

# Tworzenie warstwy prezentacji –drugi etap

## Przykład z laboratorium5\_6

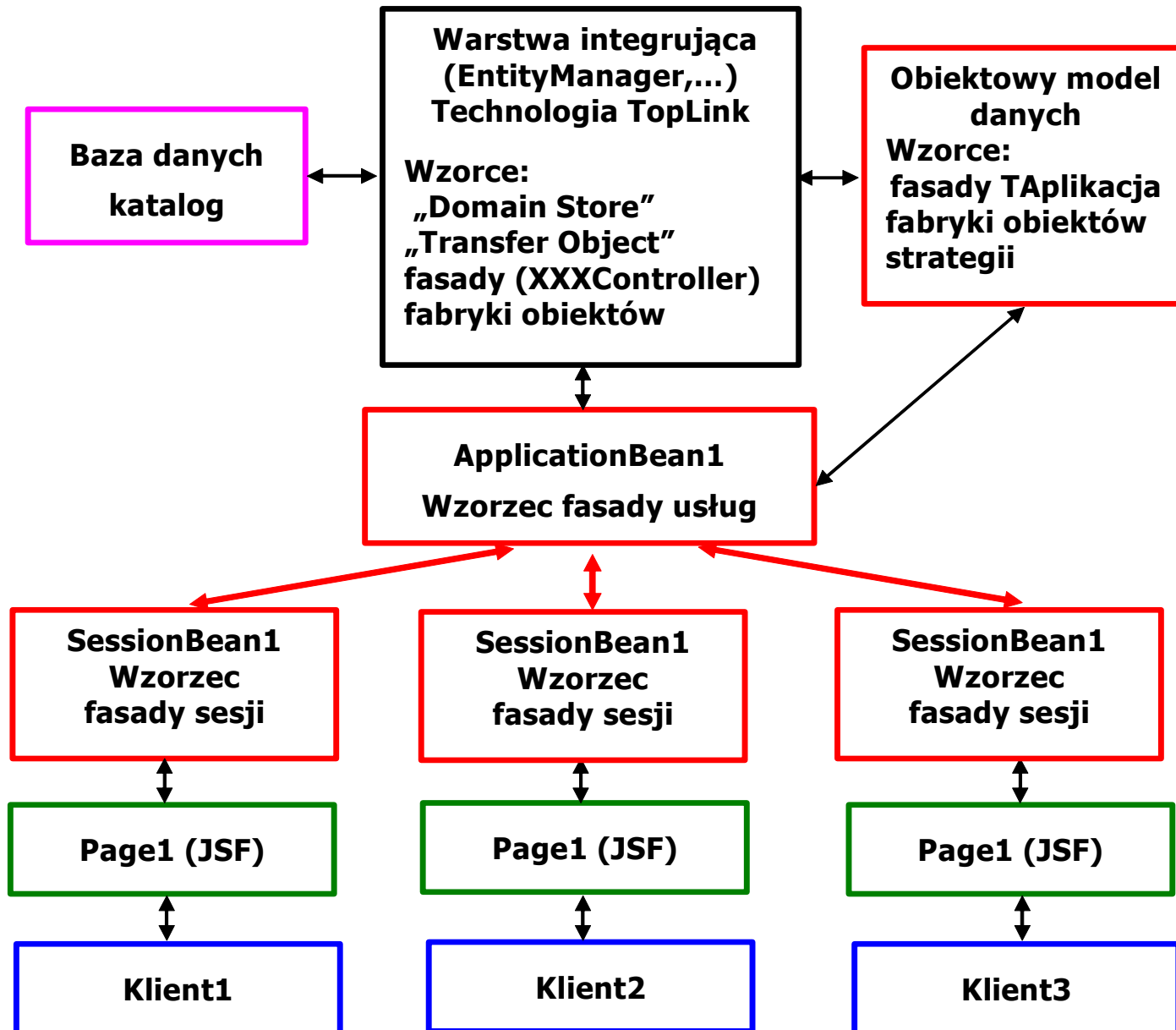
Autor

Zofia Kruczkiewicz

Wzorce oprogramowania –  
laboratorium7\_8

- 1. Wykonanie formularzy typu JSP zawierających wieloużywalne formularze typu JSPF dla aplikacji przeznaczonej dla wielu klientów ze wspólną warstwą biznesową istniejącą podczas sesji i wspólną warstwą integrującą z bazą danych, Przykład projektu składającego się z wielu formularzy opartych na fragmentach stron internetowych typu JSPF (wg wykładu 5)**

# Architektura aplikacji pięciowarstwowej



# Projekty formularza głównego „Strona główna” (Page1.jsp)

The screenshot displays an IDE interface for a web application project named "WebWypożyczalnia2". The main workspace shows the design view of "Page1.jsp" in the "Design" mode. The design view contains a header image of a boat on water and a list of menu items:

- [Strona główna](#)
- [Dodaj tytuły w aplikacji](#)
- [Dodaj książki w aplikacji](#)
- [Dodaj tytuł do bazy](#)
- Dodaj książkę do bazy
- [Przepisz tytuły do bazy](#)
- [Przepisz książki do bazy](#)

The left sidebar shows the project structure:

- WebWypożyczalnia2
  - Web Pages
    - WEB-INF
    - resources
    - Baza\_książki.jsp
    - Baza\_tytul.jsp
    - Baza\_tytuly.jsp
    - FormKsiążka.jspf
    - FormTytul.jspf
    - Książki.jsp
    - Książkiaplikacja.jspf
    - Książkibaza.jspf
    - Logo.jspf
    - Menu.jspf
    - Page1.jsp
    - Tytuly.jsp
    - Tytulyaplikacja.jspf
    - Tytulybaza.jspf

The bottom-left pane shows the "jsp:directive.include:Menu.jspf - Navigator" view, displaying the structure of the included menu file:


```
Page1
├── page1
│   ├── html1
│   │   ├── head1
│   │   └── body1
│   │       └── form1
│   │           ├── div
│   │           └── div
│   └── jsp:directive.include:Menu.jspf
```

The bottom status bar shows "HTTP Monitor".

http://localhost:8080/WebWypożyczalnia2/faces/Page1.jsp - Windows Internet Explorer

http://localhost:8080/WebWyp

Norton™ Monitorowanie fałszywych witryn jest włączono... Opcje



## Przykład wielowarstwowej aplikacji

- [Strona główna](#)
- [Dodaj tytuły w aplikacji](#)
- [Dodaj książki w aplikacji](#)
- [Dodaj tytuł do bazy](#)
- [Dodaj książkę do bazy](#)
- [Przepisz tytuły do bazy](#)
- [Przepisz książki do bazy](#)

Internet | Tryb chroniony: włączony 100%

# Projekty formularza „Dodaj tytuły w aplikacji” (Tytuly.jsp)

The screenshot displays an IDE interface for a web application project named "WebWypożyczalnia2". The main workspace shows a design view of a JSP page titled "Tytuly.jsp". The page content includes a navigation menu with links such as "Strona główna", "Dodaj tytuły w aplikacji", "Dodaj książki w aplikacji", "Dodaj tytuł do bazy", "Dodaj książkę do bazy", "Przepisz tytuły do bazy", and "Przepisz książki do bazy". A form for adding titles is visible, with fields for "Tytuł", "Autor", "ISBN", "Wydawnictwo", and "Aktor", and a "Dodaj tytuł" button. A "System Messages" box displays a message: "List of all message summaries".

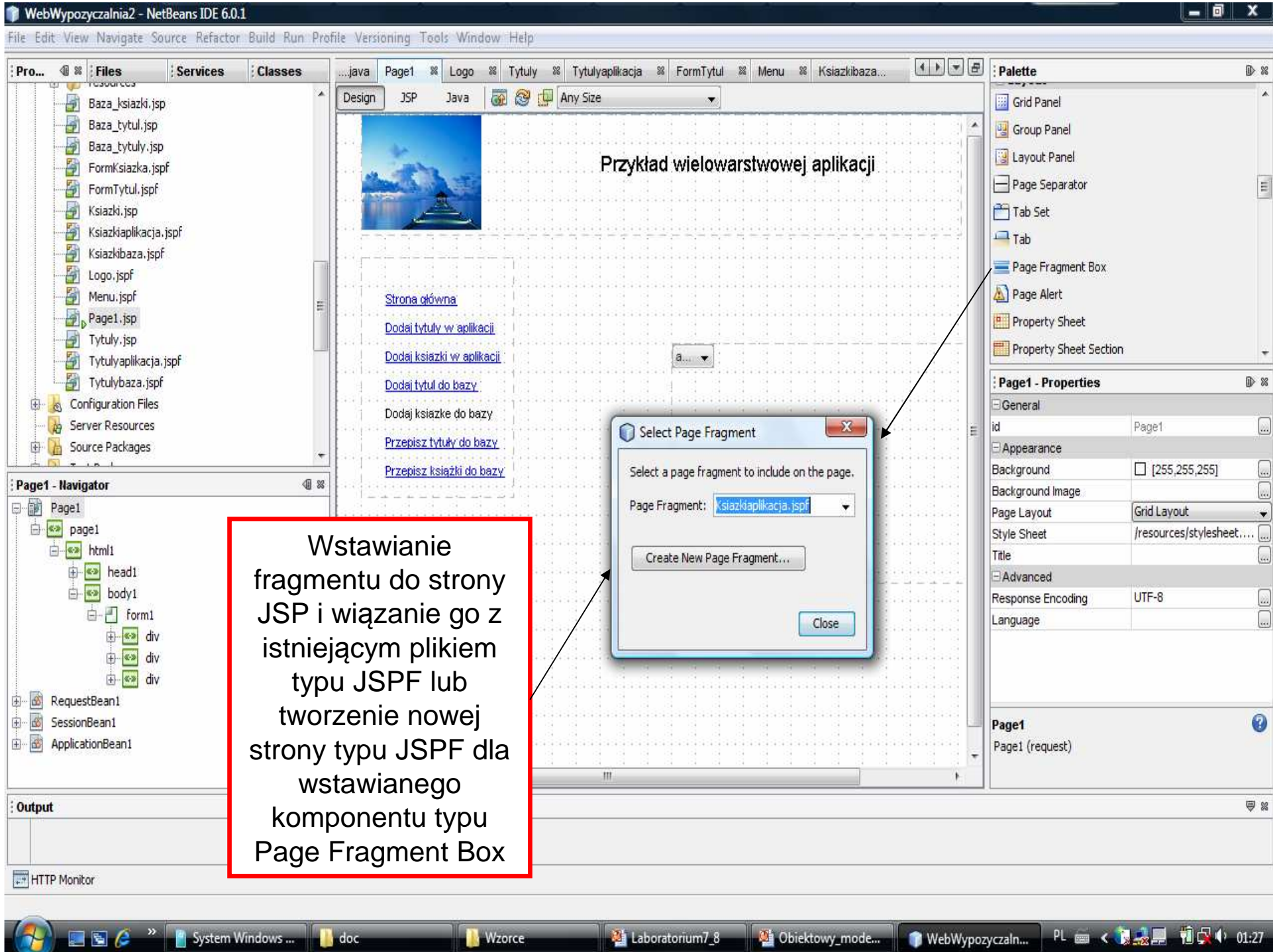
The left sidebar shows the project structure under "Web Pages":

- Web Pages
  - WEB-INF
  - resources
  - Baza\_książki.jsp
  - Baza\_tytul.jsp
  - Baza\_tytuly.jsp
  - FormKsiążka.jspf
  - FormTytul.jspf
  - Książki.jsp
  - Książkiaplikacja.jspf
  - Książkibaza.jspf
  - Logo.jspf
  - Menu.jspf
  - Page1.jsp
  - Tytuly.jsp
  - Tytulyaplikacja.jspf
  - Tytulybaza.jspf

The "Tytuly - Navigator" window shows the following structure:

- Tytuly
  - page1
    - html1
      - head1
      - body1
        - form1
          - div
          - div
          - div
          - messageGroup1
          - div
          - addTitle:Dodaj tytuł

At the bottom, the "HTTP Monitor" window is visible.




Wstawianie fragmentu do strony JSP i wiązanie go z istniejącym plikiem typu JSPF lub tworzenie nowej strony typu JSPF dla wstawianego komponentu typu Page Fragment Box

http://localhost:8080/WebWypożyczalnia2/faces/Tytuly.jsp - Windows Internet Explorer

http://localhost:8080/WebWyp

Wyszukaj

Norton™ Monitorowanie fałszywych witryn jest włączono... Opcje



## Przykład wielowarstwowej aplikacji

- Strona główna
- Dodaj tytuły w aplikacji
- Dodaj książki w aplikacji
- Dodaj tytuł do bazy
- Dodaj książkę do bazy
- Przepisz tytuły do bazy
- Przepisz książki do bazy

<b>Tytuł</b>	<input type="text"/>
<b>Autor</b>	<input type="text"/>
<b>ISBN</b>	<input type="text"/>
<b>Wydawnictwo</b>	<input type="text"/>
<b>Aktor</b>	<input type="text"/>

**Dodaj tytuł**

Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1

**Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1**

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2

Tytuł: 3 Autor: 3 ISBN: 3 Wydawnictwo: 3

Tytuł: 3 Autor: 3 ISBN: 3 Wydawnictwo: 3 Aktor: 3

Tytuł: 4 Autor: 4 ISBN: 4 Wydawnictwo: 4

Tytuł: 5 Autor: 5 ISBN: 5 Wydawnictwo: 5 Aktor: 5

Tytuł: 6 Autor: 6 ISBN: 6 Wydawnictwo: 6

Internet | Tryb chroniony: włączony 100%



# Projekty formularza „Dodaj książki w aplikacji” (Ksiazki.jsp)

The screenshot displays an IDE interface for a web application project named "WebWypożyczalnia2". The main workspace shows a design view of a JSP page titled "Przykład wielowarstwowej aplikacji". The design is organized into several layers, indicated by dashed blue boxes:

- Top Layer:** Contains a header image of a boat on water and the text "Przykład wielowarstwowej aplikacji".
- Form Layer:** A central form with two input fields labeled "Numer" and "Termin", a dropdown menu, and a "Dodaj książkę" button.
- Navigation Layer:** A list of links: "Strona główna", "Dodaj tytuły w aplikacji", "Dodaj książki w aplikacji", "Dodaj tytuł do bazy", "Dodaj książkę do bazy", "Przepisz tytuły do bazy", and "Przepisz książki do bazy".
- Bottom Layer:** A large empty rectangular area at the bottom of the design.


On the left, the "Files" pane shows the project structure, including files like "FormKsiazka.jspf" and "Ksiazki.jsp". The "Ksiazki - Navigator" pane shows the DOM tree for the "page1" element, highlighting the "Form1" and "Dodaj książkę" button.

http://localhost:8080/WebWypożyczalnia2/faces/Ksiazki.jsp - Windows Internet Explorer

http://localhost:8080/WebWypożyczalnia2/faces/Ksiazki.j... Live Search

http://localhost:8080/WebWypożyczalni... Strona Narzędzia

Norton Monitorowanie fałszywych witryn jest włączone Opcje



## Przykład wielowarstwowej aplikacji

Strona główna

Dodaj tytuły w aplikacji

Dodaj książki w aplikacji

Dodaj tytuł do bazy

Dodaj książkę do bazy

Przepisz tytuły do bazy

Przepisz książki do bazy

Numer

Termin

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2 ▼

**Dodaj książkę**

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2 Numer: 2

**Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2 Numer: 2**

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2 Numer: 1 termin: Tue May 06 00:00:00 CEST 2008

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2 Numer: 3

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2 Numer: 4

Gotowe Internet | Tryb chroniony: włączony 100%

# Projekty formularza „Dodaj tytuł do bazy” (Baza\_tytul.jsp)

The screenshot displays an IDE interface for a web application project named "WebWypożyczalnia2". The main design area shows a multi-layered application layout titled "Przykład wielowarstwowej aplikacji".

**Navigation Menu (Left):**

- [Strona główna](#)
- [Dodaj tytuły w aplikacji](#)
- [Dodaj książki w aplikacji](#)
- [Dodaj tytuł do bazy](#)
- Dodaj książkę do bazy
- [Przepisz tytuły do bazy](#)
- [Przepisz książki do bazy](#)

**Form (Center):**

Form fields for adding a title:

- Tytuł
- Autor
- ISBN
- Wydawnictwo
- Aktor

**System Messages (Right):**

- System Messages
- List of all message summaries
- Zapisz tytuł do bazy

**Table (Bottom):**

Tytuły						
id	tytuł	autor	ISBN	wydawnictwo	aktor	
123	abc	abc	abc	abc	abc	
123	abc	abc	abc	abc	abc	
123	abc	abc	abc	abc	abc	

**File Explorer (Top Left):**

- Web Pages
  - WEB-INF
  - resources
  - Baza\_książki.jsp
  - Baza\_tytul.jsp
  - Baza\_tytuly.jsp
  - FormKsiążka.jspf
  - FormTytul.jspf
  - Książki.jsp
  - Książkiaplikacja.jspf
  - Książki baza.jspf
  - Logo.jspf
  - Menu.jspf
  - Page1.jsp
  - Tytuly.jsp
  - Tytulyaplikacja.jspf
  - Tytulybaza.jspf

**Baza\_tytul - Navigator (Bottom Left):**

- Baza\_tytul
  - page1
    - html1
      - head1
      - body1
        - div
        - div
        - div
        - div
        - bazatytul:Zapisz tytuł do bazy
        - messageGroup1

**HTTP Monitor (Bottom):**



### Przykład wielowarstwowej aplikacji

- Strona główna
- Dodaj tytuły w aplikacji
- Dodaj książki w aplikacji
- Dodaj tytuł do bazy
- Dodaj książki do bazy
- Przepisz tytuły do bazy
- Przepisz książki do bazy

**Tytuł**

**Autor**

**ISBN**

**Wydawnictwo**

**Aktor**

Zapisz tytuł do bazy

Tytuły					
id ↑↓	tytuł ↑↓	autor ↑↓	ISBN ↑↓	wydawnictwo ↑↓	aktor ↑↓
2	1	1	1	1	
4	2	2	2	2	2
152	2	2	2	2	
252	3	3	3	3	
352	3	3	3	3	3
353	4	4	4	4	
452	5	5	5	5	5
453	6	6	6	6	

# Projekty formularza „Przepisz tytuły do bazy” (Baza\_tytuly.jsp)

The screenshot shows an IDE window for a web application project named "WebWypożyczalnia2". The design view displays a page titled "Przykład wielowarstwowej aplikacji". The page layout includes a menu of links, a button labeled "Zapisz tytuły do bazy", and a table of titles.

**Menu Links:**

- [Strona główna](#)
- [Dodaj tytuły w aplikacji](#)
- [Dodaj książki w aplikacji](#)
- [Dodaj tytuł do bazy](#)
- [Przepisz tytuły do bazy](#)
- [Przepisz książki do bazy](#)

**Table: Tytuły**

id	tytuł	autor	ISBN	wydawnictwo	aktor
123	abc	abc	abc	abc	abc
123	abc	abc	abc	abc	abc
123	abc	abc	abc	abc	abc

**Project Structure (Files):**

- Web Pages
  - WEB-INF
  - resources
  - Baza\_książki.jsp
  - Baza\_tytul.jsp
  - Baza\_tytuly.jsp
  - FormKsiążka.jspf
  - FormTytul.jspf
  - Książki.jsp
  - Książkiaplikacja.jspf
  - Książkibaza.jspf
  - Logo.jspf
  - Menu.jspf
  - Page1.jsp
  - Tytuly.jsp
  - Tytulyaplikacja.jspf
  - Tytulybaza.jspf

**Baza\_tytuly - Navigator:**

- Baza\_tytuly
  - page1
    - html1
      - head1
      - body1
        - form1
          - div
          - div
          - div
          - jsp:directive.include:Tytylyba
          - dodajtytulbaza:Zapisz tytuły do b

**RequestBean1:**

- RequestBean1



## Przykład wielowarstwowej aplikacji

Zapisz tytuły do bazy

- Strona główna
- Dodaj tytuły w aplikacji
- Dodaj książki w aplikacji
- Dodaj tytuł do bazy
- Dodaj książkę do bazy
- Przepisz tytuły do bazy
- Przepisz książki do bazy

Tytuły					
id ↑	tytuł ↑	autor ↑	ISBN ↑	wydawnictwo ↑	aktor ↑
2	1	1	1	1	
4	2	2	2	2	2
152	2	2	2	2	
252	3	3	3	3	
352	3	3	3	3	3
353	4	4	4	4	
452	5	5	5	5	5
453	6	6	6	6	

# Projekty formularza „Przepisz książki do bazy” (Baza\_książki.jsp)

The screenshot shows an IDE interface with the following components:

- Project Explorer (Files):** Lists files for 'WebWypożyczalnia2', including 'Baza\_książki.jsp', 'Baza\_tytul.jsp', 'FormKsiążka.jsp', and others.
- Design View:** Displays a page titled 'Przykład wielowarstwowej aplikacji'. It features a header image, a navigation menu, a 'Zapisz książki do bazy' button, and a table.
- Navigator:** Shows the DOM tree for 'Baza\_książki', with 'dodajksiazkabaza:Zapisz książki do bazy' selected.

**Navigation Menu:**

- [Strona główna](#)
- [Dodaj tytuły w aplikacji](#)
- [Dodaj książki w aplikacji](#)
- [Dodaj tytuł do bazy](#)
- Dodaj książkę do bazy
- [Przepisz tytuły do bazy](#)
- [Przepisz książki do bazy](#)

**Table:**

id	numer	termin	tytuł książki
123	123	Thu May 08 16:21:40 CEST 2008	Tytuł: abc Autor: abc ISBN: abc Wydawnictwo: abc
123	123	Thu May 08 16:21:40 CEST 2008	Tytuł: abc Autor: abc ISBN: abc Wydawnictwo: abc
123	123	Thu May 08 16:21:40 CEST 2008	Tytuł: abc Autor: abc ISBN: abc Wydawnictwo: abc



## Przykład wielowarstwowej aplikacji

Zapisz książki do bazy

- [Strona główna](#)
- [Dodaj tytuły w aplikacji](#)
- [Dodaj książki w aplikacji](#)
- [Dodaj tytuł do bazy](#)
- [Dodaj książkę do bazy](#)
- [Przepisz tytuły do bazy](#)
- [Przepisz książki do bazy](#)

Table			
id	numer	termin	tytuł_książki
3	1	Tue May 06 00:00:00 CEST 2008	Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1
52	2		Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2
102	1	Tue May 06 00:00:00 CEST 2008	Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2
153	1		Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2
202	3		Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2
302	4		Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2
402	1	Fri May 09 00:00:00 CEST 2008	Tytuł: 4 Autor: 4 ISBN: 4 Wydawnictwo: 4

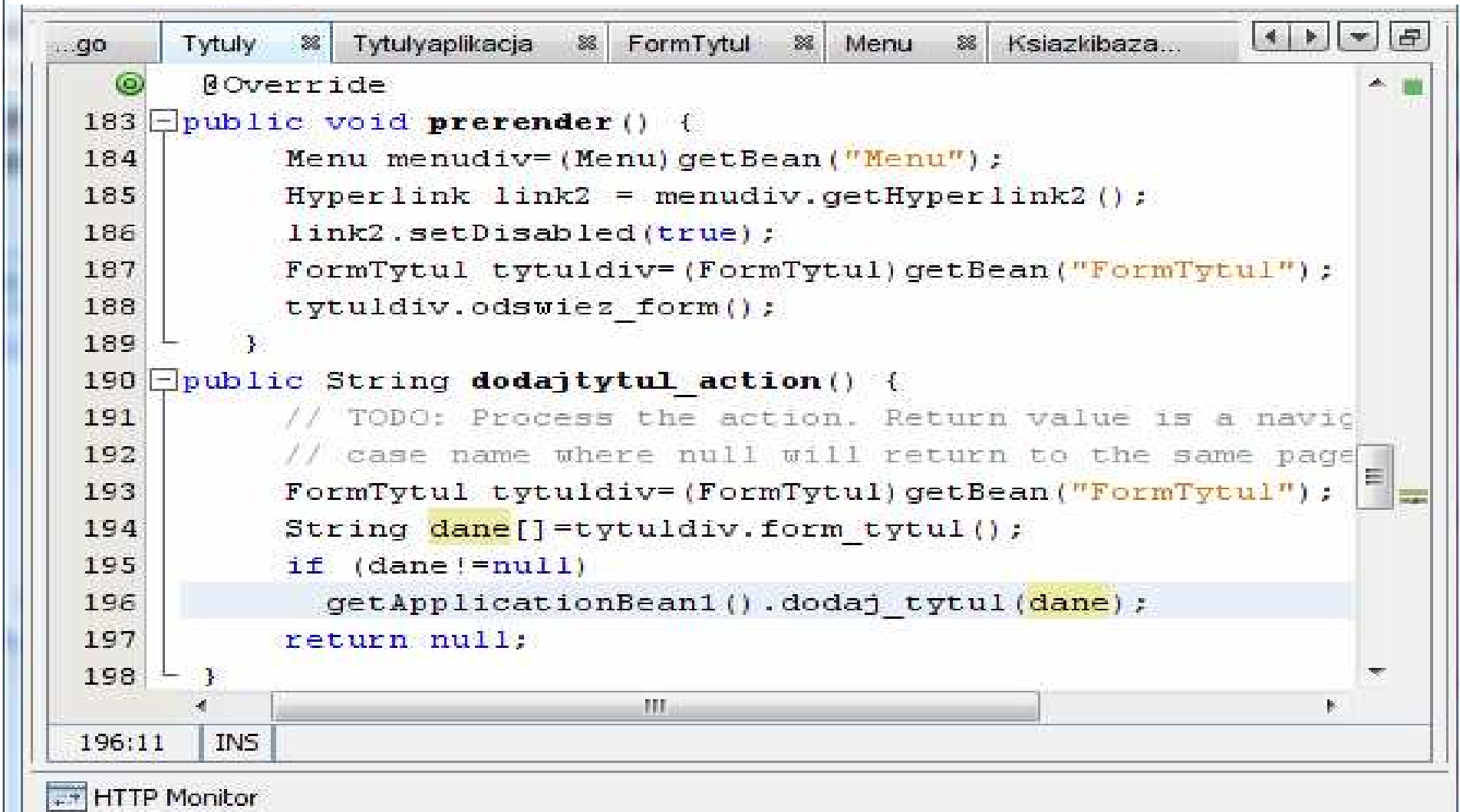


**2. Wykonanie oprogramowania systemu dla wielu klientów ze wspólną warstwą biznesową istniejącą podczas sesji i wspólną warstwą integrującą z bazą danych,**

**Przykład projektu składającego się z wielu formularzy opartych na fragmentach stron internetowych typu JSPF**

## 2.1. Oprogramowanie dotyczące formularza Tytuly.jsp

Definicje metod w klasie Tytuly dla strony typu JSP – do wstawiania nowego tytułu do warstwy biznesowej (obsługa zdarzenia dodajtytul\_action) oraz generowania widoku w fazie Response przetwarzania strony metodą prerender (wygaszanie linku do bieżącej strony w formularzu Menu typu JSPF i czyszczenie pól formularza FormTytul typu JSPF jego metodą odswiez\_form)



```
@Override
183 public void prerender() {
184     Menu menudiv=(Menu)getBean("Menu");
185     Hyperlink link2 = menudiv.getHyperlink2();
186     link2.setDisabled(true);
187     FormTytul tytuldiv=(FormTytul)getBean("FormTytul");
188     tytuldiv.odswiez_form();
189 }
190 public String dodajtytul_action() {
191     // TODO: Process the action. Return value is a navig
192     // case name where null will return to the same page
193     FormTytul tytuldiv=(FormTytul)getBean("FormTytul");
194     String dane[]=tytuldiv.form_tytul();
195     if (dane!=null)
196         getApplicationBean1().dodaj_tytul(dane);
197     return null;
198 }
```

196:11 INS

HTTP Monitor

## Definicje metod w klasie FormTytul typu BackingBean dla strony typu JSPF –

- do pobierania danych o nowym tytule (form\_tytul): dane dla wstawianych tytułów książek
- do czyszczenia pól formularza (odswiez\_form)

```
211 public void odswiez_form()
212 {   tytulYaplikacjaPanel.setRendered(true);
213     tytul.setText("");
214     autor.setText("");
215     ISBN.setText("");
216     wydawnictwo.setText("");
217     aktor.setText("");
218 }
219 public String [] form_tytul()
220 {
221     String jaki;
222     if ((autor.getText().equals("") || tytul.getText().equals("") ||
223         ISBN.getText().equals("") || wydawnictwo.getText().equals(""))
224         return null;
225     if (aktor.getText().equals("")) {
226         jaki = "1";
227     } else {
228         jaki = "3";
229     }
230     String dane[] = {jaki, (String) autor.getText(),
231                     (String) tytul.getText(), (String) ISBN.getText(),
232                     (String) wydawnictwo.getText(), (String) aktor.getText()};
233     return dane;
234 }
```

212:6 INS

HTTP Monitor

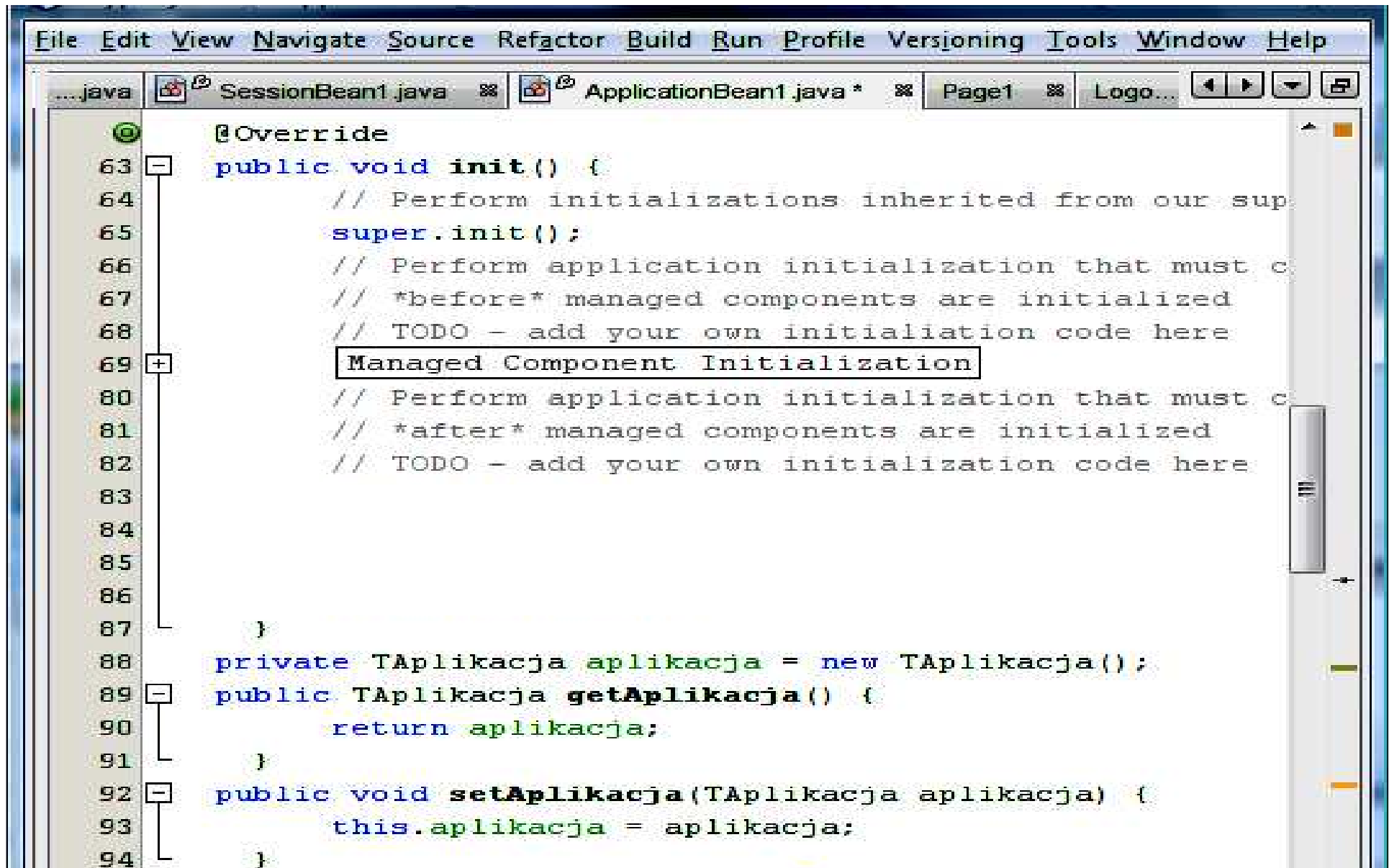
Save All finished.

Definicje metod w klasie ApplicationBean1 związanych z zapisem (dodaj\_tytul) i odczytem (przygotujtytuly) danych typu kolekcja obiektów TTytul\_książki i TTytul\_książki\_na\_kasce w warstwie biznesowej – odczytane dane wstawiane są do tablicy tytul\_, która jest wyświetlana w komponencie typu DropDown List na stronie Tytulyaplikacja typu JSPF

```
...java  SessionBean1.java  ApplicationBean1.java  Page1  Logo  Tytuly...
135  public void  dodaj_tytul(String dane[])
136  {
137      {getAplikacja().dodaj_tytul(dane); //przypadek użycia "dodaj tytul
138      przygotujtytuly(); //wyswietlenie kolek
139  }
139  private Option tytul_[] = new Option[0];
140  public Option[] getTytuly_() {
141      return tytul_;
142  }
143  public void setTytuly_(Option[] tytul_) {
144      this.tytuly_ = tytul_;
145  }
146  public void przygotujtytuly() {
147      ArrayList<TTytul_książki> tytul_ = aplikacja.getTytul_książki();
148      int ile = tytul_.size();
149      if (ile > 0) {
150          Option pom[] = new Option[ile];
151          Iterator iterator = tytul_.iterator();
152          int i = 0;
153          while (iterator.hasNext()) {
154              pom[i++] =
155                  new Option(Integer.toString(i), iterator.next().toString());
156          }
157          tytul_ = pom;
158      }
159  }
```

HTTP Monitor

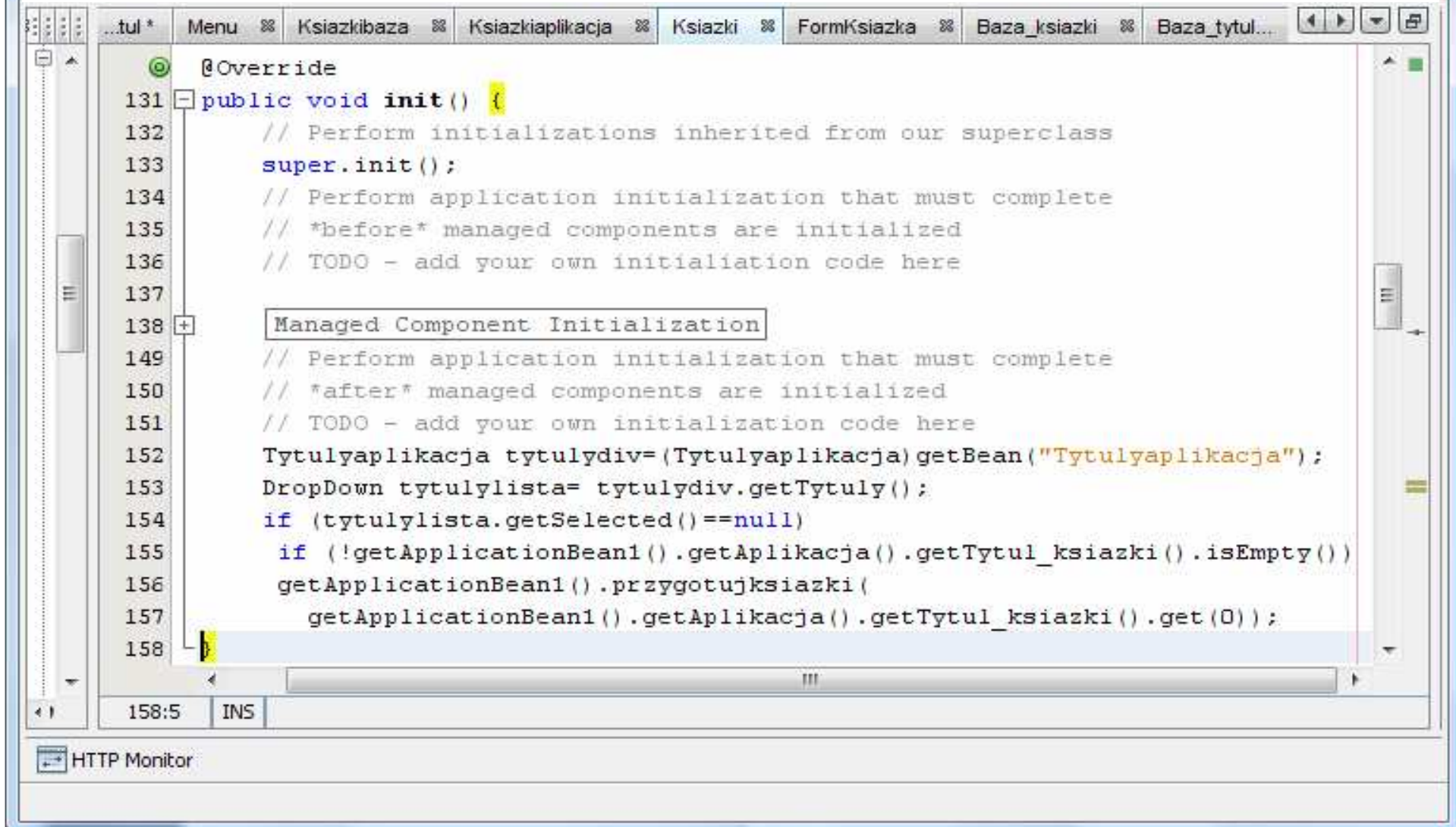
## Utworzenie warstwy biznesowej oraz obiektu typu TApplikacja, który jest fasadą warstwy biznesowej w postaci zwykłego obiektu Javy



```
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
...java SessionBean1.java ApplicationBean1.java * Page1 Logo...
@Override
63 public void init() {
64     // Perform initializations inherited from our sup
65     super.init();
66     // Perform application initialization that must c
67     // *before* managed components are initialized
68     // TODO - add your own initialization code here
69     Managed Component Initialization
80     // Perform application initialization that must c
81     // *after* managed components are initialized
82     // TODO - add your own initialization code here
83
84
85
86
87 }
88 private TApplikacja aplikacja = new TApplikacja();
89 public TApplikacja getApplikacja() {
90     return aplikacja;
91 }
92 public void setApplikacja(TApplikacja aplikacja) {
93     this.aplikacja = aplikacja;
94 }
```

## 2.2. Oprogramowanie dotyczące formularza Ksiazki.jsp

Definicje metody `init` w klasie `Ksiazki` dla stron typu JSP – zainicjowanie zawartości komponentu typu DropDown List na stronie `Ksiazkiaplikacja` typu JSPF informacjami o książkach (metoda `przygotujksiazki` opisana dalej) przez wybór pierwszego tytułu na stronie `Tytulyaplikacja` typu JSPF (o ile ten zbiór tytułów nie jest pusty)

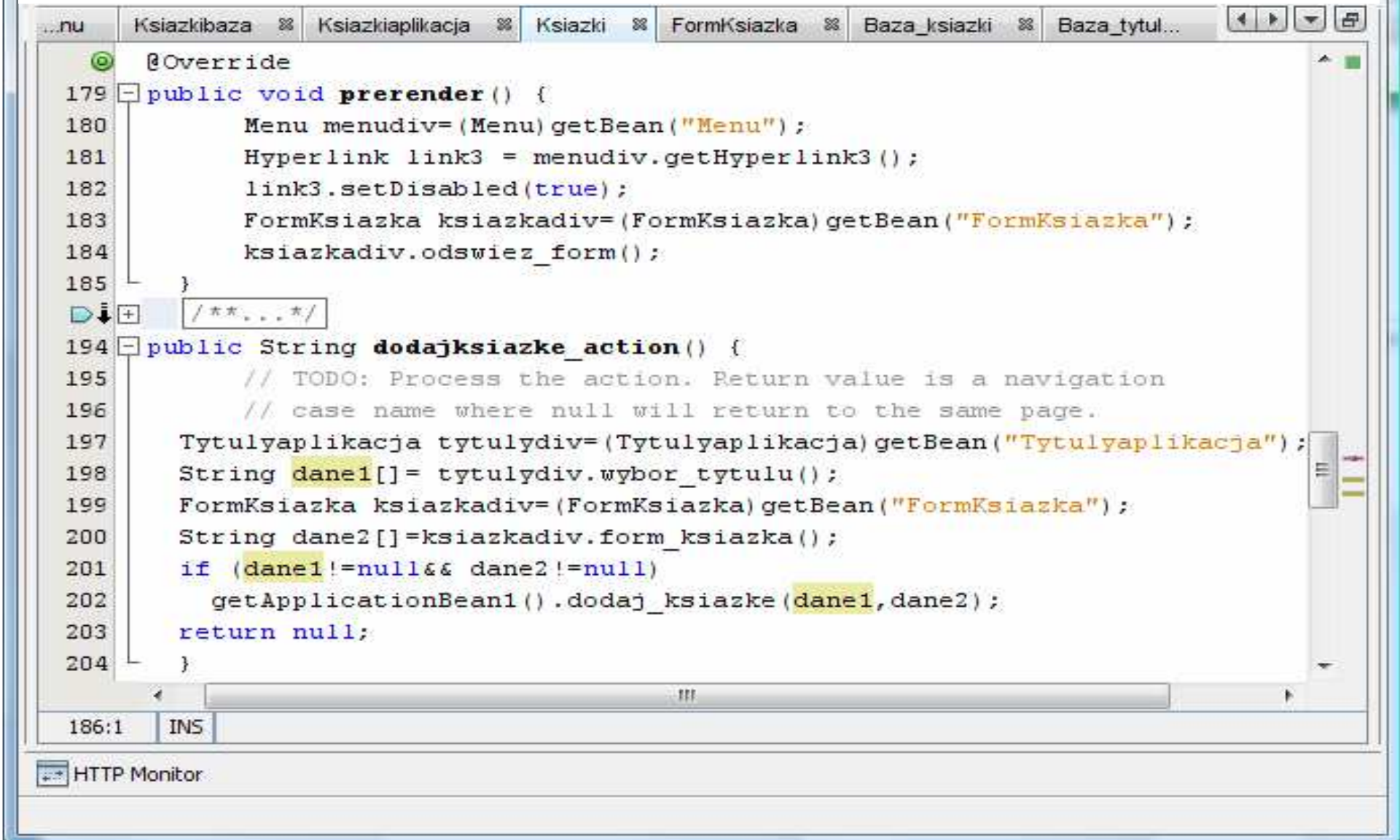


```
...tul *  Menu  Ksiazkibaza  Ksiazkiaplikacja  Ksiazki  FormKsiazka  Baza_ksiazki  Baza_tytul...
@Override
131 public void init() {
132     // Perform initializations inherited from our superclass
133     super.init();
134     // Perform application initialization that must complete
135     // *before* managed components are initialized
136     // TODO - add your own initialization code here
137
138     Managed Component Initialization
139
140     // Perform application initialization that must complete
141     // *after* managed components are initialized
142     // TODO - add your own initialization code here
143
144     Tytulyaplikacja tytulydiv=(Tytulyaplikacja)getBean("Tytulyaplikacja");
145     DropDown tytulylista= tytulydiv.getTytuly();
146     if (tytulylista.getSelected()==null)
147         if (!getApplicationBean1().getAplikacja().getTytul_ksiazki().isEmpty())
148             getApplicationBean1().przygotujksiazki(
149                 getApplicationBean1().getAplikacja().getTytul_ksiazki().get(0));
150
151 }
```

158:5 INS

HTTP Monitor

**Definicje metod w klasie Ksiazki dla strony typu JSP – do wstawiania nowej książki do warstwy biznesowej (obsługa zdarzenia dodajksiazke\_action) oraz generowania widoku w fazie Response przetwarzania strony metoda prerender (wygaszanie linku do bieżącej strony w formularzu Menu typu JSPF i czyszczenie pól formularza FormKsiazka typu JSPF jego metodą odswiez\_form)**



```
...nu | Ksiazkibaza | Ksiazkiaplikacja | Ksiazki | FormKsiazka | Baza_ksiazki | Baza_tytul...
@Override
179 public void prerender() {
180     Menu menudiv=(Menu) getBean("Menu");
181     Hyperlink link3 = menudiv.getHyperlink3();
182     link3.setDisabled(true);
183     FormKsiazka ksiazkativ=(FormKsiazka) getBean("FormKsiazka");
184     ksiazkativ.odswiez_form();
185 }
+ /**...*/
194 public String dodajksiazke_action() {
195     // TODO: Process the action. Return value is a navigation
196     // case name where null will return to the same page.
197     Tytulyaplikacja tytulydiv=(Tytulyaplikacja) getBean("Tytulyaplikacja");
198     String dane1[]= tytulydiv.wybor_tytulu();
199     FormKsiazka ksiazkativ=(FormKsiazka) getBean("FormKsiazka");
200     String dane2[]=ksiazkativ.form_ksiazka();
201     if (dane1!=null&& dane2!=null)
202         getApplicationBean1().dodaj_ksiazke(dane1,dane2);
203     return null;
204 }
186:1 | INS |
HTTP Monitor
```

**Definicje metod w klasie FormKsiazka typu BackingBean dla stron typu JSPF – do pobierana danych o nowej książce (form\_ksiazka): dane1 dla wyszukania tytułów wstawianych książek (metoda wybor\_tytulu z ze strony Tytulyaplikacja typu JSPF) i dane2 z danymi wstawianej książki; oraz do czyszczenia pól formularza (odswiez\_form)**

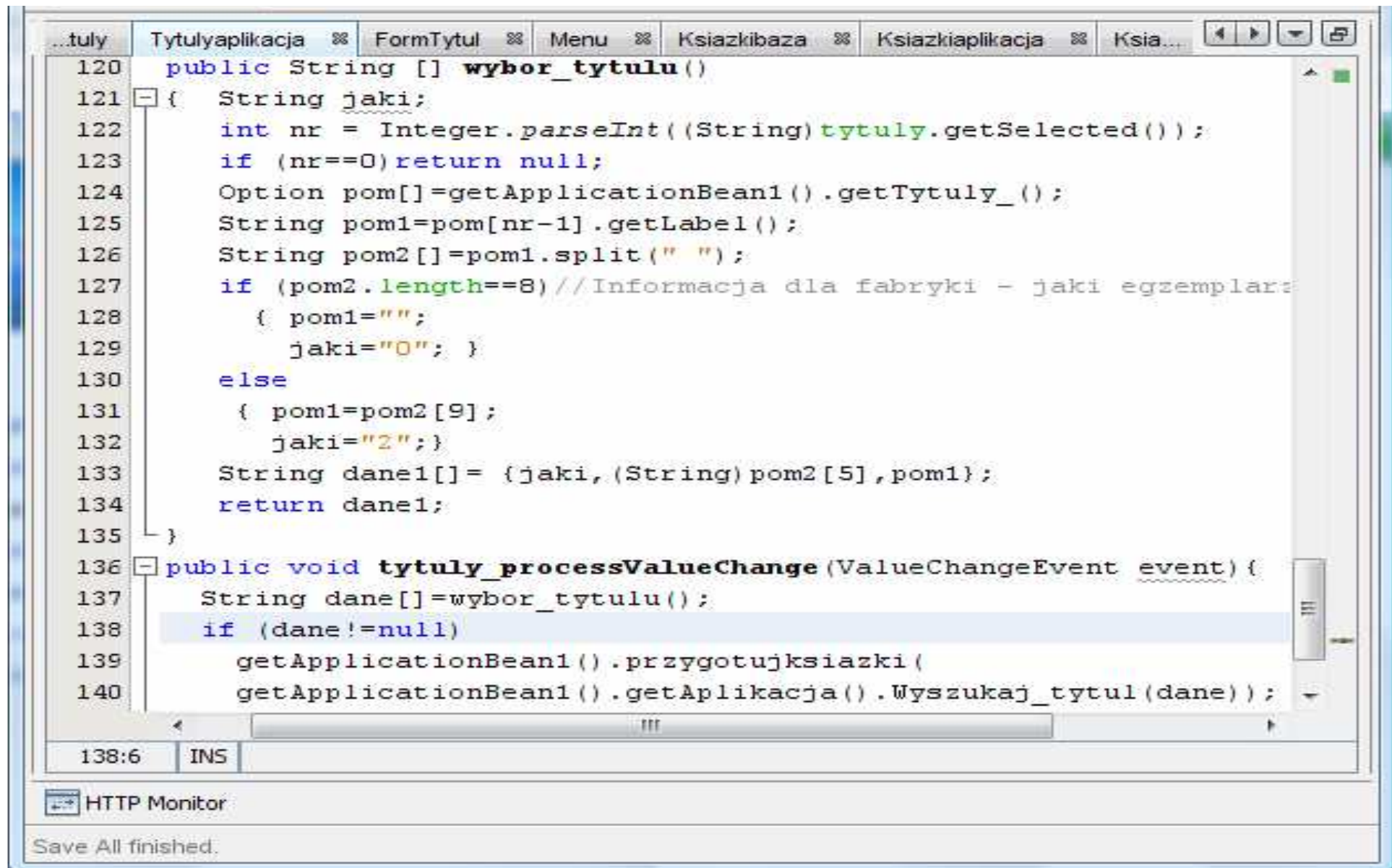
```
158 public void odswiez_form()
159 {
160     ksiazkiaplikacjaPanel.setRendered(true);
161     numer.setText("");
162     termin.setText("");
163 }
164 public String[] form_ksiazka() {
165     // TODO: Process the action. Return value is a navigation
166     // case name where null will return to the same page.
167     String jaka;
168     if (numer.getText().equals(""))
169         return null;
170     if (termin.getText().equals(""))
171     { jaka = "0"; }
172     else
173     { jaka = "1"; } // Informacja dla fabryki - jaki tytul wyszukac
174     String dane[] = {jaka, (String) numer.getText(), (String) termin.getText()};
175     return dane;
176 }
```

168:39 INS

HTTP Monitor



Definicje metod w klasie Tytulyaplikacja typu BackingBean dla stron typu JSPF – do pobierania danych o tytule dane1 do wyszukiwania (wybor\_tytulu): ISBN lub ISBN i nazwisko aktora; oraz do zmiany zawartości komponentu typu DropDown List na stronie Ksiazkiaplikacja typu JSPF (metodą przygotujksiazki) za pomocą obsługi zdarzenia wyboru pozycji z listy metodą tytuly\_processValueChange



```
...tytu  Tytulyaplikacja  FormTytul  Menu  Ksiazkibaza  Ksiazkiaplikacja  Ksia...
120  public String [] wybor_tytulu()
121  {  String jaki;
122     int nr = Integer.parseInt((String)tytuly.getSelected());
123     if (nr==0) return null;
124     Option pom[]=getApplicationBean1().getTytuly_();
125     String pom1=pom[nr-1].getLabel();
126     String pom2[]=pom1.split(" ");
127     if (pom2.length==8) //Informacja dla fabryki - jaki egzemplarz
128     { pom1="";
129       jaki="0"; }
130     else
131     { pom1=pom2[9];
132       jaki="2";}
133     String dane1[] = {jaki, (String) pom2[5], pom1};
134     return dane1;
135  }
136  public void tytuly_processValueChange (ValueChangeEvent event) {
137     String dane[]=wybor_tytulu();
138     if (dane!=null)
139     getApplicationBean1().przygotujksiazki (
140     getApplicationBean1().getApplikacja().Wyszukaj_tytul(dane));
```

138:6 INS

HTTP Monitor

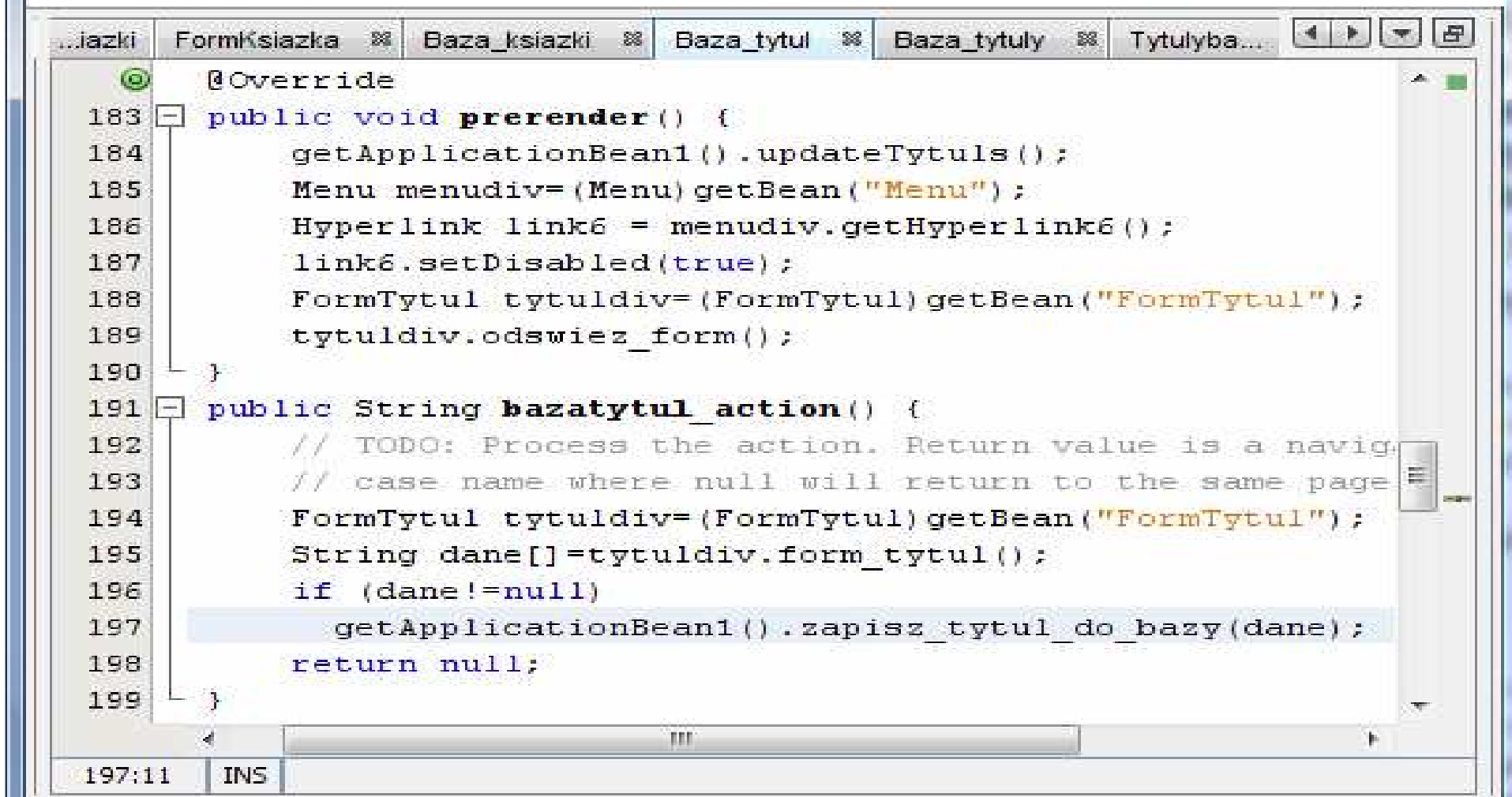
Save All finished.

**Definicje metod w klasie ApplicationBean1 związanych z zapisem (dodaj\_książke) i odczytem (przygotujksiążki) danych typu kolekcja obiektów TEgzemplarz i TEgzemplarz\_termin w warstwie biznesowej – odczytane dane wstawiane są do tablicy książki\_, która jest wyświetlana w komponencie typu DropDown List na stronie Książki aplikacja typu JSPF**

```
129 public void przygotujksiążki (TTytul_książki tytuł) {
130     if (tytuł == null) {
131         return;
132     }
133     ArrayList<TEgzemplarz> książki = tytuł.getMKsiążka();
134     int ile = książki.size();
135     if (ile > 0) {
136         Option pom[] = new Option[ile];
137         Iterator iterator = książki.iterator();
138         int i = 0;
139         while (iterator.hasNext()) {
140             pom[i++] = new Option(Integer.toString(i), iterator.next().toString());
141         }
142         książki_ = pom;
143     } else {
144         książki_ = new Option[0];
145     }
146 }
```

## 2.3. Oprogramowanie dotyczące formularza „Dodaj tytuł do bazy”

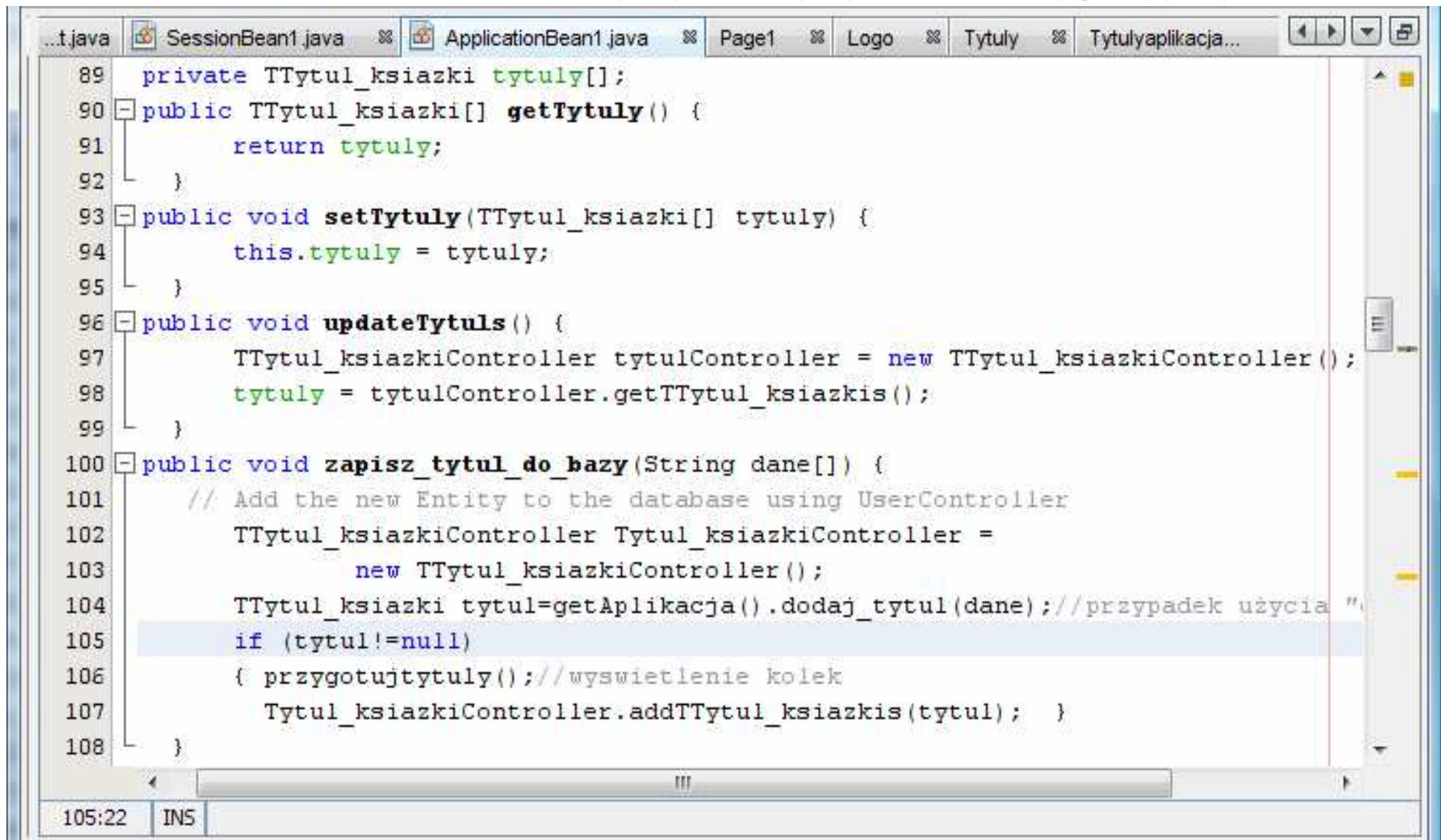
Definicje metod w klasie `Baza_tytul` dla strony typu JSP – do zapisu tytułów z warstwy biznesowej (obsługa zdarzenia `bazatytul_action`) oraz generowania widoku w fazie Response przetwarzania strony metoda `prerender` (wygaszanie linku do bieżącej strony w formularzu Menu typu JSPF, aktualizacja tablicy tytuły metodą `updateTytuls` w klasie `ApplicationBean1`, wyświetlanej w komponencie Table strony `Tytulybaza` typu JSPF, czyszczenie pól formularza `FormTytul` typu JSPF jego metodą `odswiez_form`)



```
...iazki | FormKsiazka | Baza_ksiazki | Baza_tytul | Baza_tytuly | Tytulyba...
@Override
183 public void prerender() {
184     getApplicationBean1().updateTytuls();
185     Menu menudiv=(Menu) getBean("Menu");
186     Hyperlink link6 = menudiv.getHyperlink6();
187     link6.setDisabled(true);
188     FormTytul tytuldiv=(FormTytul) getBean("FormTytul");
189     tytuldiv.odswiez_form();
190 }
191 public String bazatytul_action() {
192     // TODO: Process the action. Return value is a navig
193     // case name where null will return to the same page
194     FormTytul tytuldiv=(FormTytul) getBean("FormTytul");
195     String dane[]=tytuldiv.form_tytul();
196     if (dane!=null)
197         getApplicationBean1().zapisz_tytul_do_bazy(dane);
198     return null;
199 }
```

197:11 | INS

Definicje metod w klasie ApplicationBean1: odczytującą dane typu kolekcja obiektów TTytul\_książki i TTytul\_książki\_na\_kasecie z bazy danych (updateTytuls) oraz metoda zapisz\_tytul\_do\_bazy zapisująca do bazy danych (addTTytul\_książkis) i warstwy biznesowej (dodaj\_tytul – dokonano zmiany nagłówka metody wg slajdu 38) pojedynczych danych typu TTytul\_książki i TTytul\_książki\_na\_kasecie oraz odświeżająca zawartość komponentu typu DropDownList na stronie Tytulyaplikacja typu JSPF (przygotujtytuly)

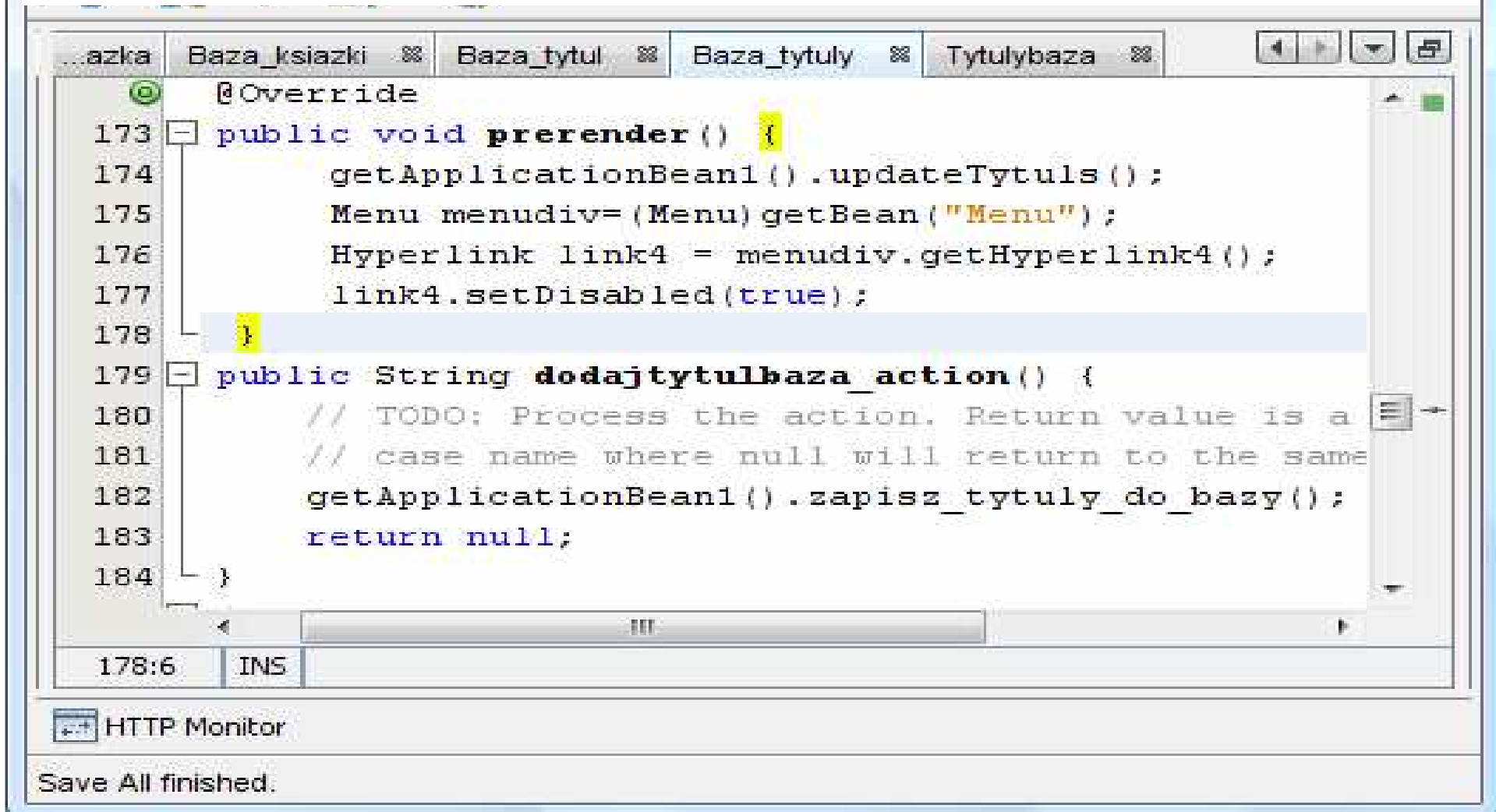


```
..t.java  SessionBean1.java  ApplicationBean1.java  Page1  Logo  Tytuly  Tytulyaplikacja...
89  private TTytul_książki tytuly[];
90  public TTytul_książki[] getTytuly() {
91      return tytuly;
92  }
93  public void setTytuly(TTytul_książki[] tytuly) {
94      this.tytuly = tytuly;
95  }
96  public void updateTytuls() {
97      TTytul_książkiController tytulController = new TTytul_książkiController();
98      tytuly = tytulController.getTTytul_książkis();
99  }
100 public void zapisz_tytul_do_bazy(String dane[]) {
101     // Add the new Entity to the database using UserController
102     TTytul_książkiController Tytul_książkiController =
103         new TTytul_książkiController();
104     TTytul_książki tytul=getAplikacja().dodaj_tytul(dane);//przypadek użycia ""
105     if (tytul!=null)
106     { przygotujtytuly();//wyswietlenie kolek
107         Tytul_książkiController.addTTytul_książkis(tytul); }
108 }
```

105:22 INS

## 2.4. Oprogramowanie dotyczące formularza Baza\_tytuly.jsp

Definicje metod w klasie Baza\_tytuly dla strony typu JSP – do zapisu tytułów z warstwy biznesowej (obsługa zdarzenia dodajtytulbaza\_action) oraz generowania widoku w fazie Response przetwarzania strony metoda prerender (wygaszanie linku do bieżącej strony w formularzu Menu typu JSPF i aktualizacja tablicy tytuły metodą updateTytuls w klasie ApplicationBean1 wyświetlanej w komponencie Table strony Tytulybaza typu JSPF)



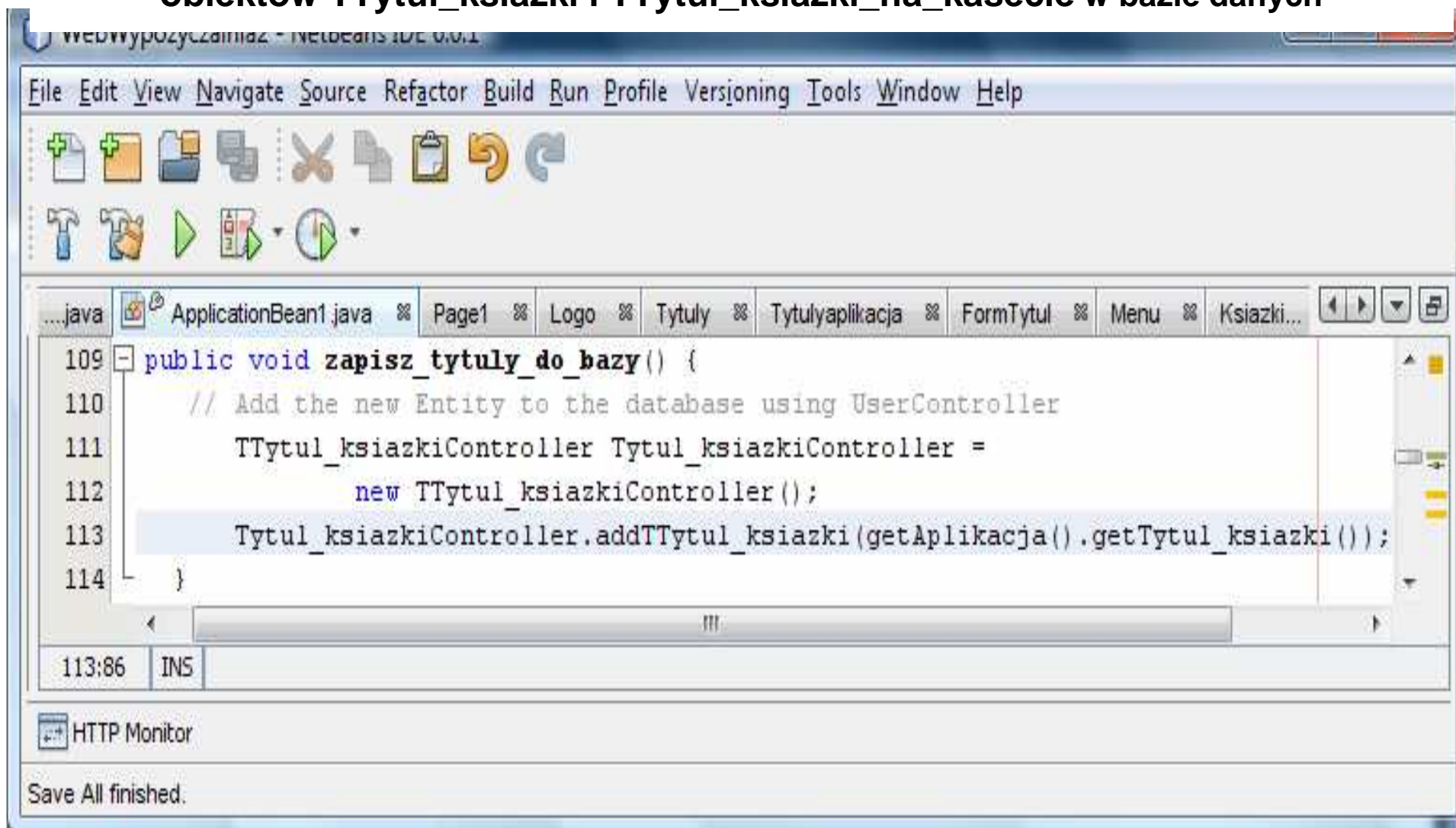
```
@Override
173 public void prerender() {
174     getApplicationBean1().updateTytuls();
175     Menu menudiv=(Menu)getBean("Menu");
176     Hyperlink link4 = menudiv.getHyperlink4();
177     link4.setDisabled(true);
178 }
179 public String dodajtytulbaza_action() {
180     // TODO: Process the action. Return value is a
181     // case name where null will return to the same
182     getApplicationBean1().zapisz_tytuly_do_bazy();
183     return null;
184 }
```

178:6 INS

HTTP Monitor

Save All finished.

## Definicje metody w klasie ApplicationBean1 związanej z zapisem danych typu kolekcja obiektów TTytul\_książki i TTytul\_książki\_na\_kascecie w bazie danych



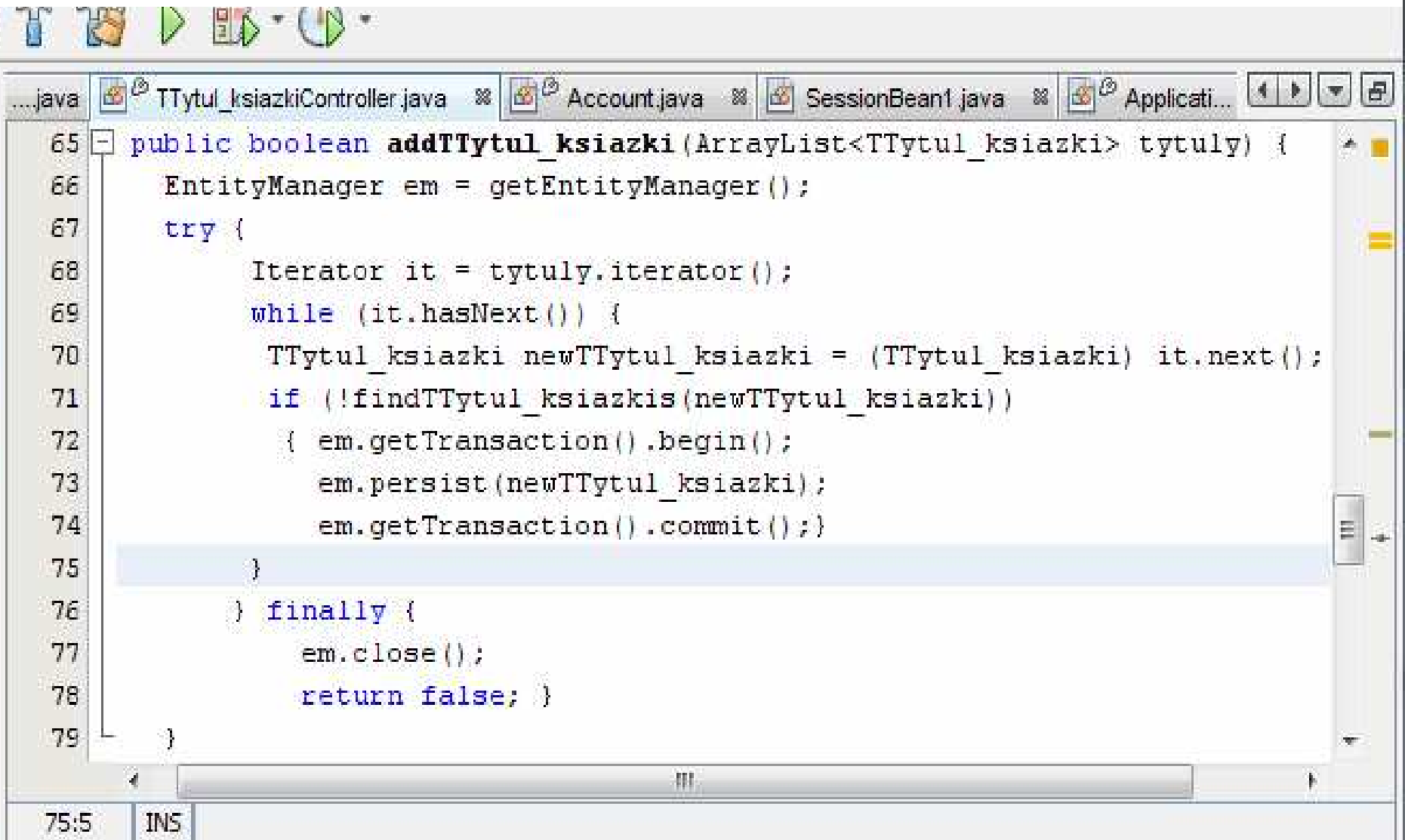
The screenshot shows the NetBeans IDE interface. The main editor window displays the following Java code:

```
109 public void zapisz_tytuly_do_bazy() {  
110     // Add the new Entity to the database using UserController  
111     TTytul_książkiController Tytul_książkiController =  
112         new TTytul_książkiController();  
113     Tytul_książkiController.addTTytul_książki(getAplikacja().getTytul_książki());  
114 }
```

The status bar at the bottom indicates the cursor is at line 113, column 86, with the text "INS". Below the editor, there is an "HTTP Monitor" tab and a message "Save All finished."

Definicja nowej metody `addTTytul_książki` w klasie `TTytul_książkiController` zapisującej kolekcję obiektów typu `TTytul_książki` i `TTytul_książki_na_kasecie` do bazy danych-  
wersja 1


`id = new Long(-1)` w konstruktorze encji `TTytul_książki`



```
65 public boolean addTTytul_książki(ArrayList<TTytul_książki> tytuly) {
66     EntityManager em = getEntityManager();
67     try {
68         Iterator it = tytuly.iterator();
69         while (it.hasNext()) {
70             TTYtul_książki newTTYtul_książki = (TTYtul_książki) it.next();
71             if (!findTTYtul_książkis(newTTYtul_książki))
72                 { em.getTransaction().begin();
73                   em.persist(newTTYtul_książki);
74                   em.getTransaction().commit();}
75         }
76     } finally {
77         em.close();
78         return false; }
79 }
```

75:5 INS

**Definicja nowej metody addTTYtul\_książki w klasie TTYtul\_książkiController zapisującej kolekcję obiektów typu TTYtul\_książki i TTYtul\_książki\_na\_kasecie do bazy danych - wersja 2 (id = null w konstruktorze encji TTYtul\_książki)**

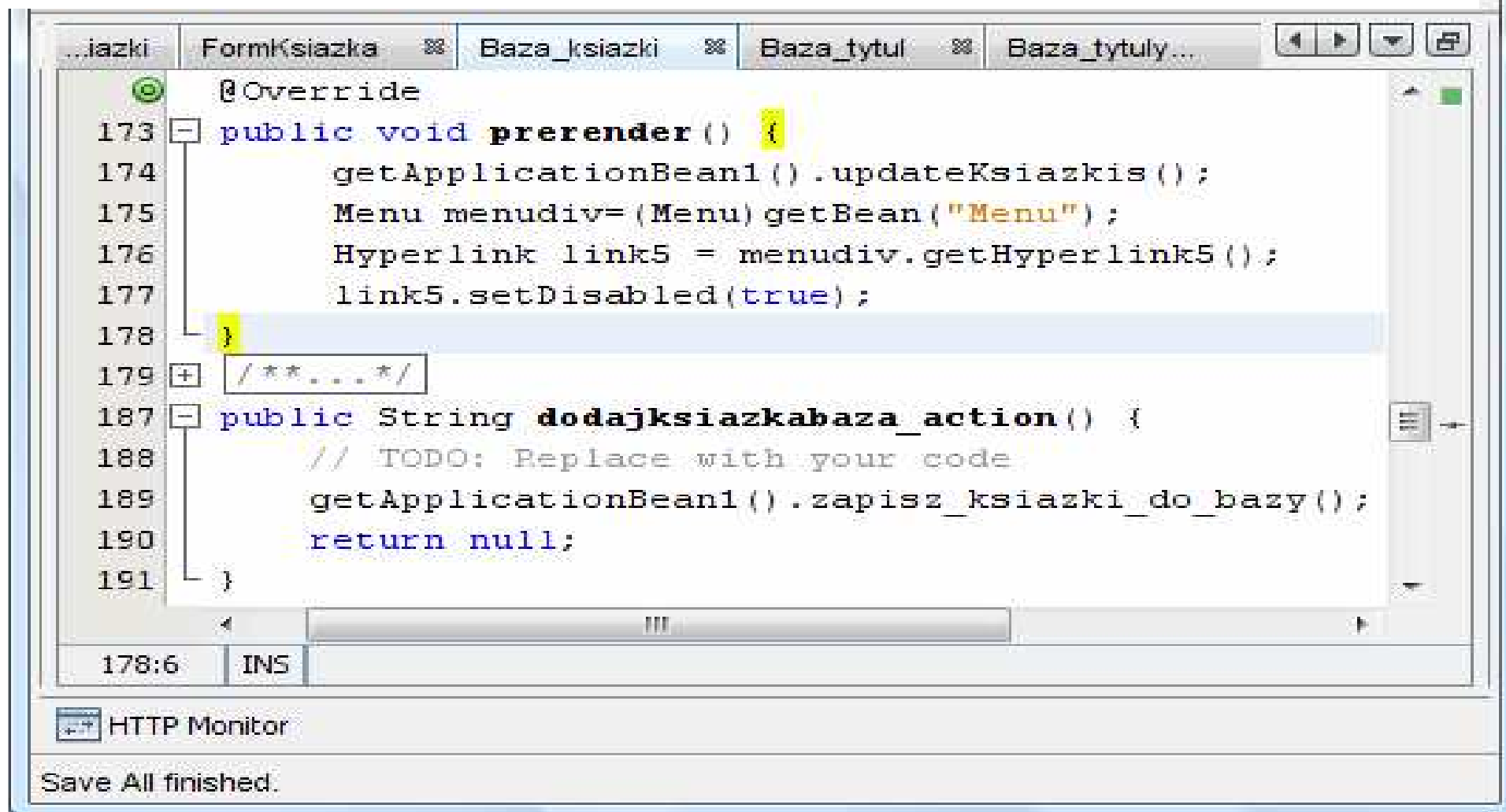


```
...a * | web.xml | Tytuly | TTYtul_książkiController.java | TApplikacja.java | b...  
[-] public boolean addTTYtul_książki (ArrayList<TTYtul_książki> tytuly) {  
    EntityManager em = getEntityManager();  
    TTYtul_książki newTTYtul_książki=null;  
    try {  
        Iterator it = tytuly.iterator();  
        em.getTransaction().begin();  
        while (it.hasNext()) {  
            newTTYtul_książki = (TTYtul_książki) it.next();  
            if (newTTYtul_książki.getId()==null)  
                em.merge(newTTYtul_książki);  
        }  
        em.persist(newTTYtul_książki);  
        em.getTransaction().commit();  
    } finally {  
        em.close();  
        return false; }  
}
```



## 2.5. Oprogramowanie dotyczące formularza Baza\_książki.jsp

Definicje metod w klasie Baza\_książki dla strony typu JSP – do zapisu książek z warstwy biznesowej (obsługa zdarzenia dodajksiążkabaza\_action) oraz generowania widoku w fazie Response przetwarzania strony metoda prerender (wygaszanie linku do bieżącej strony w formularzu Menu typu JSPF i aktualizacja tablicy książki metodą updateKsiążkis w klasie ApplicationBean1 wyświetlanej w komponencie Table strony Książkibaza typu JSPF)



```
...iazki | FormKsiążka | Baza_książki | Baza_tytul | Baza_tytuly...
@Override
173 public void prerender() {
174     getApplicationBean1().updateKsiążkis();
175     Menu menudiv = (Menu) getBean("Menu");
176     Hyperlink link5 = menudiv.getHyperlink5();
177     link5.setDisabled(true);
178 }
179 /**...*/
187 public String dodajksiążkabaza_action() {
188     // TODO: Replace with your code
189     getApplicationBean1().zapisz_książki_do_bazy();
190     return null;
191 }
178:6 | INS
HTTP Monitor
Save All finished.
```

**Definicje metod w klasie ApplicationBean1 związanych z zapisem (zapisz\_książki\_do\_bazy) i odczytem (updateKsiążkis) danych typu kolekcja obiektów TEgzemplarz i TEgzemplarz\_termin w bazie danych**



...java ApplicationBean1.java Page1 Logo Tytuly Tytulyaplikacja FormTytul Menu...

```
186 private TEgzemplarz ksiazki[];
187 public TEgzemplarz[] getKsiazki() {
188     return ksiazki;
189 }
190 public void setKsiazki(TEgzemplarz[] ksiazki) {
191     this.ksiazki = ksiazki;
192 }
193 public void updateKsiazkis() {
194     TEgzemplarzController egzController = new TEgzemplarzController();
195     ksiazki = egzController.getTEgzemplarz();
196 }
197 public void zapisz_ksiazki_do_bazy() {
198     // Add the new Entity to the database using UserController
199     TEgzemplarzController egzemplarzController =
200         new TEgzemplarzController();
201     egzemplarzController.addTEgzemplarze(getAplikacja().getTytul_ksiazki());
202 }
```

186:3 INS

HTTP Monitor

**Definicja nowej metody addTEgzemplarze w klasie TEgzemplarzController zapisującej kolekcję obiektów typu TEgzemplarz i TEgzemplarz\_termin do bazy danych – wersja 1  
id = new Long(-1) w konstruktorze encji TEgzemplarz**

```
56 public boolean addTEgzemplarze(ArrayList<TTytul_książki> tytuly) {  
57     EntityManager em = getEntityManager();  
58     Iterator it = tytuly.iterator();  
59     try {  
60         while (it.hasNext()) {  
61             TTytul_książki newTTytul_książki = (TTytul_książki) it.next();  
62             Iterator it_ = newTTytul_książki.getMKsiążka().iterator();  
63             while (it_.hasNext()) {  
64                 TEgzemplarz newTEgzemplarz = (TEgzemplarz) it_.next();  
65                 if (!findTEgzemplarzs(newTEgzemplarz))  
66                     { em.getTransaction().begin();  
67                       em.persist(newTEgzemplarz);  
68                       em.getTransaction().commit();}  
69                 }  
70             }  
71         } finally {  
72             em.close();  
73             return false;  
74         }  
75     }  
}
```

14:1

INS

HTTP Monitor

## Definicja nowej metody addTEgzemplarze w klasie TEgzemplarzController zapisującej kolekcję obiektów typu TEgzemplarz i TEgzemplarz\_termin do bazy danych – wersja 2 id = null w konstruktorze encji TEgzemplarz

```
...ava  TEGzemplarzController.java *  TEGzemplarz.java  TTYtul_ksiazki.java  TTYt...  
public boolean addTEgzemplarze(ArrayList<TTYtul_ksiazki> tytuly) {  
    EntityManager em = getEntityManager();  
    TEgzemplarz newTEgzemplarz = null;  
    Iterator it = tytuly.iterator();  
    em.getTransaction().begin();  
    try {  
        while (it.hasNext()) {  
            TTYtul_ksiazki newTTYtul_ksiazki = (TTYtul_ksiazki) it.next();  
            if(newTTYtul_ksiazki.getId()==null) continue;  
            Iterator it_ = newTTYtul_ksiazki.getMKsiazka().iterator();  
            while (it_.hasNext()) {  
                newTEgzemplarz = (TEgzemplarz) it_.next();  
                if (newTEgzemplarz.getId()==null)  
                    em.merge(newTEgzemplarz); }  
            }  
            em.persist(newTEgzemplarz);  
            em.getTransaction().commit();  
        } finally {  
            em.close();  
            return false; }  
    }
```

## 2.6. Zsynchronizowanie zawartości bazy danych i warstwy biznesowej przy starcie aplikacji – wywołanie metod update... przy tworzeniu obiektu ApplicationBean1

```
@Override
public void init() { // Perform initializations inherited from
    super.init();
    // Perform application initialization that must complete
    // *before* managed components are initialized
    // TODO - add your own initialization code here
    Managed Component Initialization
    // *after* managed components are initialized
    // TODO - add your own initialization code here
    updateTytuls();
    updateKsiazkis();
    updateAplikacja();
    przygotujtytul();
}

private T Aplikacja aplikacja = new T Aplikacja();
public T Aplikacja getAplikacja() {
    return aplikacja;
}

public void setAplikacja(T Aplikacja aplikacja) {
    this.aplikacja = aplikacja;
}

public void updateAplikacja() {
    for (int i = 0; i < tytul.length; i++)
        aplikacja.getTytul_ksiazki().add(tytul[i]);
    Iterator it = aplikacja.getTytul_ksiazki().iterator();
    while (it.hasNext()) {
        T Tytul_ksiazki tytul = (T Tytul_ksiazki) it.next();
        for (int j = 0; j < ksiazki.length; j++) {
            T Tytul_ksiazki tytul1 = ksiazki[j].getMTytul_ksiazki();
            if (tytul1 != null)
                if (tytul1.equals(tytul))
                    { tytul.getMKsiazka().add(ksiazki[j]); }
        }
    }
}
```

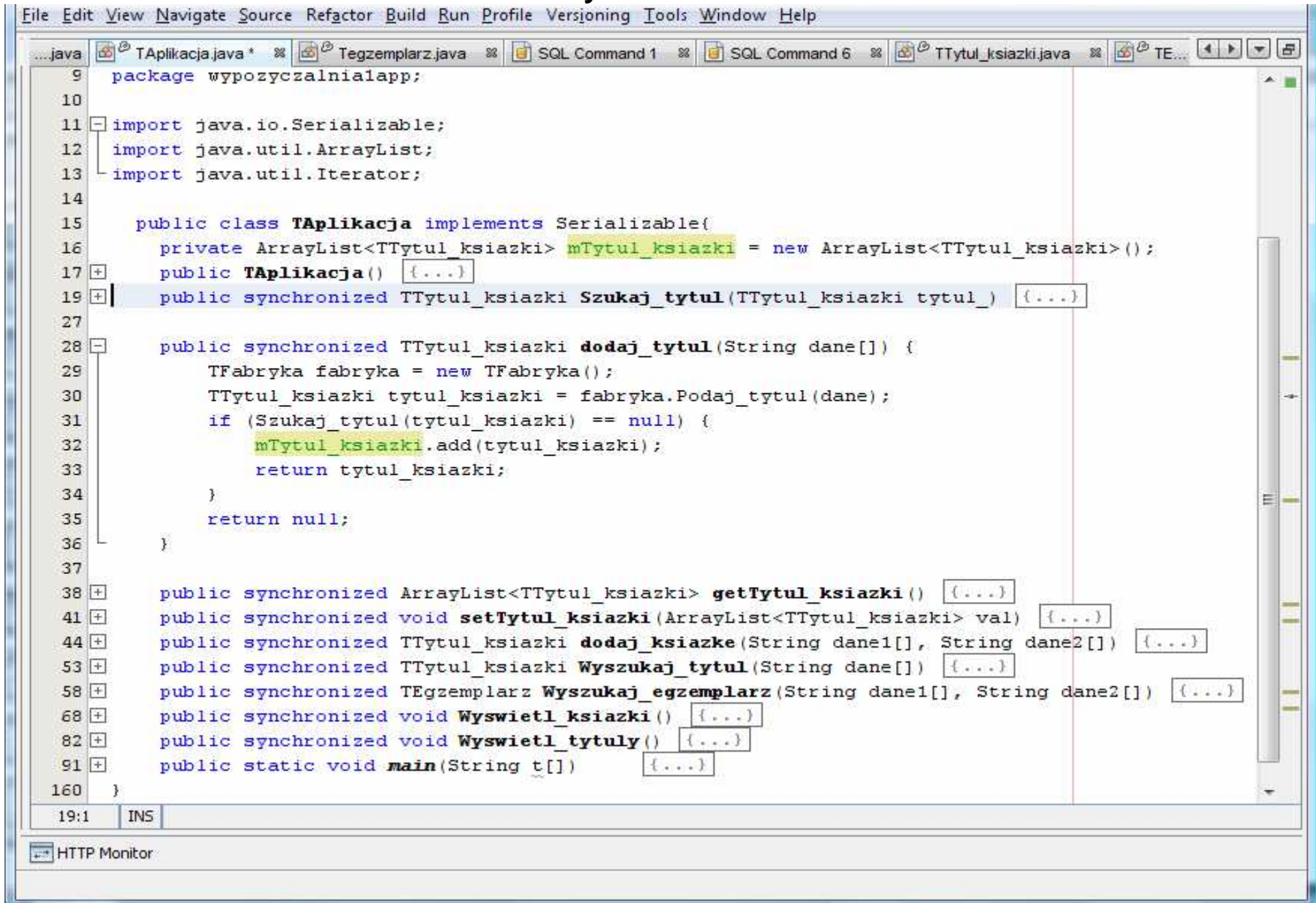
Nazwa metody po  
zestandaryzowaniu  
nazw dla atrybutu  
mTytul\_ksiazki w  
klasie TEgzemplarz

Stąd:

getMTytul\_ksiazki  
oraz

setMTytul\_ksiazki

## 2.7. Przystosowanie do pracy z wieloma wątkami warstwy biznesowej – metody typu synchronized



```
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
...java TApplikacja.java * T egzemplarz.java SQL Command 1 SQL Command 6 T Tytul_książki.java TE...
9 package wypożyczalniaapp;
10
11 import java.io.Serializable;
12 import java.util.ArrayList;
13 import java.util.Iterator;
14
15 public class TApplikacja implements Serializable{
16     private ArrayList<TTytul_książki> mTytul_książki = new ArrayList<TTytul_książki>();
17     public TApplikacja() {...}
19     public synchronized TTytul_książki Szukaj_tytul(TTytul_książki tytul_) {...}
27
28     public synchronized TTytul_książki dodaj_tytul(String dane[]) {
29         Tfabryka fabryka = new Tfabryka();
30         TTytul_książki tytul_książki = fabryka.Podaj_tytul(dane);
31         if (Szukaj_tytul(tytul_książki) == null) {
32             mTytul_książki.add(tytul_książki);
33             return tytul_książki;
34         }
35         return null;
36     }
37
38     public synchronized ArrayList<TTytul_książki> getTytul_książki() {...}
41     public synchronized void setTytul_książki(ArrayList<TTytul_książki> val) {...}
44     public synchronized TTytul_książki dodaj_książke(String dane1[], String dane2[]) {...}
53     public synchronized TTytul_książki Wyszukaj_tytul(String dane[]) {...}
58     public synchronized TEgzemplarz Wyszukaj_egzemplarz(String dane1[], String dane2[]) {...}
68     public synchronized void Wswietl_książki() {...}
82     public synchronized void Wswietl_tytuly() {...}
91     public static void main(String t[]) {...}
160 }
```

19:1 | INS

HTTP Monitor