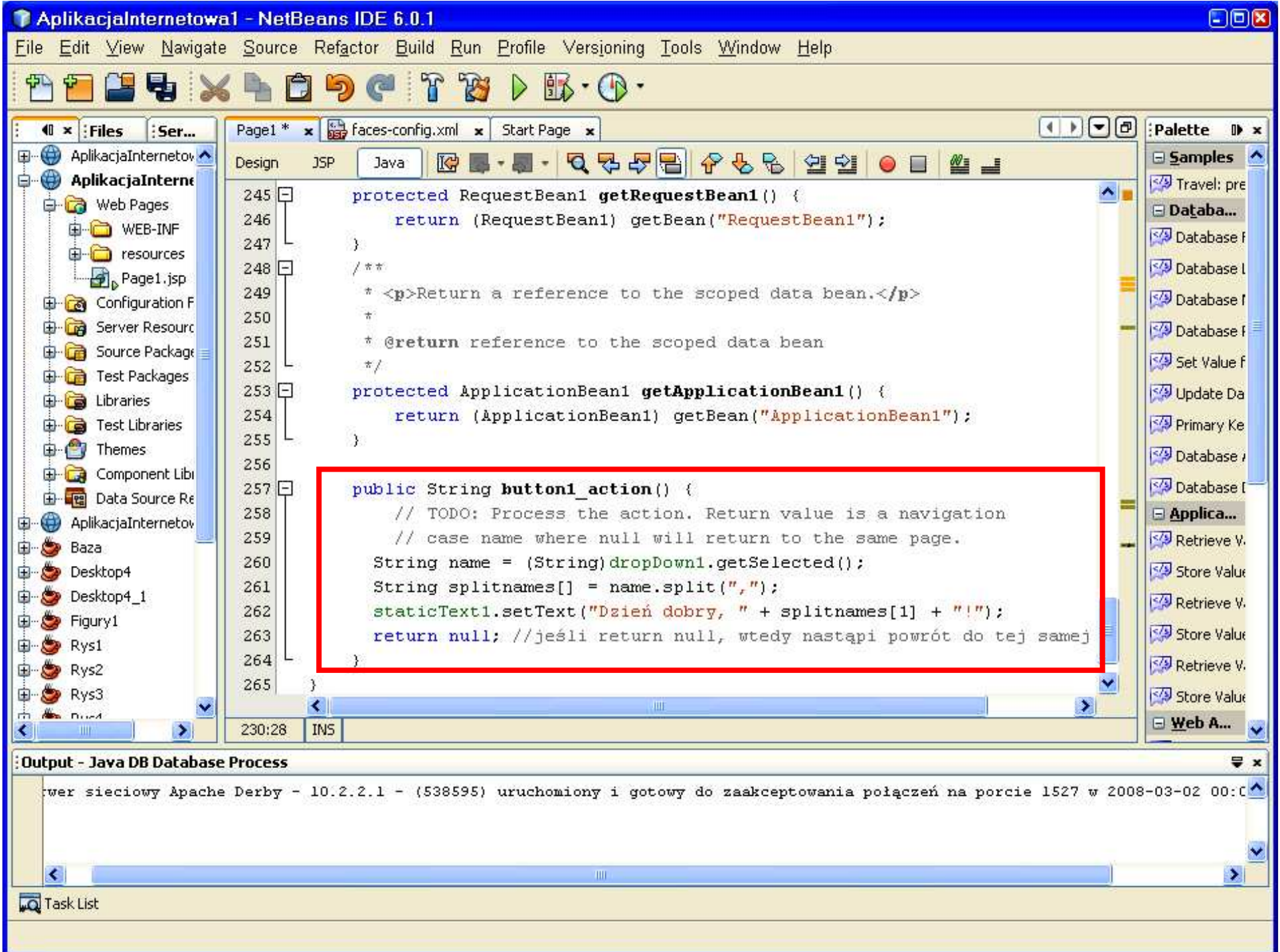
```
public String button1_action() {  
    String name = (String)dropDown1.getSelected();  
    String splitnames[] = name.split(",");  
    staticText1.setText("Dzień dobry, " + splitnames[1] + "!");  
    return null; //jeśli return null, wtedy nastąpi powrót do tej samej strony  
}
```

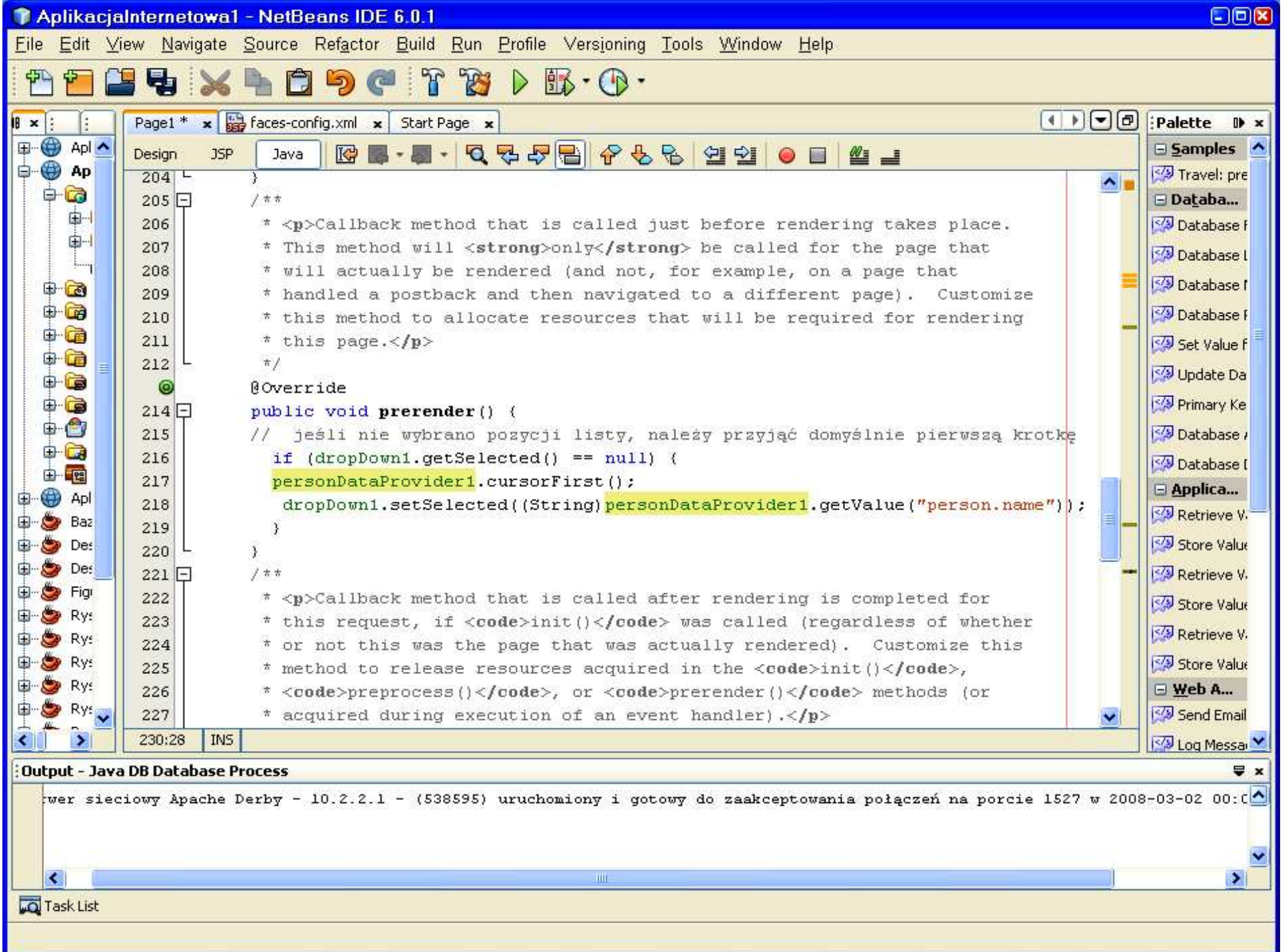
Kod ten pozwala na wyświetlenie danej wybranej z listy komponentu typu **DropDownList** w komponencie typu **StaticText**. Dane są pobrane z bazy danych.

2. Zdefiniowanie metody **prerender** pozwala w przypadku braku wybranej pozycji wyświetlać domyślną pierwszą daną występującą w bazie danych na pozycji pierwszej. Dane są wyświetlane w liście komponentu **DropDownList** jako nazwisko i imię, stąd w polu **StaticText** wyświetla się imię, jako łańcuch po przecinku, wyświetlany w liście komponentu **DropDownList**, wydobyty funkcją metodą **split** (z klasy typu **String**).

Ciało metody prerender

```
public void prerender() {  
    // jeśli nie wybrano pozycji listy, należy przyjąć domyślnie pierwszą krotkę  
    if (dropDown1.getSelected() == null) {  
        personDataProvider.cursorFirst();  
        dropDown1.setSelected((String)personDataProvider.getValue("person.name"));  
    }  
}
```





9) Uruchomienie aplikacji – podobnie jak w p.5

