

**Utrwalanie danych –  
zastosowanie obiektowego  
modelu danych warstwy  
biznesowej do generowania  
schematu relacyjnej bazy danych**  
**Przykład**

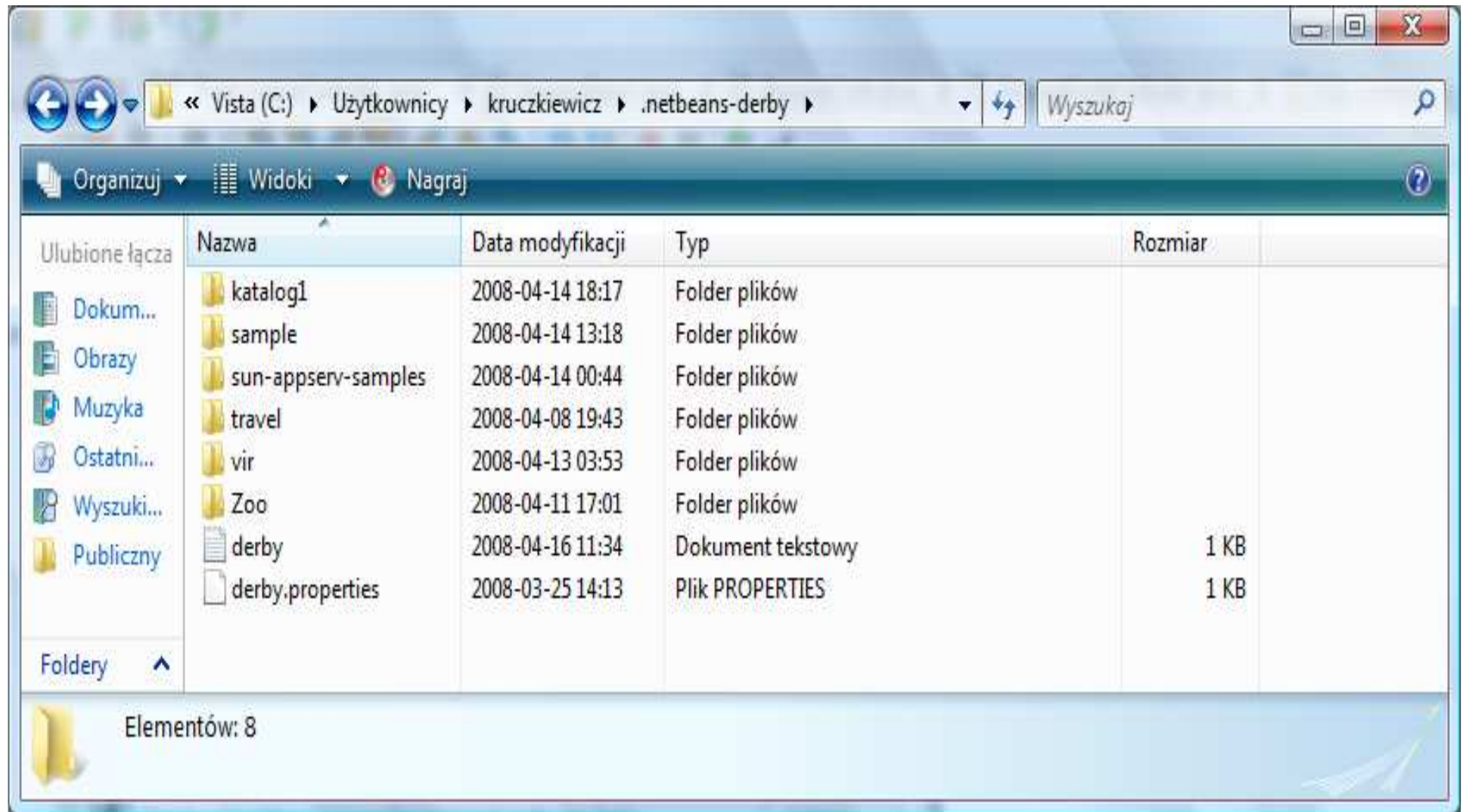
Autor

Zofia Kruczkiewicz

Wzorce oprogramowania 5

# **1. Tworzenie bazy danych w systemie baz danych Derby**

# 1.1. Zakładanie nowej bazy danych w domyślnym katalogu dla baz danych typu Derby (1)



## Zakładanie pustej bazy danych dla systemu baz danych Derby (2)

The screenshot shows the NetBeans IDE 6.0.1 interface. The 'Tools' menu is open, and the 'Java DB Database' sub-menu is selected, with 'Create Database...' highlighted. The project 'Wypożyczalnia1app' is visible in the left sidebar, and the 'TTytul\_książkiController.java' file is open in the editor. The code in the editor shows the implementation of the 'getEntityManager()' method, which uses 'EntityManagerFactory.createEntityManagerFactory()' to create an EntityManager.

```
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class TTytul_książkiController {
    private EntityManagerFactory emf = null;

    private void createEntityManager() {
        emf = Persistence.createEntityManagerFactory("Wypożyczalnia1appPU");
    }

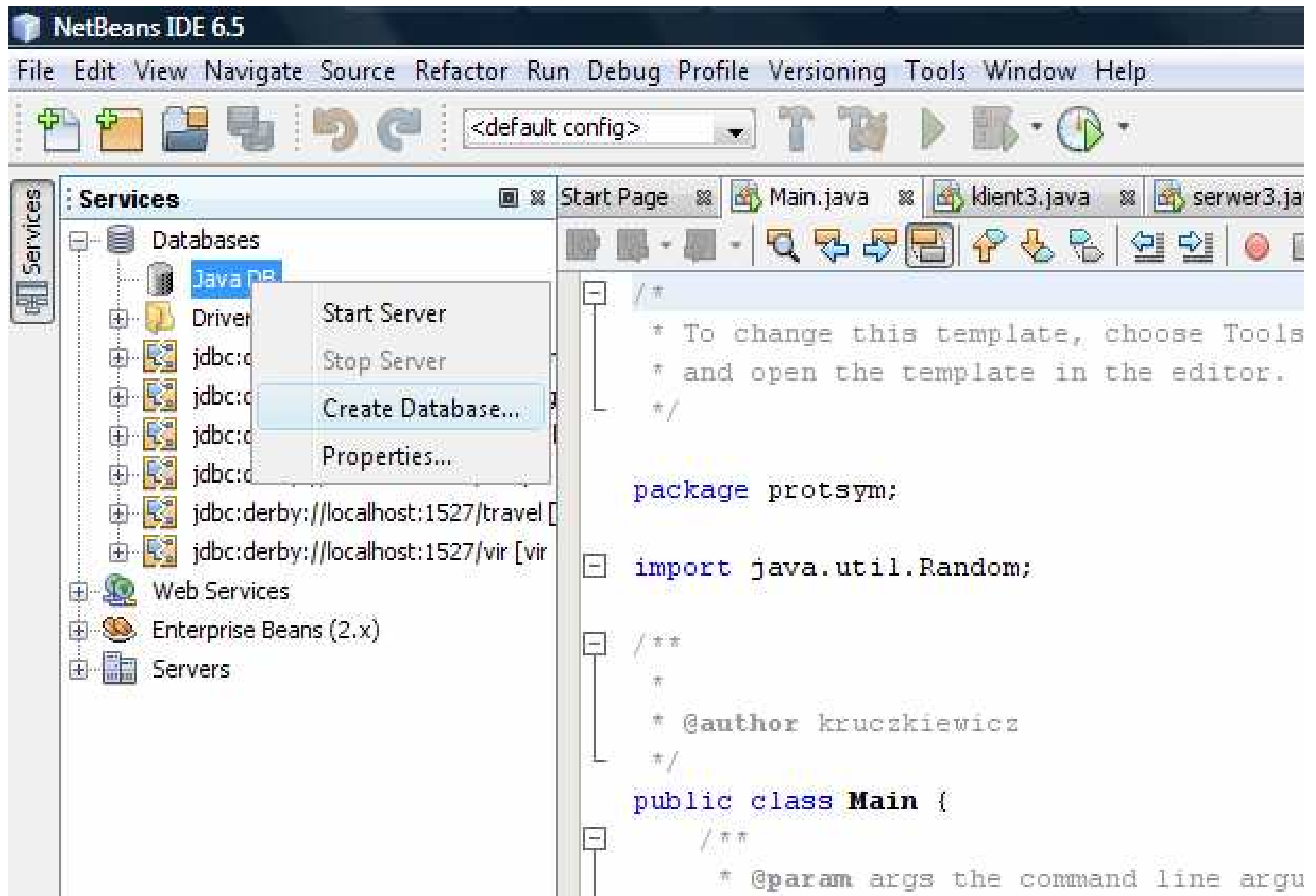
    public EntityManager getEntityManager() {
        if (emf == null) {
            createEntityManager();
        }
        return emf.createEntityManager();
    }

    public TTytul_książki[] getTTytul_książkis() {
        return (TTytul_książki[]) getTTytul_książki().toArray(new TTytul_książki[0]);
    }

    public List<TTytul_książki> getTTytul_książki() {
        EntityManager em = getEntityManager();
        try {
            javax.persistence.Query q =

```

## Zakładanie pustej bazy danych dla systemu baz danych Derby (3)



The screenshot shows the NetBeans IDE 6.5 interface. The 'Services' window is open, displaying a tree view of services. Under 'Databases', there is a 'Java DB' entry. A context menu is open over 'Java DB', showing the following options: 'Start Server', 'Stop Server', 'Create Database...' (highlighted), and 'Properties...'. The main editor window shows the following Java code:

```
/*
 * To change this template, choose Tools
 * and open the template in the editor.
 */

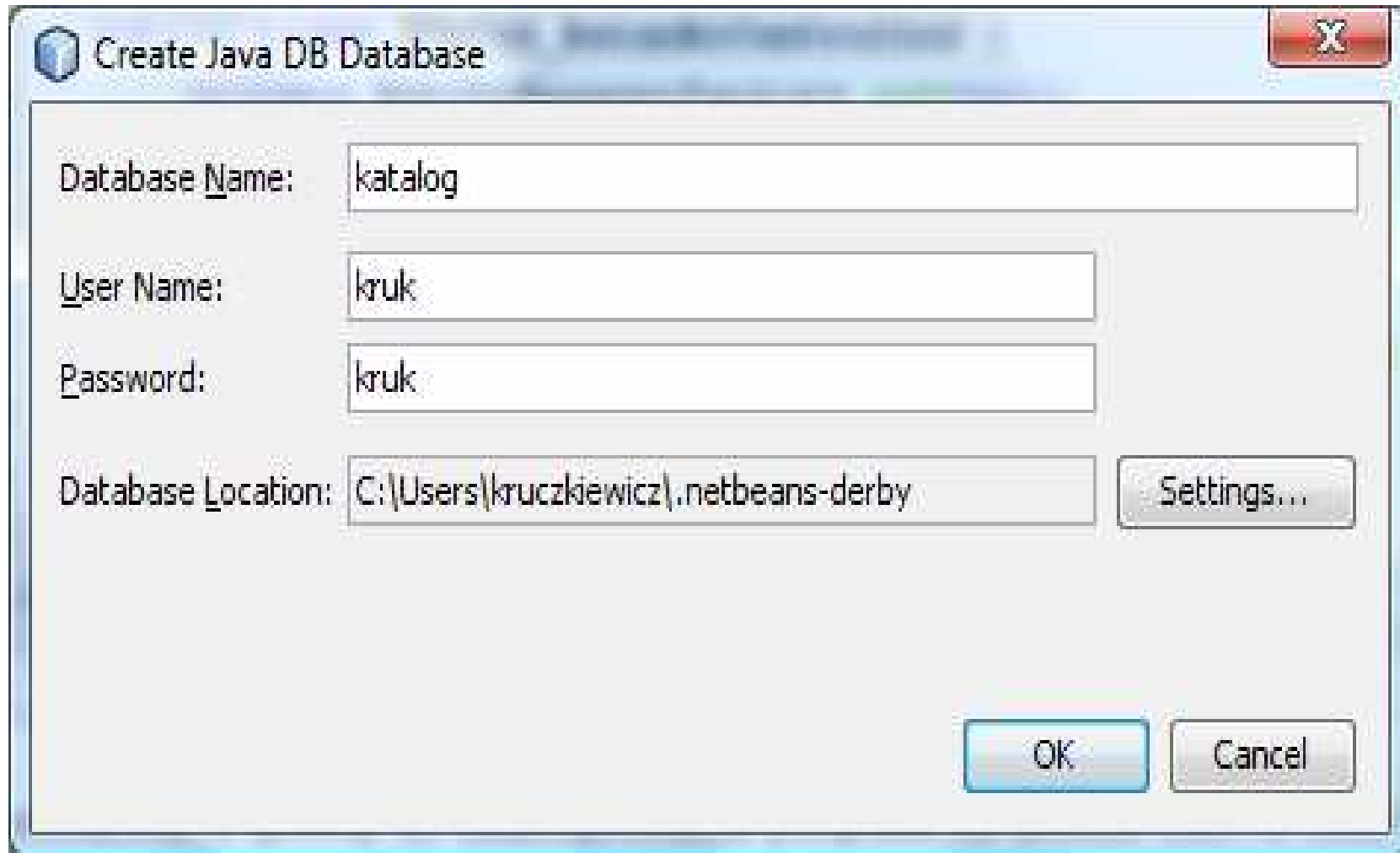
package protsym;

import java.util.Random;

/**
 *
 * @author kruczkiewicz
 */
public class Main {

    /**
     * @param args the command line argu
```

## Zakładanie bazy danych **katalog** w systemie baz danych Derby (4)

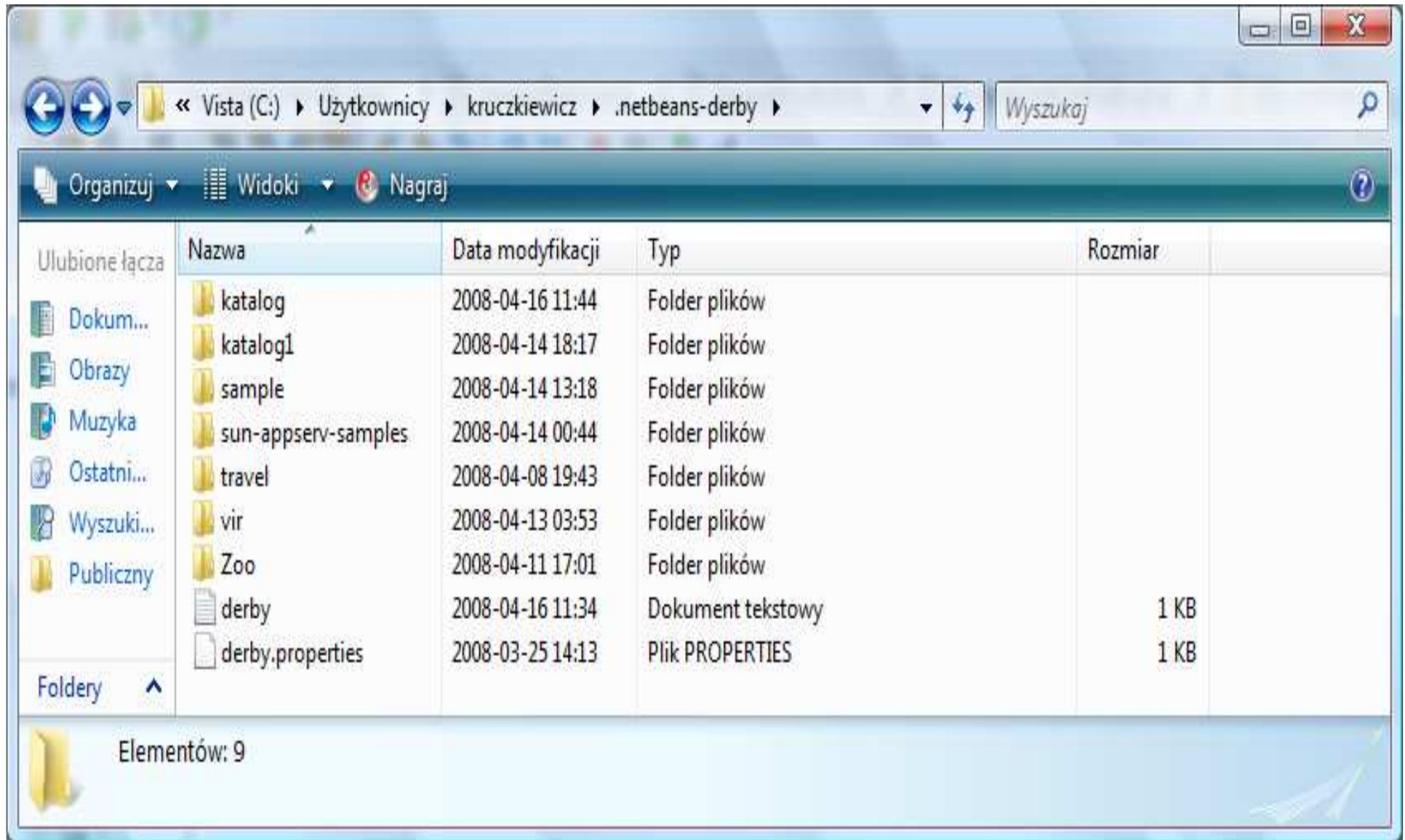


The screenshot shows a dialog box titled "Create Java DB Database" with the following fields and buttons:

- Database Name:** katalog
- User Name:** kruk
- Password:** kruk
- Database Location:** C:\Users\kruczkiewicz\.netbeans-derby

Buttons: Settings..., OK, Cancel

Założono nową bazę danych **katalog** w domyślnym katalogu dla baz danych typu Derby (5)





## 1.2. Łączenie z pustą bazą danych

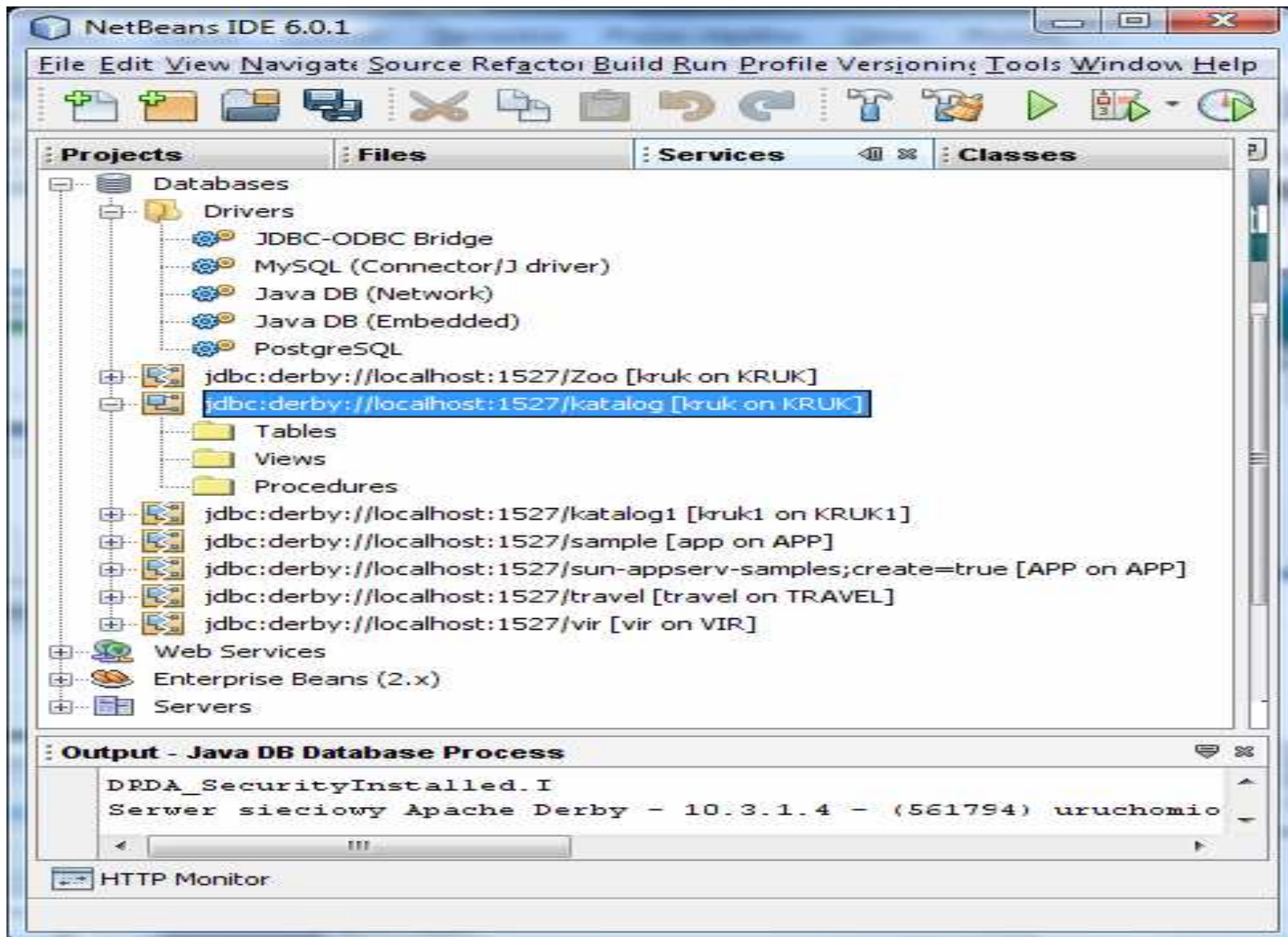
The screenshot displays the NetBeans IDE 6.0.1 interface. The main window shows the 'Welcome to NetBeans IDE 6.0.1' screen with a 'My NetBeans' sidebar. A context menu is open over the 'Databases' tree, with 'Connect...' selected. The 'Output - Java DB Database Process' window at the bottom shows the following text:

```
DRDA_SecurityInstalled.I  
Serwer sieciowy Apache Derby - 10.3.1.4 - (561794) uruchomiony i gotowy do zaakceptowania połączeń na porcie 1527 w 2008-04-16 09:34:18.928 GMT |
```

The 'HTTP Monitor' window is also visible at the bottom left. The system tray at the bottom shows the time as 11:55 and the language as PL.



## Połączenie z pustą bazą danych



## **2. Tworzenie warstwy integracji w projekcie Java Application. Zastosowanie wzorców projektowych typu **Domain Store** i **Transfer Object**.**

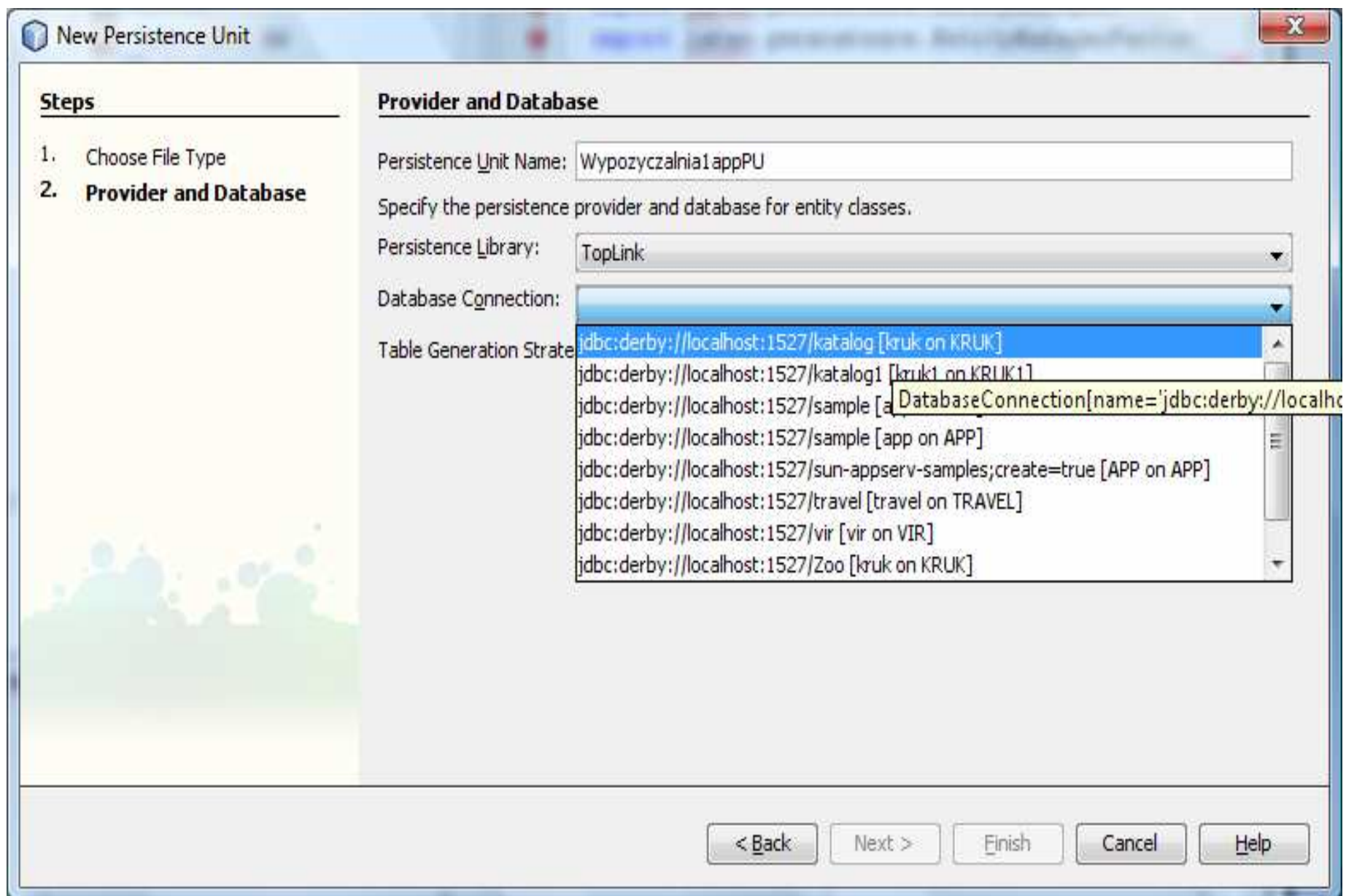
Przykład mapowania modelu obiektowego w  
środowisku NetBeans 6.01  
za pomocą technologii TopLink (przykład z  
Laboratorium3\_4).

## 2.1. Wstawianie do projektu typu **Java Application** modułu typu **Persistence Unit**

The screenshot shows the NetBeans IDE 6.0.1 interface. The 'New' menu is open, and 'Persistence Unit...' is selected. The code editor displays the following code:

```
java.util.List,  
javax.persistence.EntityManager;  
javax.persistence.EntityManagerFactory;  
javax.persistence.Persistence;  
  
hor kruczkiewicz  
  
class TTytul_książkiController {  
    private EntityManagerFactory emf=null;  
  
    private EntityManager getEntityManager() {  
        if (emf == null) {  
            emf = Persistence.createEntityManagerFactory("Wypożyczalnia1appPU");  
        }  
        return emf.createEntityManager();  
    }  
  
    public TTytul_książki[] getTTytul_książkis() {  
        return (TTytul_książki[])getTTytul_książki().toArray(new TTytul_książki[0]);  
    }  
  
    public List<TTytul_książki> getTTytul_książki() {  
        EntityManager em = getEntityManager();  
        try {  
            javax.persistence.Query q =
```

Wybór bazy danych, w której będą utrwalane obiekty - pustej





## Tworzenie modułu utrwalania danych dla technologii TopLink

**New Persistence Unit**

**Steps**

1. Choose File Type
2. **Provider and Database**

**Provider and Database**

Persistence Unit Name: Wypożyczalnia1appPU

Specify the persistence provider and database for entity classes.

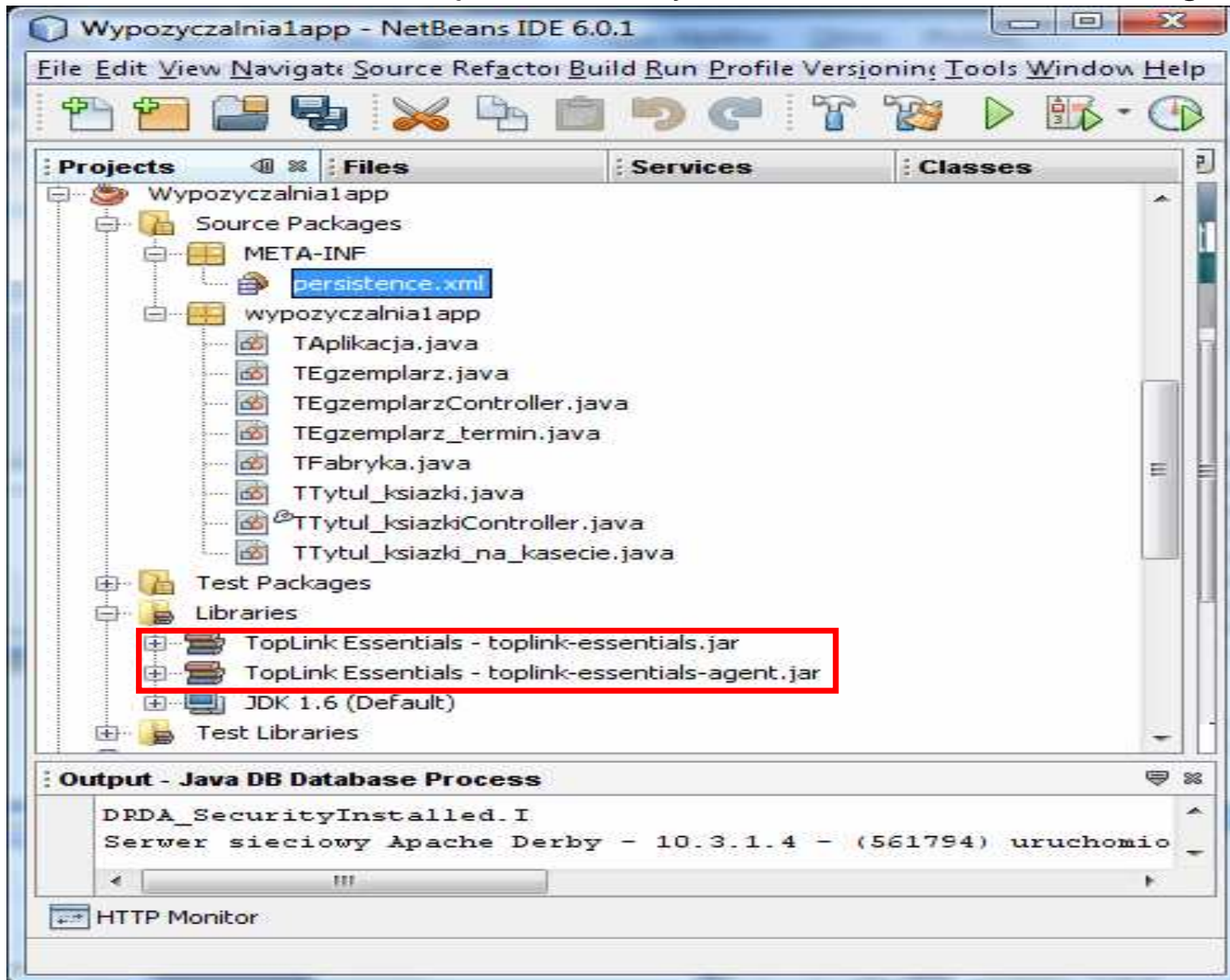
Persistence Library: TopLink

Database Connection: jdbc:derby://localhost:1527/katalog [kruk on KRUK]

Table Generation Strategy:  Create  Drop and Create  None

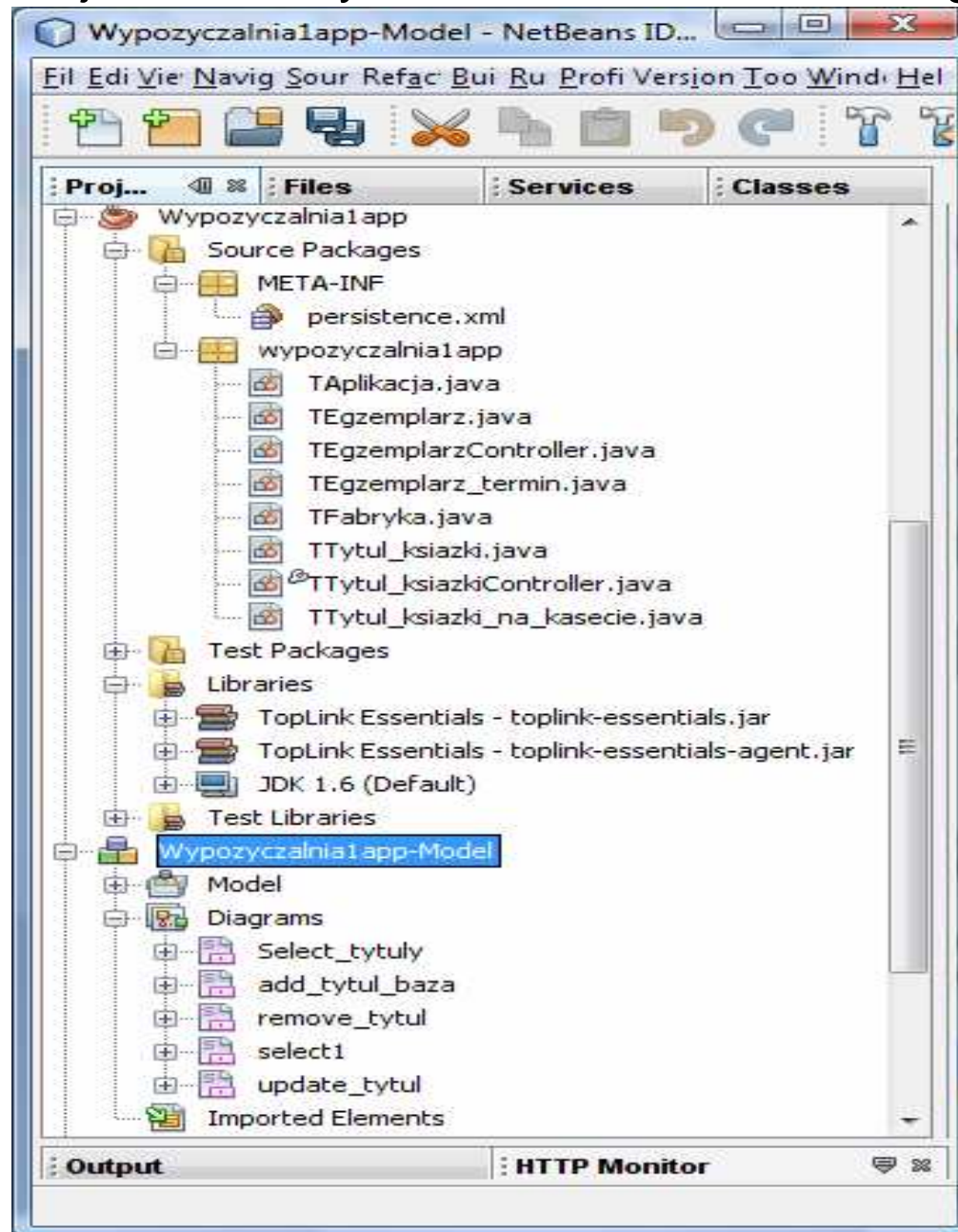
< Back   Next >   **Finish**   Cancel   Help

Plik **persistence.xml** reprezentujący moduł utrwalania danych typu **TopLink**  
Uzupełnianie zawartości projektu typu **Java Application** o klasy typu **Controller** dla każdej z utrwalanych klas modelu obiektowego

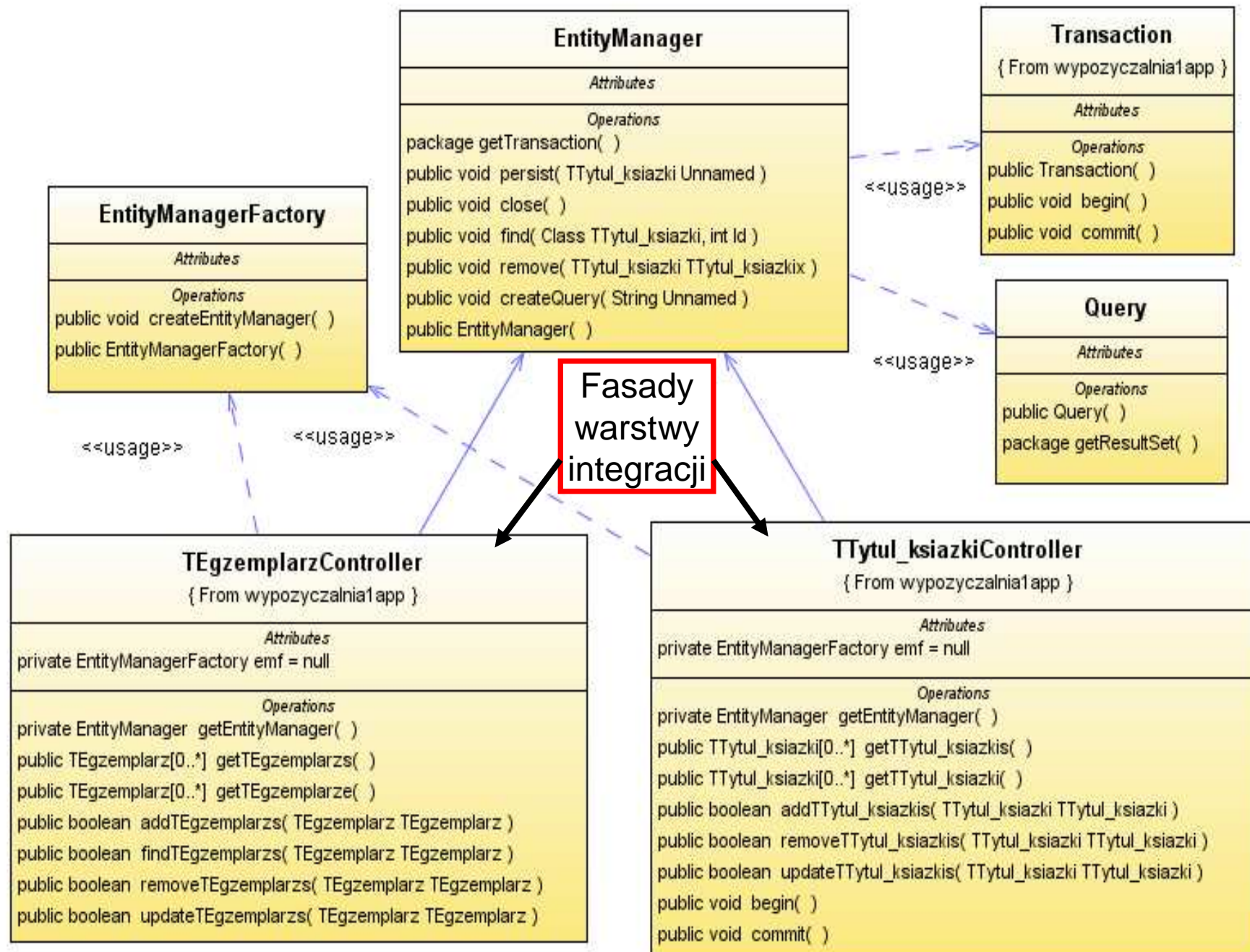




2.2. Inżynieria odwrotna dla utworzonych przez programistę klas typu **Controller** dla każdej z utrwalanych klas modelu obiektowego (**TTytul\_książki**)

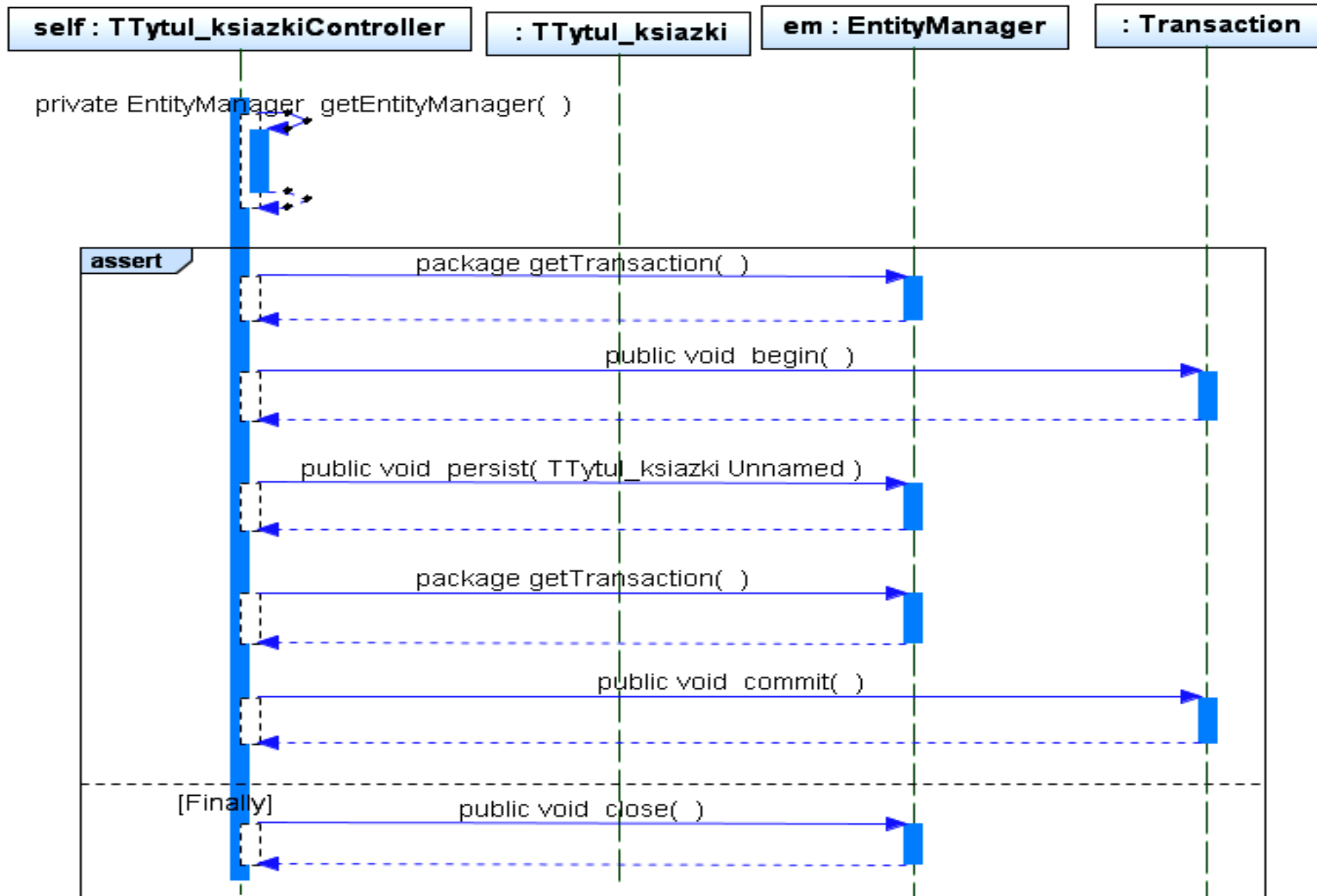


## Diagram klas – uproszczony schemat warstwy integracji

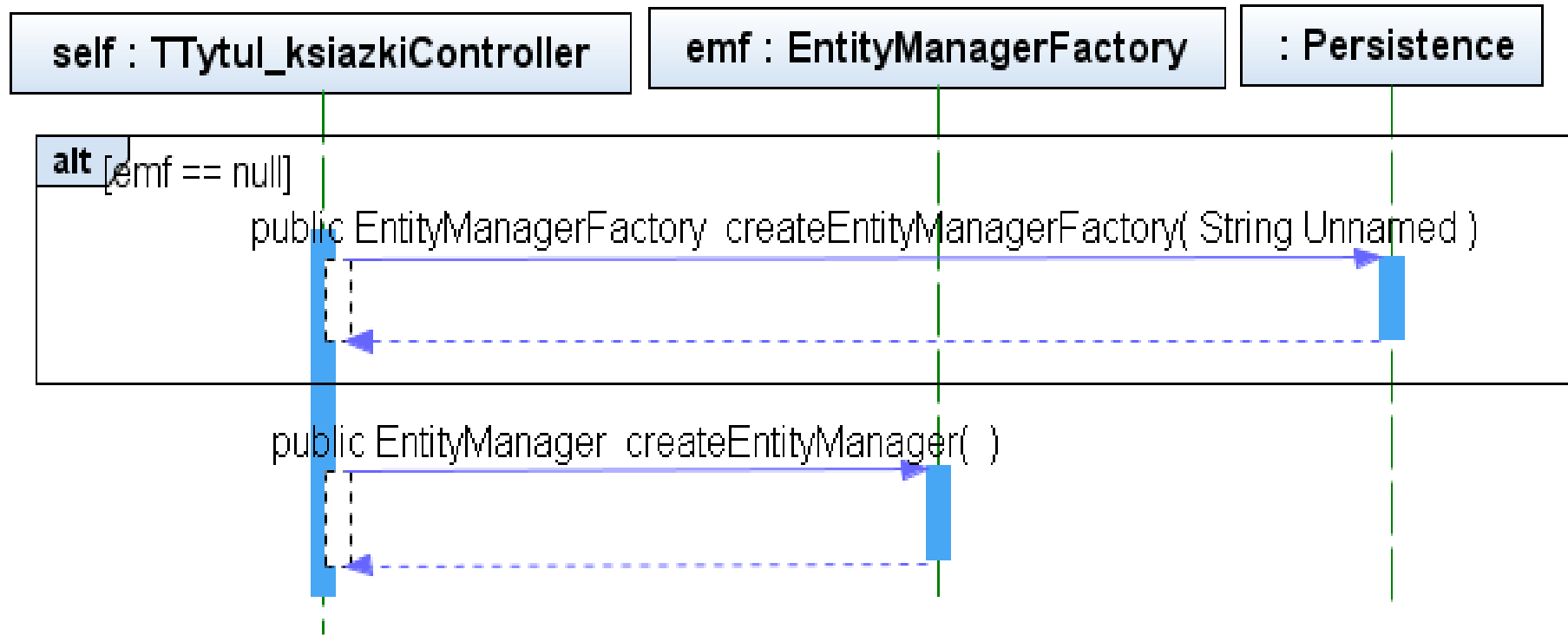


## Dodawanie tytułów do bazy danych -

**public boolean addTTytul\_książki(TTytul\_książki TTytul\_książki)**



## Dodawanie tytułów do bazy danych - **private EntityManager getEntityManager()**



**Klasa typu Controller dla każdej z klas utrwalanych  
obiektów – fragment 1**  
**Realizacja wzorców typu [Domain Store](#) i [Transfer Object](#)**

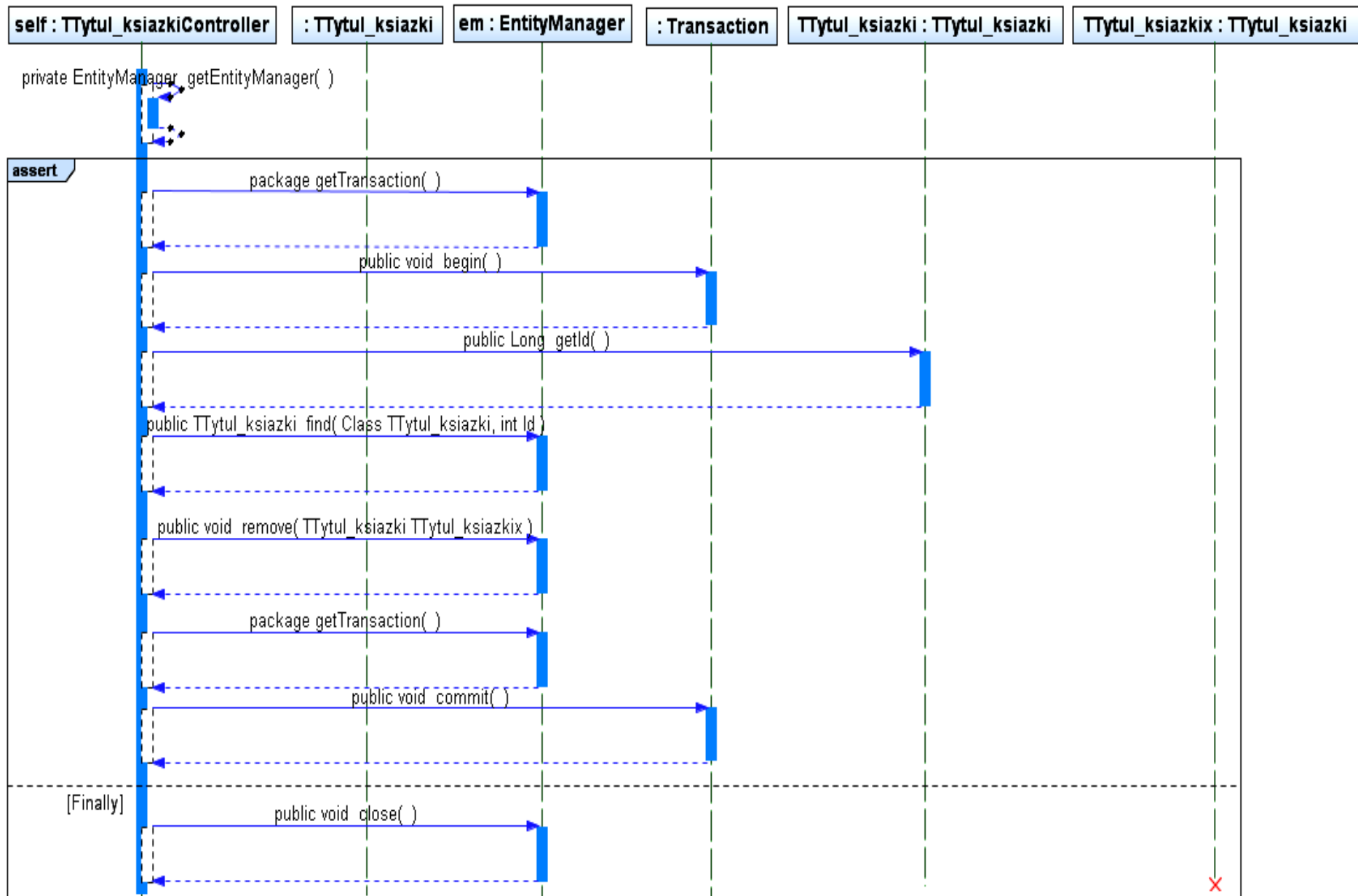
```
public class TTytul_ksiazkiController {  
    private EntityManagerFactory emf=null;  
  
    private EntityManager getEntityManager() {  
        if (emf == null) {  
            emf = Persistence.createEntityManagerFactory(  
                "Wypożyczalnia1appPU");  
        }  
        return emf.createEntityManager();  
    }  
}
```

```
public boolean addTTytul_ksiazkis(  
    TTYtul_ksiazki TTYtul_ksiazki) {  
    EntityManager em = getEntityManager();  
try {  
        em.getTransaction().begin();  
        em.persist(TTYtul_ksiazki);  
        em.getTransaction().commit();  
    } finally {  
        em.close();  
        return false; }  
}
```



# Usuwanie tytułów z bazy danych

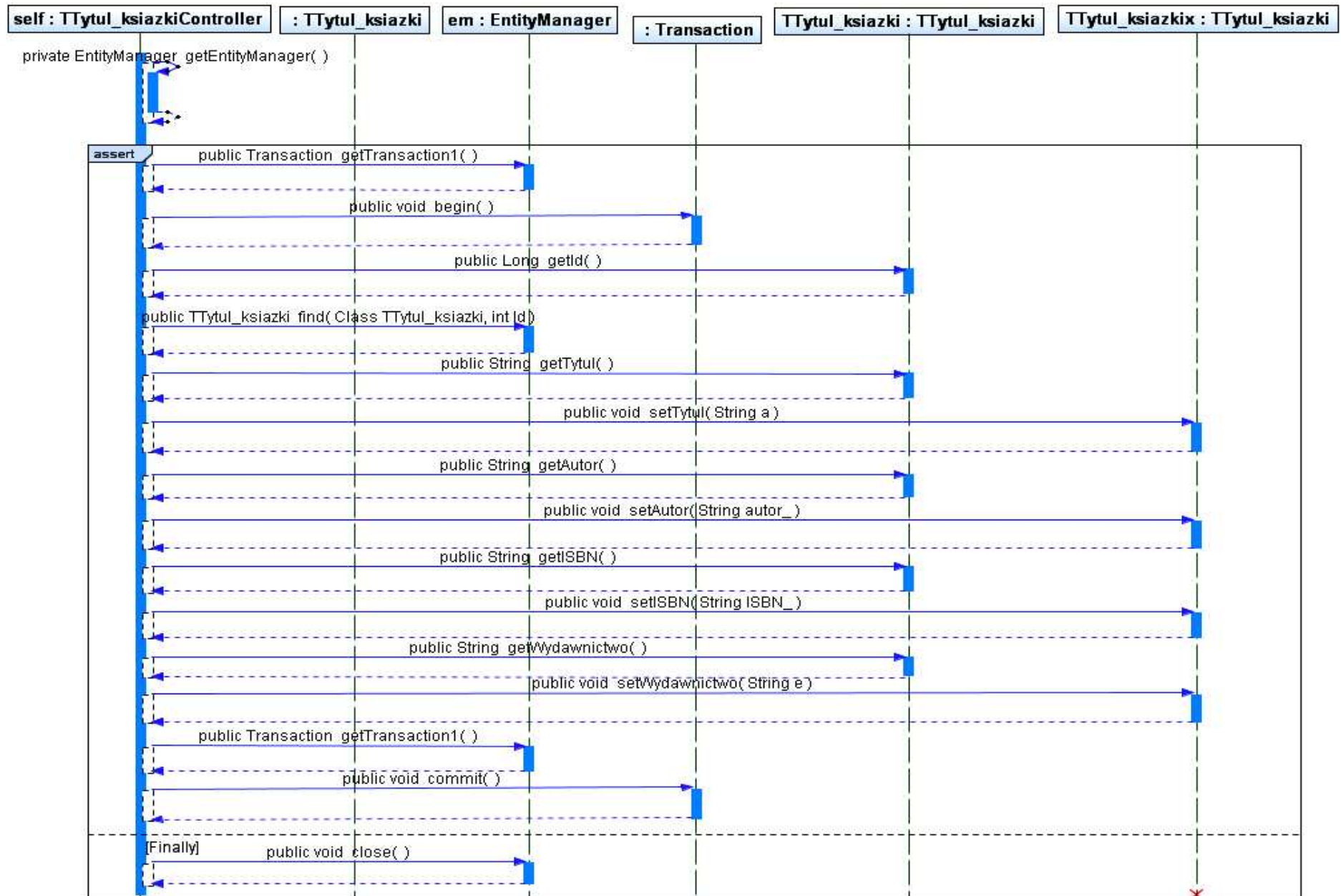
**public boolean removeTTytul\_ksiazkis(TTytul\_ksiazki TTytul\_ksiazki)**



```
public boolean removeTTytul_ksiazkis(
    TTYtul_ksiazki TTYtul_ksiazki) {
    EntityManager em = getEntityManager();
try {
        em.getTransaction().begin();
        TTYtul_ksiazki TTYtul_ksiazkix =
            em.find(TTYtul_ksiazki.class,
                TTYtul_ksiazki.getId());
        em.remove(TTYtul_ksiazkix);
        em.getTransaction().commit();
    } finally {
        em.close();
        return false; }
}
```

# Modyfikacja danych w bazie danych

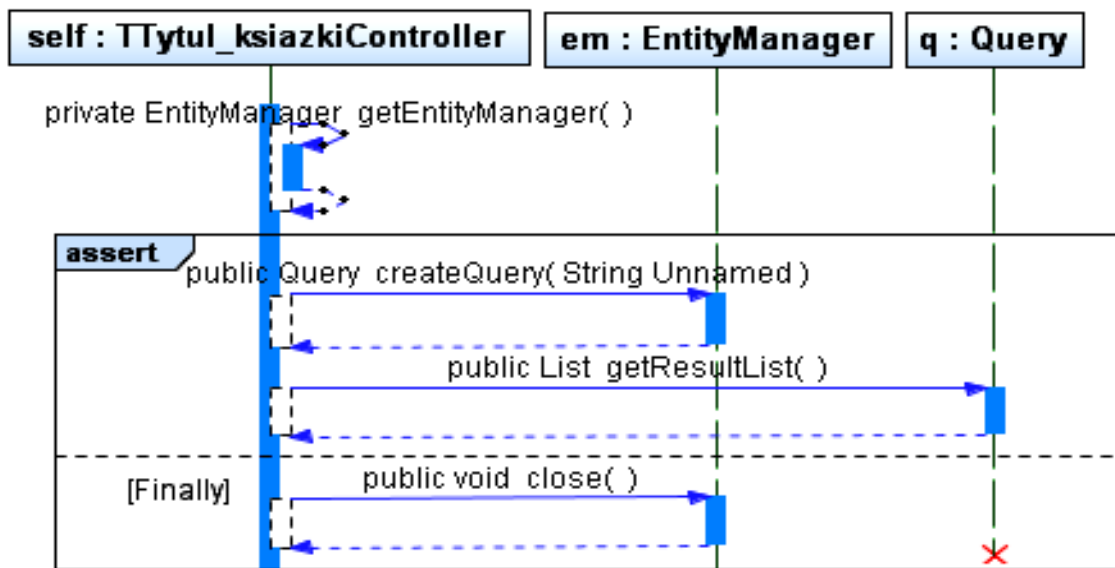
public boolean updateTTytul\_ksiazkis(TTytul\_ksiazki TTytul\_ksiazki)



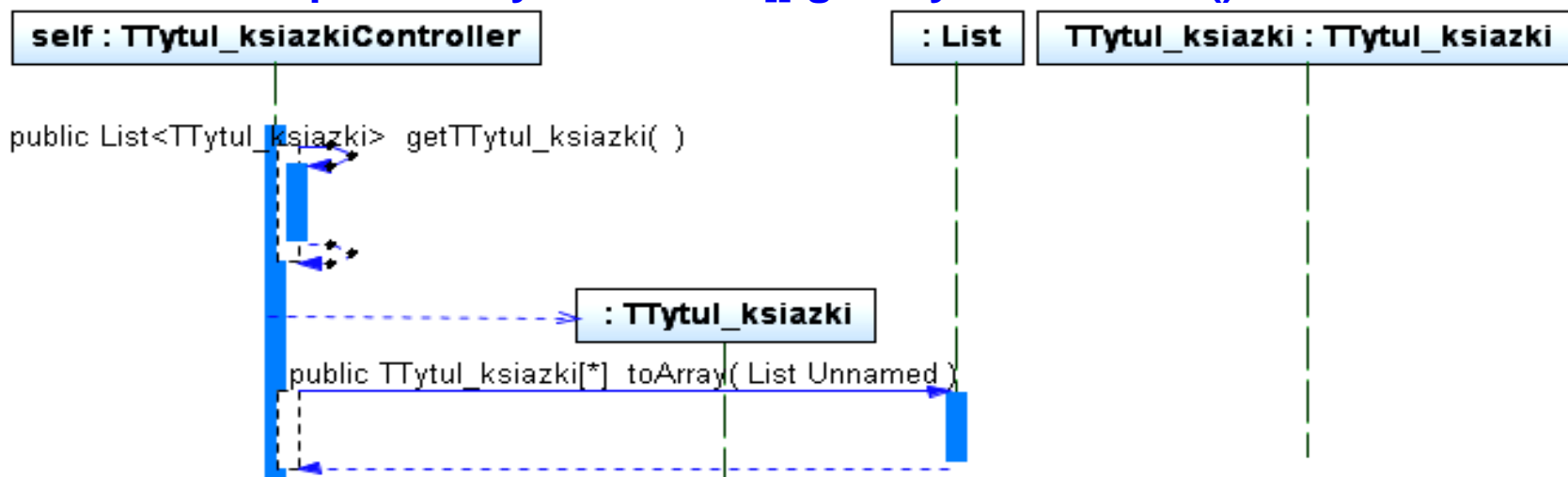
```
public boolean updateTTytul_ksiazkis(TTytul_ksiazki TTytul_ksiazki)
{ EntityManager em = getEntityManager();
try
{
    em.getTransaction().begin();
    TTytul_ksiazki TTytul_ksiazkix =
        em.find(TTytul_ksiazki.class, TTytul_ksiazki.getId());
    TTytul_ksiazkix.setTytul(TTytul_ksiazki.getTytul());
    TTytul_ksiazkix.setAutor(TTytul_ksiazki.getAutor());
    TTytul_ksiazkix.setISBN(TTytul_ksiazki.getISBN());
    TTytul_ksiazkix.setWydawnictwo(TTytul_ksiazki.getWydawnictwo());
    em.getTransaction().commit();
}
finally
{
    em.close();
    return false; }
}
```

# Odczytanie tytułów z bazy danych w postaci listy i tablicy TTytul\_książki (wzorzec typu Transfer Object – TTytul\_książki[0..\*])

**public List<TTytul\_książki> getTTytul\_książki()**



**public TTytul\_książki[] getTTytul\_książkis()**



```
public TTytul_ksiazki[] getTTytul_ksiazkis()
{
    return (TTytul_ksiazki[]) getTTytul_ksiazki().toArray(
                                                new TTytul_ksiazki[0]);
}
```

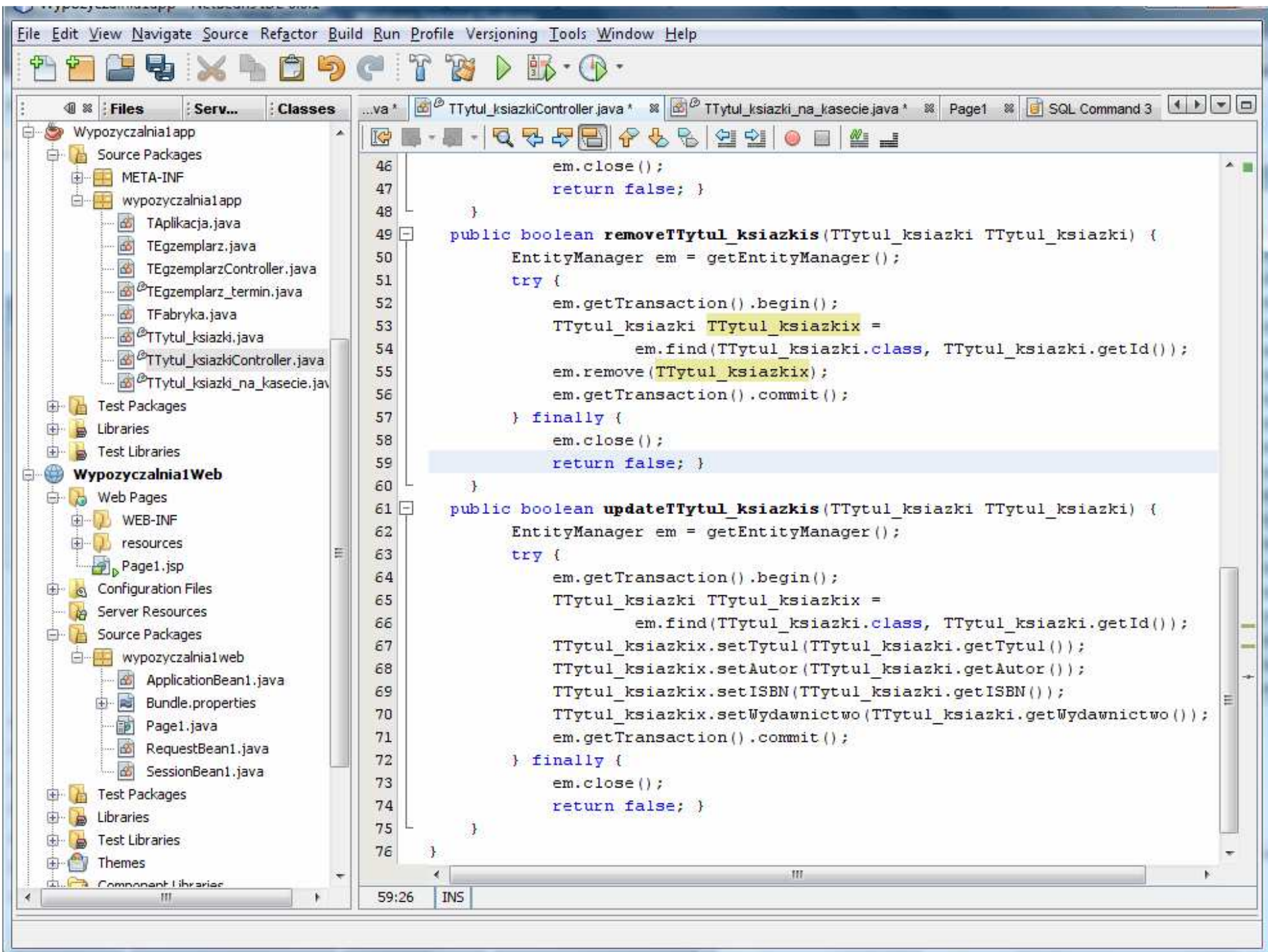
---

```
public List<TTytul_ksiazki> getTTytul_ksiazki()
{
    EntityManager em = getEntityManager();
    try {
        javax.persistence.Query q =
            em.createQuery("select c from TTytul_ksiazki as c");
        return q.getResultList();
    } finally {
        em.close();
    }
}
```



## Klasa warstwy integracji (szablon) – dla każdej klasy typu **Entity** (tutaj dla **TTytul\_książki**)

```
17 public class TTYtul_książkiController {
18     private EntityManagerFactory emf=null;
19     private EntityManager getEntityManager() {
20         if (emf == null) {
21             emf = Persistence.createEntityManagerFactory("Wypożyczalnia1appPU"); }
22         return emf.createEntityManager();
23     }
24
25     public TTYtul_książki[] getTTYtul_książkis() {
26         return (TTYtul_książki[])getTTYtul_książki().toArray(new TTYtul_książki[0]);
27     }
28
29     public List<TTYtul_książki> getTTYtul_książki() {
30         EntityManager em = getEntityManager();
31         try {
32             javax.persistence.Query q =
33                 em.createQuery("select c from TTYtul_książki as c");
34             return q.getResultList();
35         } finally {
36             em.close();
37         }
38     }
39     public boolean addTTYtul_książkis(TTYtul_książki TTYtul_książki) {
40         EntityManager em = getEntityManager();
41         try {
42             em.getTransaction().begin();
43             em.persist(TTYtul_książki);
44             em.getTransaction().commit();
45         } finally {
46             em.close();
47             return false; }
48     }
}
```



## 2.3 Klasa typu Entity

opis klas Entity na podstawie <http://java.sun.com/javaee/5/docs/tutorial/doc/>

- Musi posiadać adnotację `javax.persistence.Entity` (**@Entity**)
- Klasa musi mieć konstruktor typu **public** lub **protected**, bezparametrowy
- Klasa nie może być typu **final**
- Klasa musi być serializowana, stąd musi wystąpić w deklaracji klasy **implements Serializable**
- Klasy typu Entity mogą dziedziczyć po klasach zarówno typu Entity jak i zwykłej oraz zwykła klasa może dziedziczyć od klasy typu Entity.
- Dostęp do danych obiektu typu Entity może odbywać się tylko poprzez metody
- Musi mieć metody **hashCode()** i **equals(Object other)**
- Podstawowe adnotacje dla kluczy głównych (PK)

**@Id** oznacza adnotację dla klucza głównego.

Adnotacja **@GeneratedValue(strategy=GenerationType.<...>)** oznacza strategię tworzenia wartości kluczy głównych, zależną od systemu baz danych (`javax.persistence.GeneratedValue`, `javax.persistence.GenerationType`).

Adnotacja **@GeneratedValue(strategy = GenerationType.IDENTITY)** oznacza odpowiedzialność za generowanie wartości kluczy głównych przez bazę danych.

Adnotacja **@GeneratedValue(strategy = GenerationType.AUTO)** pozwala na przejęcie odpowiedzialności za generowanie kluczy przez TopLink, który posługuje się pomocniczą tabelą przechowującą wartości potrzebne do generowania kluczy)



# Zalecenia dla klas typu Entity, które mogą być utrwalane w bazach danych (1)

- Typy danych pól: typy nieobiektove, java.lang.String, java.math.BigInteger, java.math.BigDecimal, java.util.Date, java.util.Calendar, java.sql.Date, java.sql.Time, java.sql.TimeStamp, user-defined serializable types, byte[], Byte[], char[], Character[], typu wyliczeniowe, kolekcje, klasy wewnętrzne
- Pomijanie pól przy utrwalaniu realizuje się za pomocą javax.persistence.Transient - **@Transient**
- Można stosować następujące interfejsy kolekcji: **java.util.Collection**, **java.util.Set**, **java.util.List**, **java.util.Map** oraz korzystać z parametryzacji kolekcji np.  
set<**PhoneNumber**> getPhoneNumbers() {}  
void setPhoneNumbers(Set<**PhoneNumber**>) {}
- Klucze główne: typy nieobiektove (adnotacja **@Id** - javax.persistence.Id ), typy złożone: java.lang.String, java.util.Date, java.sql.Date – adnotacje javax.persistence.EmbeddedId and javax.persistence.IdClass (**@EmbeddedId** oraz **@IdClass**)
  - Muszą być publiczne
  - Klasa klucza głównego musi mieć domyślny konstruktor,
  - Klasa głównego klucza musi mieć metody hashCode() and equals(Object other)
  - Klasa głównego klucza musi być serializowana

## Zalecenia dla klas typu Entity, które mogą być utrwalane w bazach danych (2)

Zaznaczone metody muszą wystąpić w klasach kluczy głównych oraz w klasach typu Entity

```
public final class LineltemKey implements Serializable {
```

```
    public Integer orderId;
```

```
    public int itemId;
```

```
    public LineltemKey() {}
```

```
    public LineltemKey(Integer orderId, int itemId)
```

```
    {    this.orderId = orderId;    this.itemId = itemId; }
```

```
    public boolean equals(Object otherOb)
```

```
    {    if (this == otherOb) {        return true;    }
```

```
        if (!(otherOb instanceof LineltemKey)) {        return false;    }
```

```
        LineltemKey other = (LineltemKey) otherOb;
```

```
        return ( (orderId == null ? other.orderId==null :
```

```
            orderId.equals (other.orderId) ) && (itemId == other.itemId) ); }
```

```
    public int hashCode()
```

```
    {    return ( (orderId==null ? 0 : orderId.hashCode()) ^ ((int) itemId) ); }
```

```
    public String toString()
```

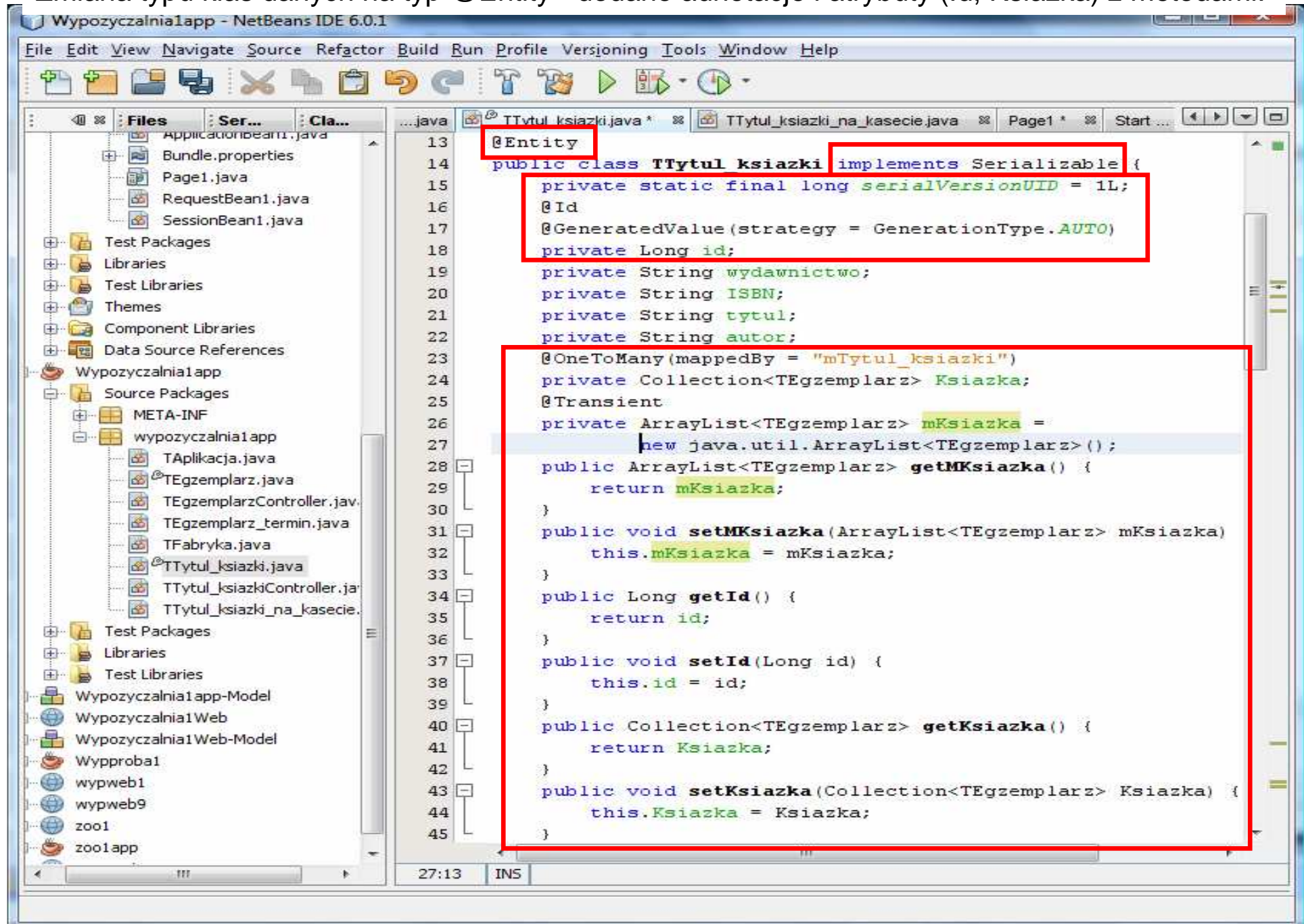
```
    {    return "" + orderId + "-" + itemId; }
```

## Zalecenia dla klas typu Entity, które mogą być utrwalane w bazach danych (3)

- Typy relacji (liczność)
  - One-to-one: adnotacja `javax.persistence.OneToOne` - **@OneToOne**
  - *One-to-many*: adnotacja `javax.persistence.OneToOne` - **@OneToMany**
  - Many-to-one: adnotacja `javax.persistence.ManyToOne` - **@ManyToOne**
  - Many-to-many: adnotacja `javax.persistence.ManyToMany` - **@ManyToMany**
- Kierunek relacji
  - Jednokierunkowa – posiada implementację po jednej stronie relacji (zapytanie może być realizowane od właściciela, który zna inny obiekt )
  - Dwukierunkowa musi spełniać wymagania:
    - Przeciwna strona relacji musi posiadać element typu **mappedBy** o adnotacji **@OneToOne**, **@OneToMany**, lub **@ManyToMany**, w zależności od definicji drugiej strony relacji. Element **mappedBy** oznacza pole właściciela relacji (drugiej strony relacji).
    - Druga strona relacji many-to-many lub many-to-one nie może posiadać elementu **mappedBy**.
    - Dla relacji one-to-one, strona właściciela odpowiada stronie z kluczem obcym.
    - Dla relacji many-to-many każda strona relacji jest właścicielem drugiej strony.
    - Zapytanie mogą być realizowane w obu kierunkach
- Kaskadowe usuwanie wszystkich obiektów powiązanych przez element **mappedBy** np.

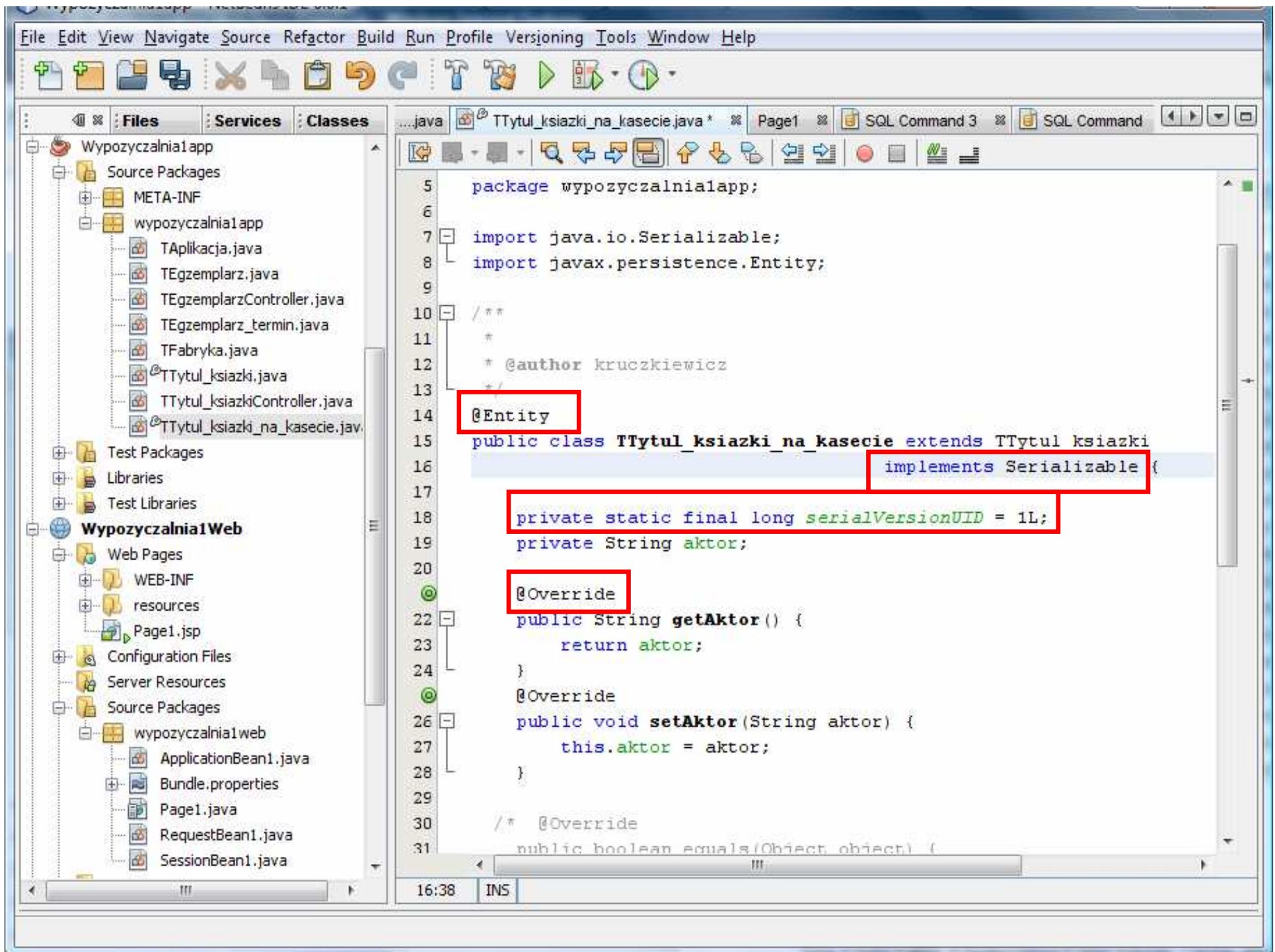
```
@OneToMany(cascade=REMOVE, mappedBy="customer ")
public Set<Order> getOrders() { return orders; }
```

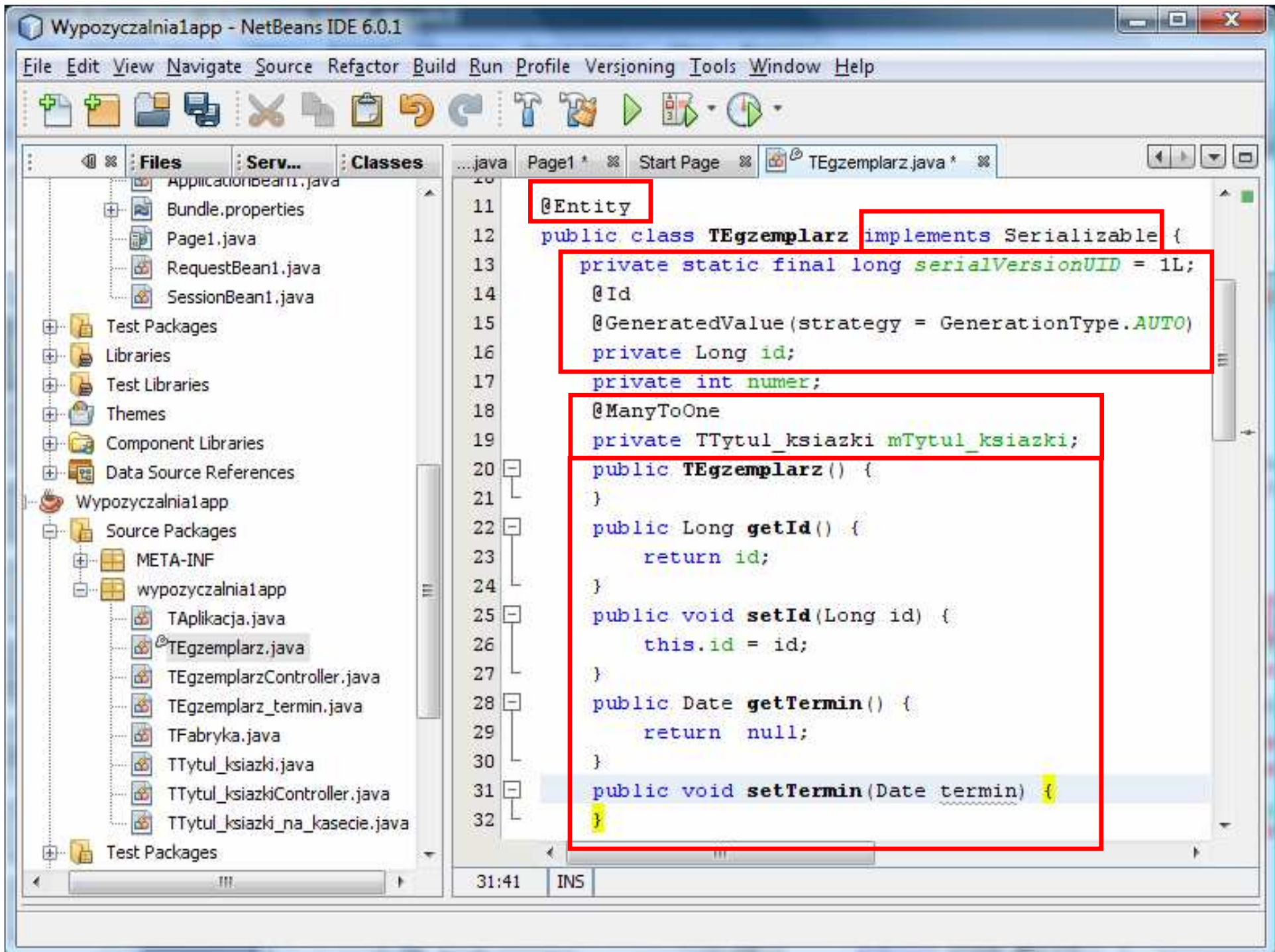
Zmiana typu klas danych na typ @Entity - dodano adnotacje i atrybuty (Id, Ksiazka) z metodami.



```
13 @Entity
14 public class TTYtul_książki implements Serializable {
15     private static final long serialVersionUID = 1L;
16     @Id
17     @GeneratedValue(strategy = GenerationType.AUTO)
18     private Long id;
19     private String wydawnictwo;
20     private String ISBN;
21     private String tytuł;
22     private String autor;
23
24     @OneToMany(mappedBy = "mTytuł_książki")
25     private Collection<TEgzemplarz> Ksiazka;
26     @Transient
27     private ArrayList<TEgzemplarz> mKsiazka =
28         new java.util.ArrayList<TEgzemplarz>();
29     public ArrayList<TEgzemplarz> getMKsiazka() {
30         return mKsiazka;
31     }
32     public void setMKsiazka(ArrayList<TEgzemplarz> mKsiazka) {
33         this.mKsiazka = mKsiazka;
34     }
35     public Long getId() {
36         return id;
37     }
38     public void setId(Long id) {
39         this.id = id;
40     }
41     public Collection<TEgzemplarz> getKsiazka() {
42         return Ksiazka;
43     }
44     public void setKsiazka(Collection<TEgzemplarz> Ksiazka) {
45         this.Ksiazka = Ksiazka;
46     }
47 }
```







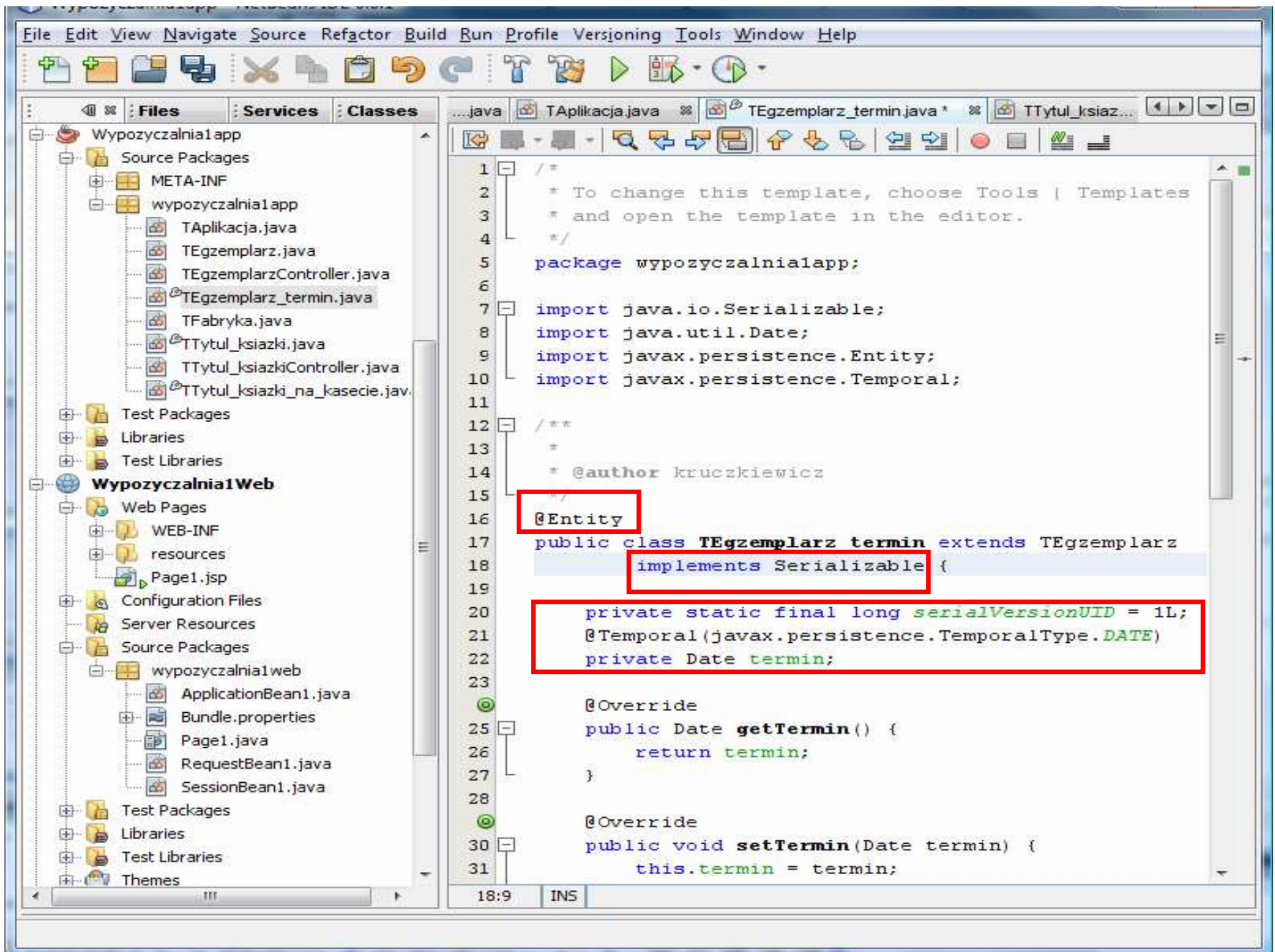


Zmiana typu klas danych na typ @Entity – dodano adnotacje, nowy atrybut (Id) z metodami. **Należy zestandaryzować nazwy metod do składowej mTytul\_książki!**

```
11 @Entity
12 public class TEgzemplarz implements Serializable {
13     private static final long serialVersionUID = 1L;
14     @Id
15     @GeneratedValue(strategy = GenerationType.AUTO)
16     private Long id;
17     private int numer;
18     @ManyToOne
19     private TTYtul_książki mTytul_książki;
20
21     public TEgzemplarz() {
22     }
23     public Long getId() {
24         return id;
25     }
26     public void setId(Long id) {
27         this.id = id;
28     }
29     public TTYtul_książki getMTytul_książki() {
30         return mTytul_książki;
31     }
32     public void setMTytul_książki(TTYtul_książki mTytul_książki) {
33         this.mTytul_książki = mTytul_książki;
34     }
35     public Date getTermin() {
36         return null;
37     }
38     public void setTermin(Date termin) {
39     }
40 }
```

The screenshot shows an IDE window with the following components:

- Project Explorer:** Shows the project structure for 'Wypożyczalnia1app', including source packages (META-INF, wypożyczalnia1app) and test packages. Files include persistence.xml, TAplikacja.java, TEgzemplarz.java, TEgzemplarzController.java, TEgzemplarz\_termin.java, TFabryka.java, TTYtul\_książki.java, TTYtul\_książkiController.java, and TTYtul\_książki\_na\_kasecie.java.
- Navigator:** Shows the 'Members View' for the 'TEgzemplarz' class, listing methods such as getId(), setId(), getMTytul\_książki(), setMTytul\_książki(), getTermin(), and setTermin().
- Code Editor:** Displays the source code for TEgzemplarz.java. Red boxes highlight the @Entity annotation, the Serializable interface, the @Id, @GeneratedValue, and @ManyToOne annotations, and the getter and setter methods for the mTytul\_książki attribute.





# Wstawienie do modułu typu Persistence Unit wszystkich klas typu Entity

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays a project named 'Wypożyczalnia1app' with a sub-package 'wypożyczalnia1app' containing several Java classes. The 'persistence.xml' file is highlighted. The main editor area shows the 'Persistence Units' configuration for 'Wypożyczalnia1appPU'. The configuration includes the Persistence Unit Name, Persistence Library (TopLink), and JDBC Connection. The Table Generation Strategy is set to 'Create'. A red box highlights the 'Include All Entity Classes in "Wypożyczalnia1app" Module' checkbox, which is currently unchecked. Below this, the 'Include Entity Classes' list contains the following classes: 'wypożyczalnia1app.TTytul\_ksiazki', 'wypożyczalnia1app.TEgzemplarz', 'wypożyczalnia1app.TEgzemplarz\_termin', and 'wypożyczalnia1app.TTytul\_ksiazki\_na\_kasecie'. There are 'Add Class...' and 'Remove' buttons next to the list.

Files: Sprzedaz1, Sprzedaz1-Model, WebApplication1, WebApplication2, Webwyp1, wypapp1, Wypożyczalnia1\_, Wypożyczalnia1app, Source Packages, META-INF, persistence.xml, wypożyczalnia1app, TAplikacja.java, TEgzemplarz.java, TEgzemplarzController.java, @TEgzemplarz\_termin.java, TFabryka.java, @TTytul\_ksiazki.java, @TTytul\_ksiazkiController.java, @TTytul\_ksiazki\_na\_kasecie.java, Test Packages, Libraries, Test Libraries, Wypożyczalnia1Web, Web Pages, WEB-INF, resources, Page1.jsp, Configuration Files

Design XML Wypożyczalnia1appPU

Persistence Units

Wypożyczalnia1appPU

Persistence Unit Name: Wypożyczalnia1appPU

Persistence Library: TopLink

JDBC Connection: jdbc:derby://localhost:1527/katalog [kruk on KRUK]

Table Generation Strategy:  Create  Drop and Create  None

Include All Entity Classes in "Wypożyczalnia1app" Module

Include Entity Classes:

- wypożyczalnia1app.TTytul\_ksiazki
- wypożyczalnia1app.TEgzemplarz
- wypożyczalnia1app.TEgzemplarz\_termin
- wypożyczalnia1app.TTytul\_ksiazki\_na\_kasecie

Add Class... Remove

Wypożyczalnia1appPU2

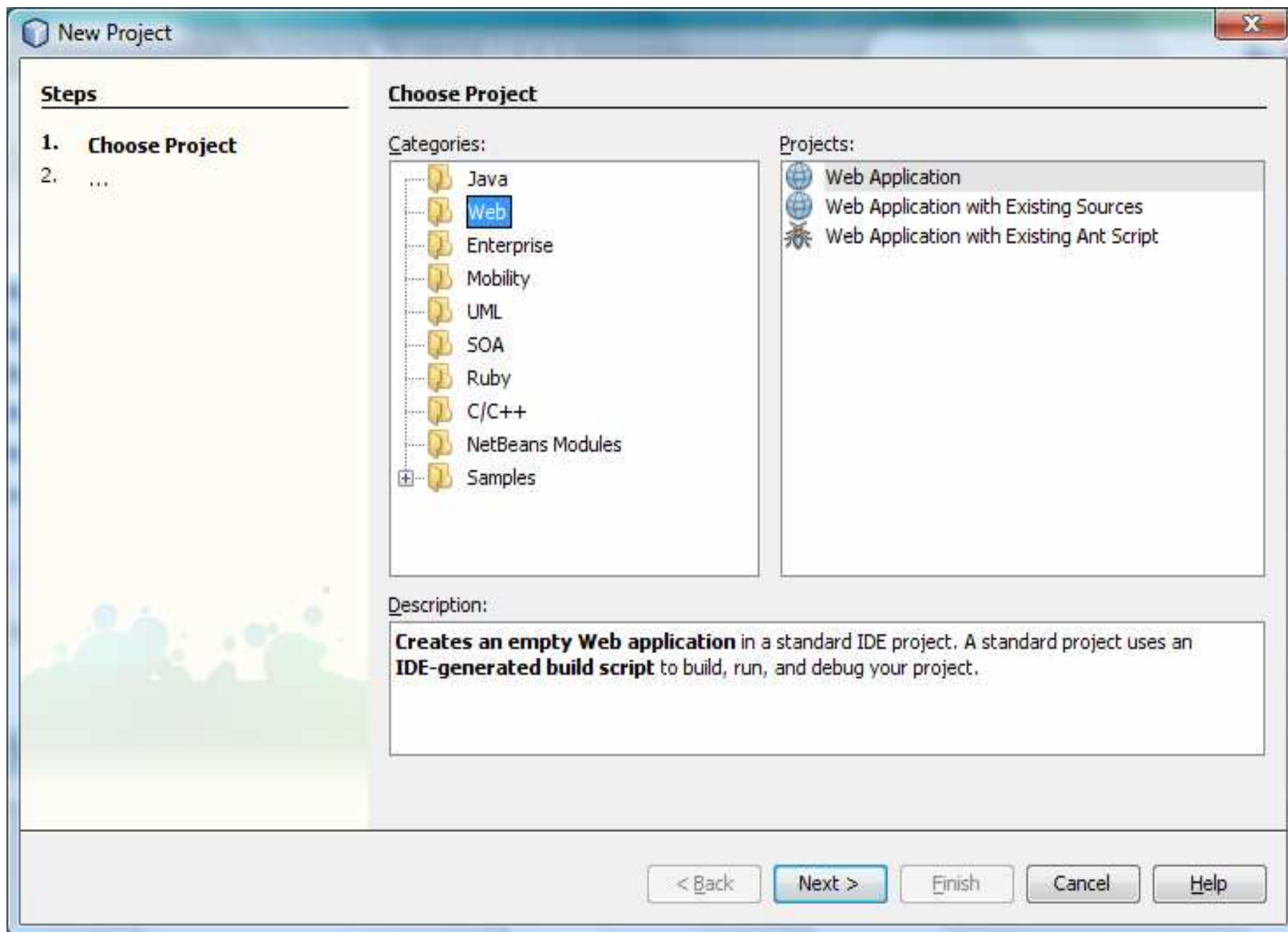
# 3. Tworzenie warstwy klienta, prezentacji

Integrowanie warstwy prezentacji i klienta wykonane w technologii **Java Server Faces**

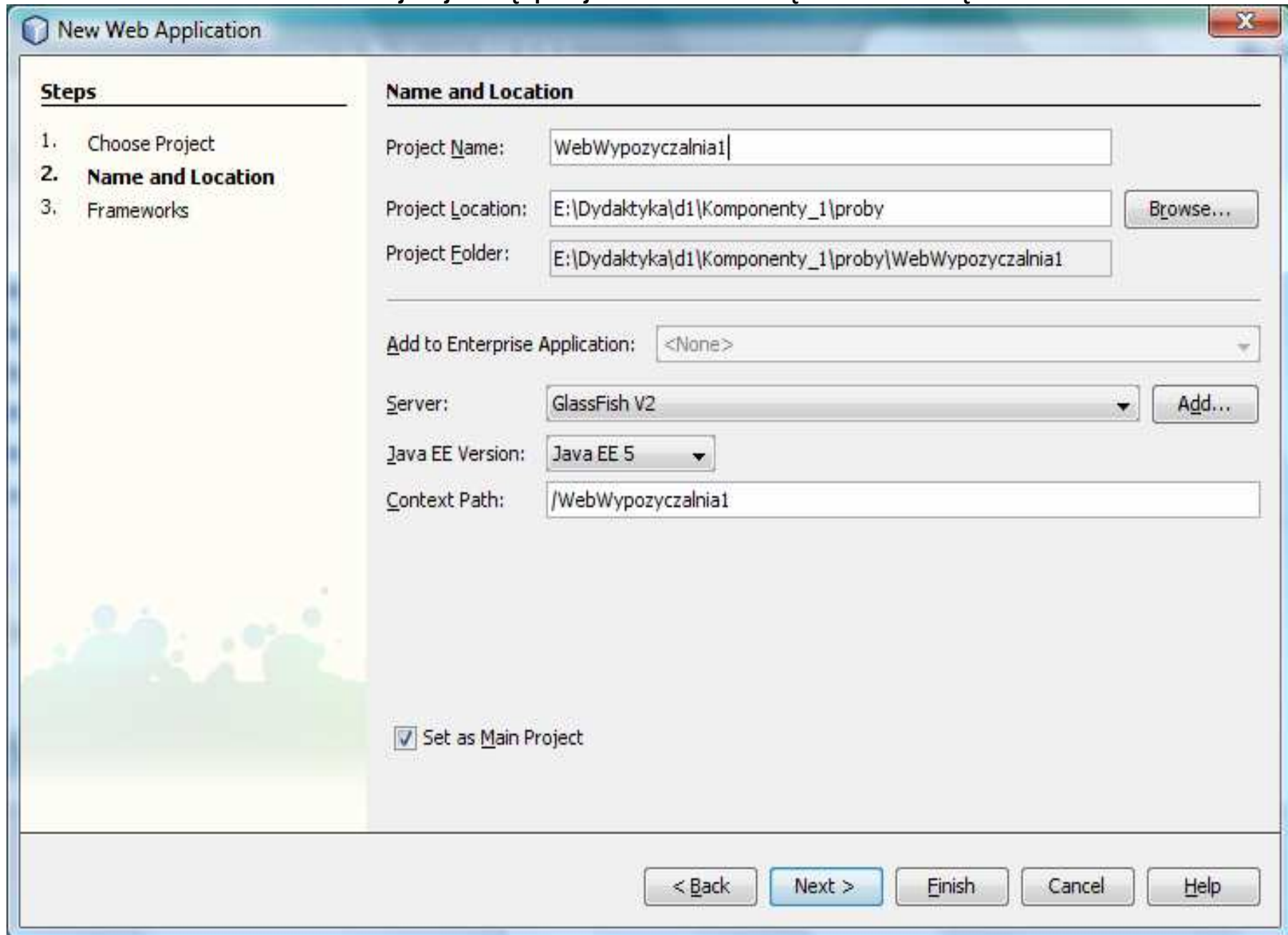
- z projektem warstwy biznesowej zawierającej model obiektowy
- oraz warstwą integracji



# Tworzenie projektu kategorii Web typu Web Application



Projekt typu WebApplication tworzony w tym samym katalogu, w którym znajduje się projekt z warstwą biznesową

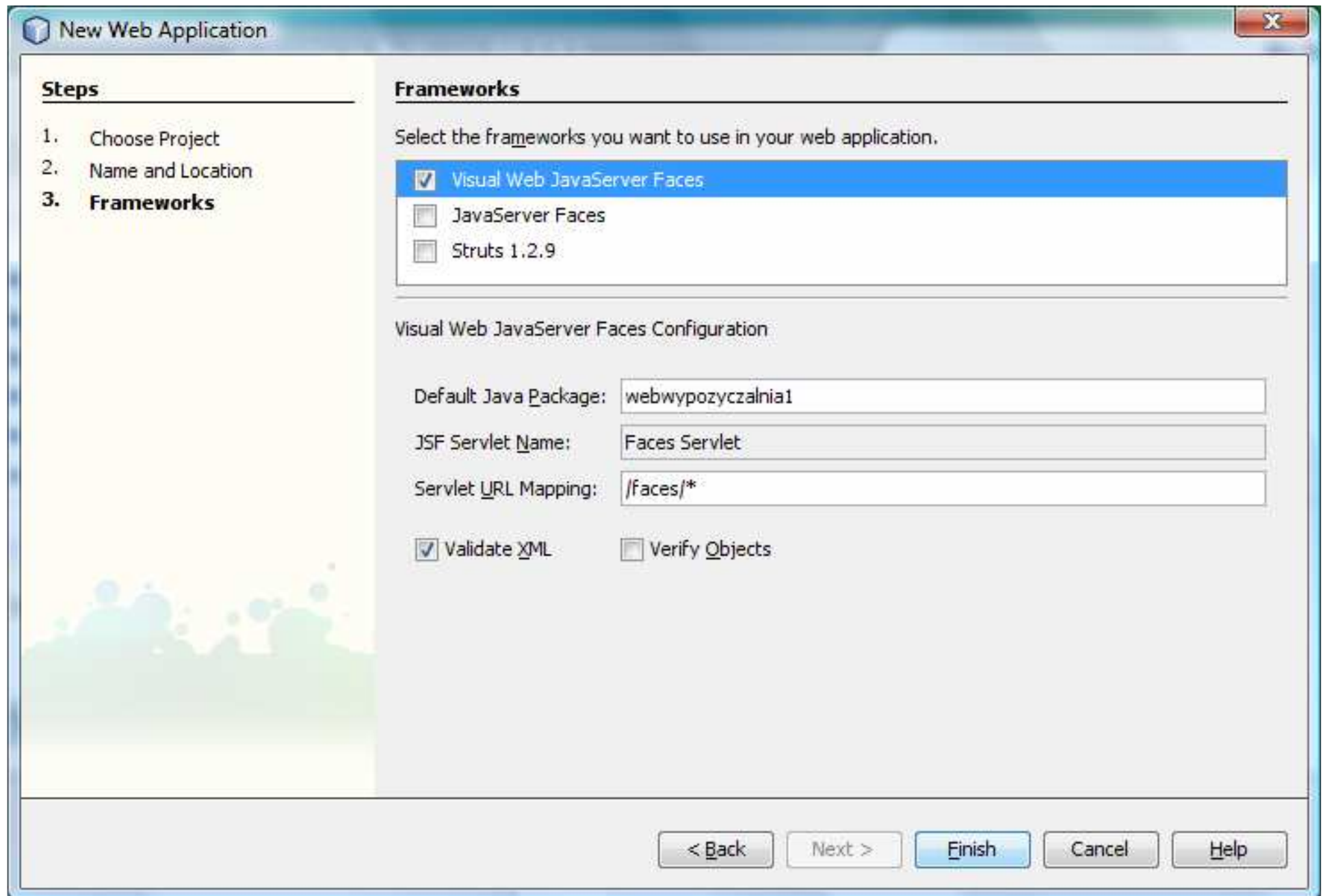


The screenshot shows the 'New Web Application' wizard window. The title bar reads 'New Web Application'. On the left, a 'Steps' pane lists: 1. Choose Project, 2. **Name and Location**, and 3. Frameworks. The main area is titled 'Name and Location' and contains the following fields and controls:

- Project Name:** WebWypożyczalnia1
- Project Location:** E:\Dydaktyka\d1\Komponenty\_1\proby (with a 'Browse...' button)
- Project Folder:** E:\Dydaktyka\d1\Komponenty\_1\proby\WebWypożyczalnia1
- Add to Enterprise Application:** <None>
- Server:** GlassFish V2 (with an 'Add...' button)
- Java EE Version:** Java EE 5
- Context Path:** /WebWypożyczalnia1
- Set as Main Project

At the bottom, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

# Projekt powinien być oparty na Visual Web JavaServer Faces



**New Web Application**

**Steps**

1. Choose Project
2. Name and Location
3. **Frameworks**

**Frameworks**

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

- Visual Web JavaServer Faces
- JavaServer Faces
- Struts 1.2.9

Visual Web JavaServer Faces Configuration

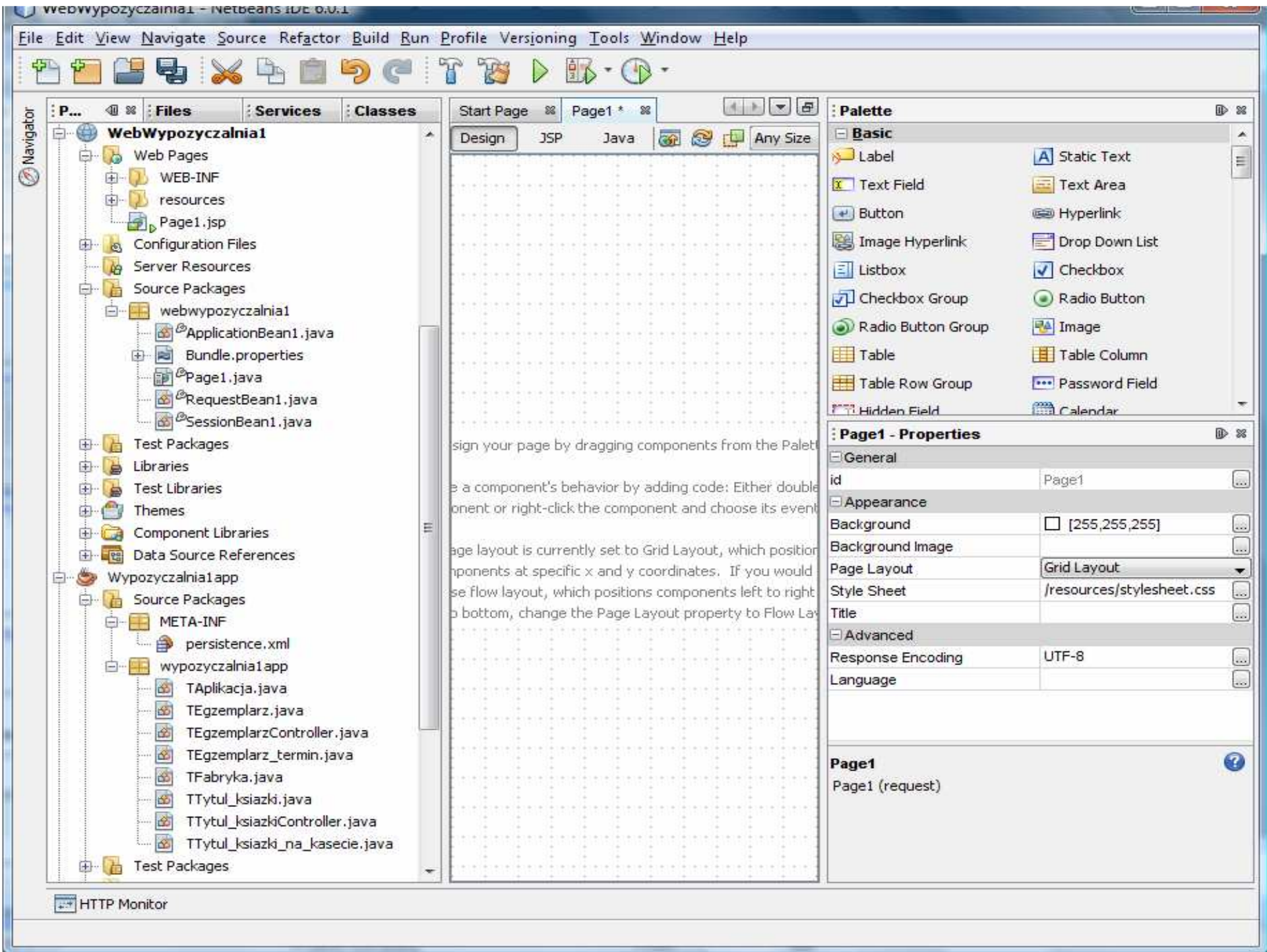
Default Java Package:

JSF Servlet Name:

Servlet URL Mapping:

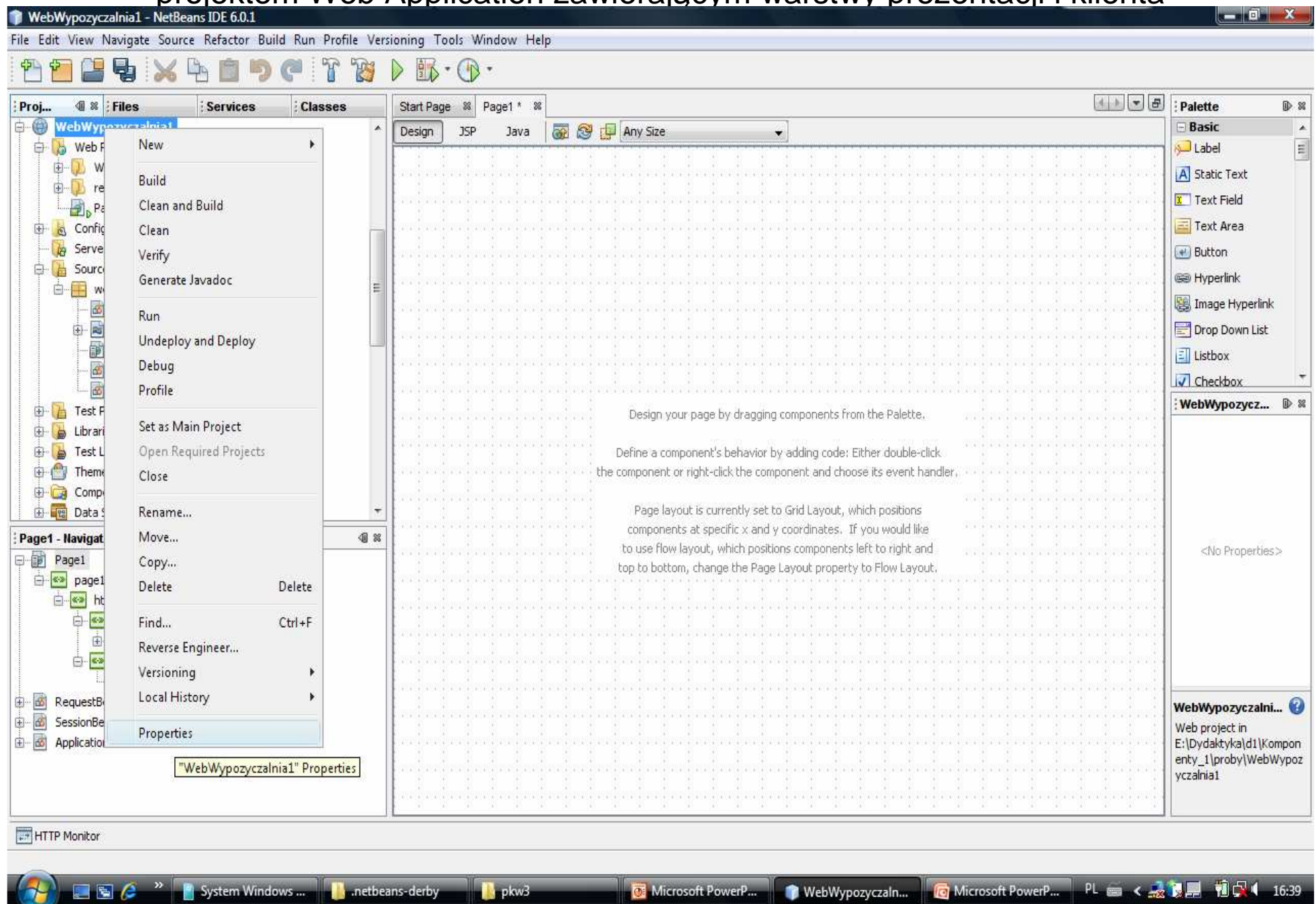
Validate XML     Verify Objects

< Back    Next >    **Finish**    Cancel    Help

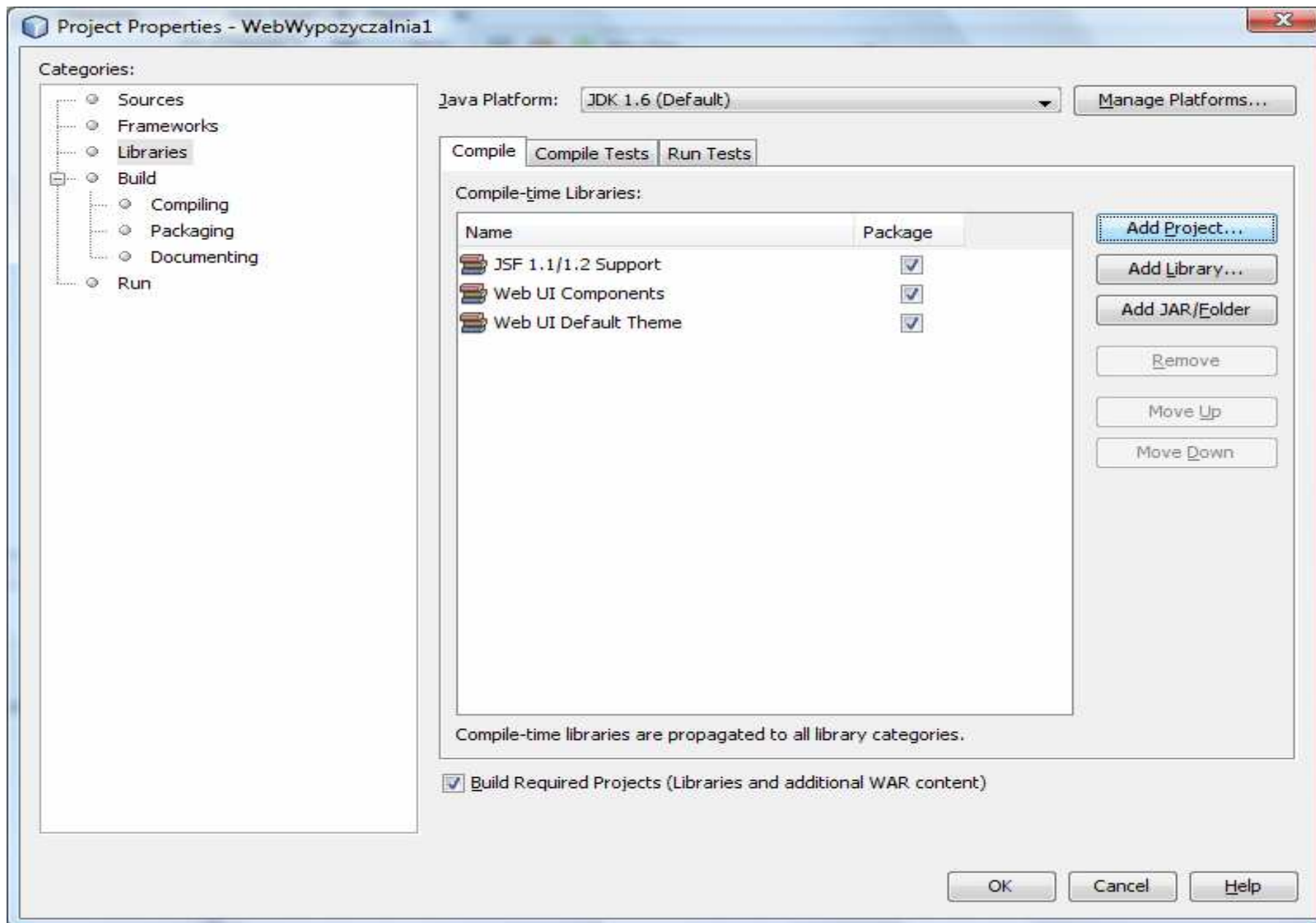




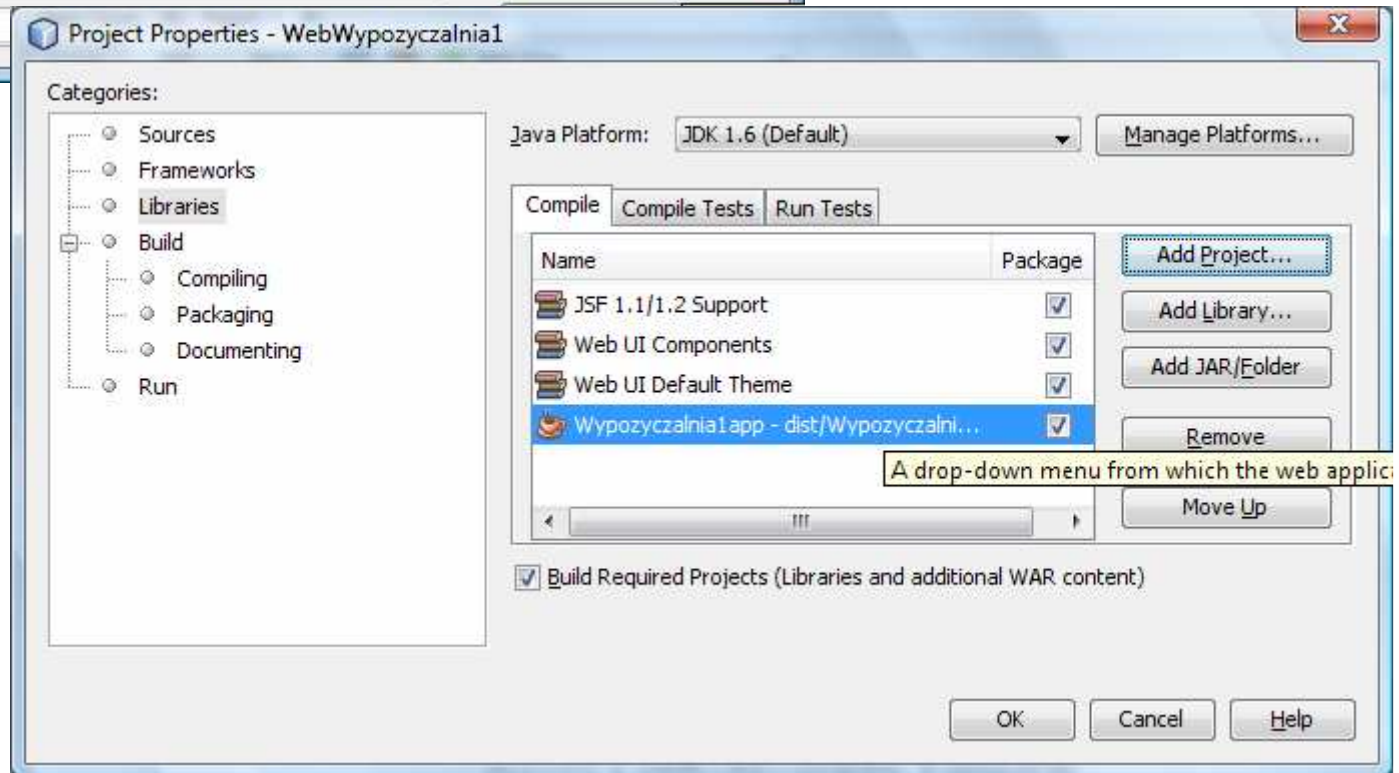
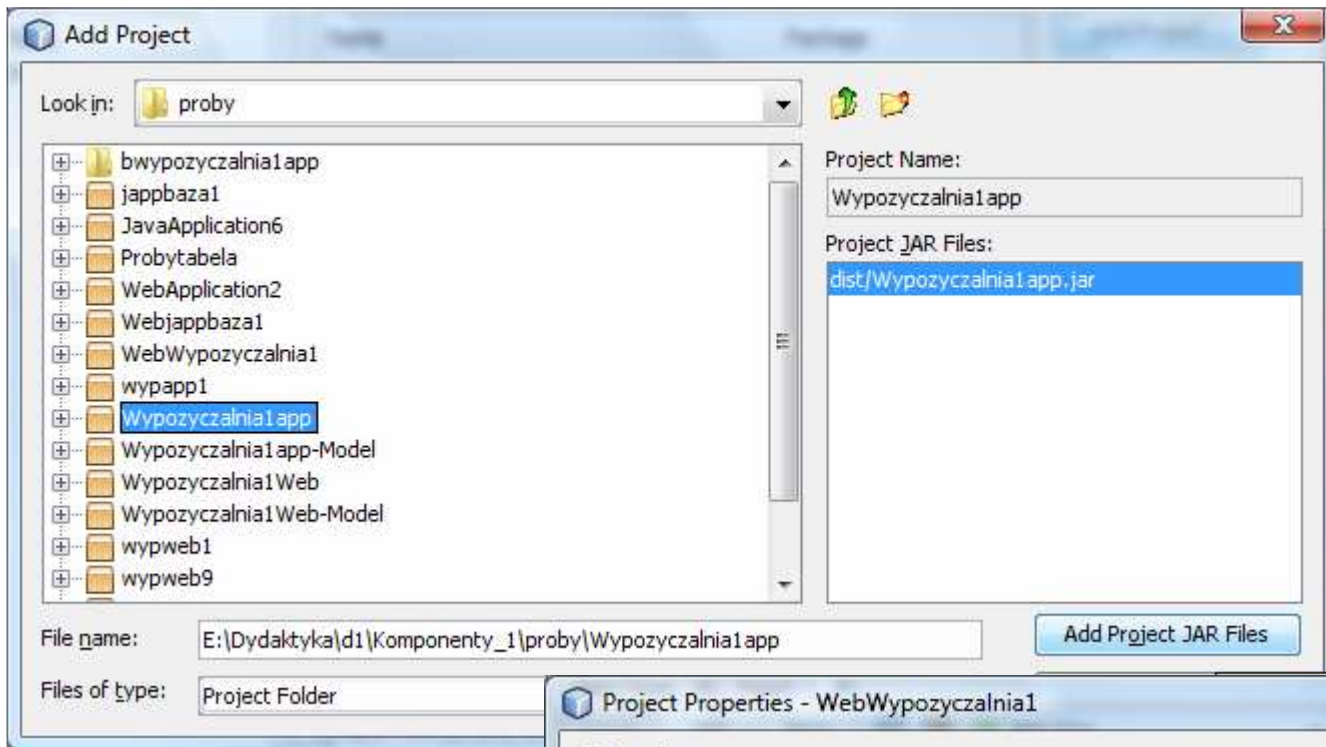
# Łączenie projektu typu JavaApplication zawierającym warstwę biznesową i integracji z projektem Web Application zawierającym warstwy prezentacji i klienta



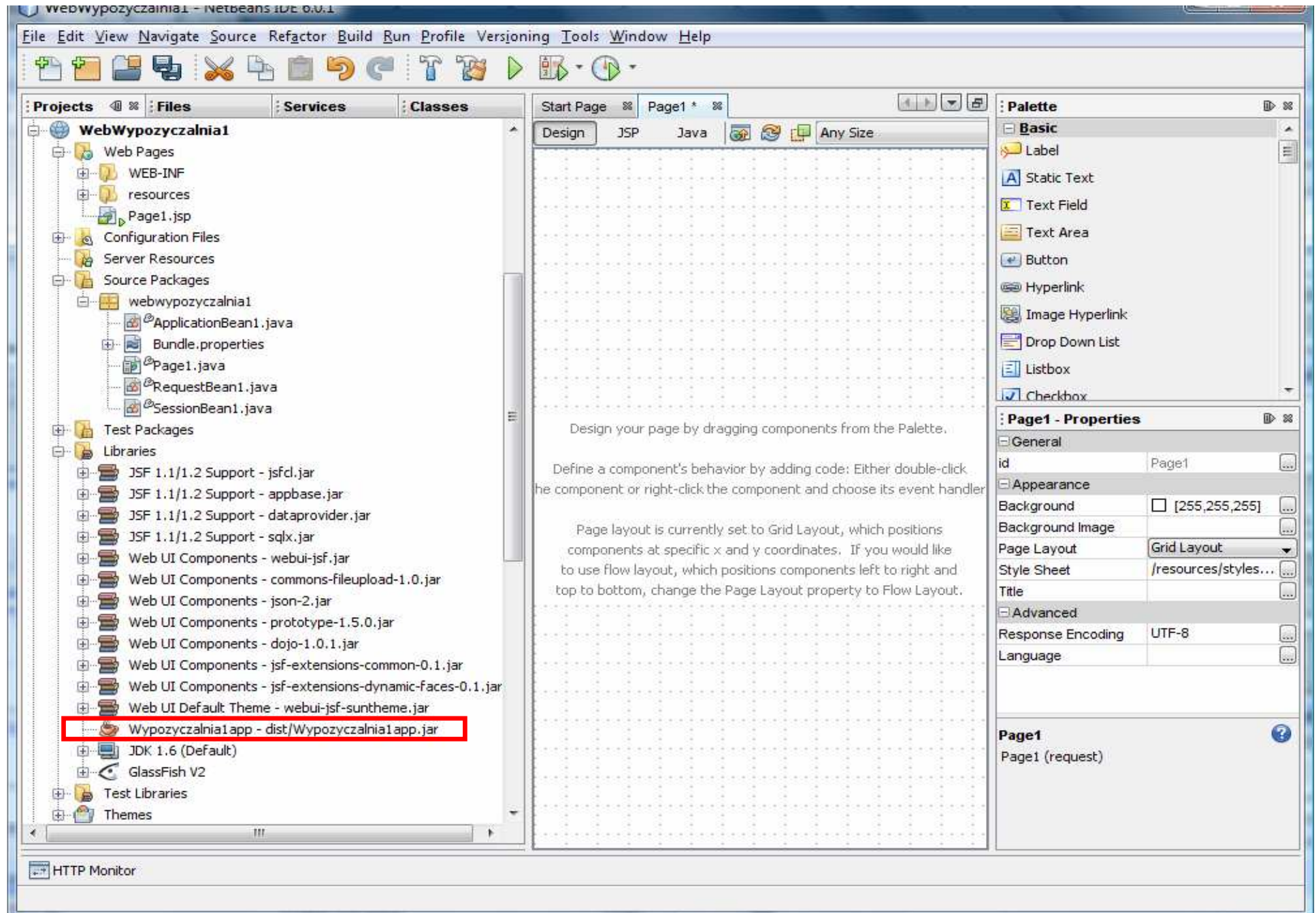
Wybór w okienku **Properties** projektu **Web Application** opcji **Libraries** oraz **Add Project**







# Połączone projekty – projekt Web Application korzysta z klas projektu Java Application



Wstawienie do warstwy prezentacji (**Page1.jsp**) komponentu **Table** w celu wyświetlania w tabeli danych typu **TTytul\_książki** oraz **TTytul\_książki\_na\_kasce** z bazy danych

The screenshot shows the NetBeans IDE interface. The main workspace displays a design view of a web page titled "Page1". A table component is placed on the page, showing a grid with 5 rows and 3 columns. The columns are labeled "column1", "column2", and "column3". The rows are labeled "row1\_column1" through "row5\_column3".

The left sidebar contains the "Projects" view, showing the project structure for "WebWypożyczalnia1". The "Files" view shows the project files, including "Page1.jsp". The "Classes" view shows the source code files, including "ApplicationBean1.java", "Page1.java", "RequestBean1.java", and "SessionBean1.java".

The bottom-left sidebar shows the "table1 - Navigator" view, which displays the component tree for the page. The tree structure is as follows:

- Page1
  - page1
    - html1
      - head1
      - body1
        - form1
          - table1
  - defaultTableDataProvider
  - RequestBean1
  - SessionBean1
  - ApplicationBean1

The right sidebar contains the "Properties" view, which shows the properties of the selected table component. The "General" tab is active, showing properties such as "id", "appearance", "augmentTi", "cellPadding", "cellSpacing", "clearSortBu", "deselectMut", "deselectSin", "footerText", "itemsText", "lite", and "paginateBut". The "table1" component is selected, and its properties are displayed.



# Wpisanie kodu wywołującego kod warstwy integracji

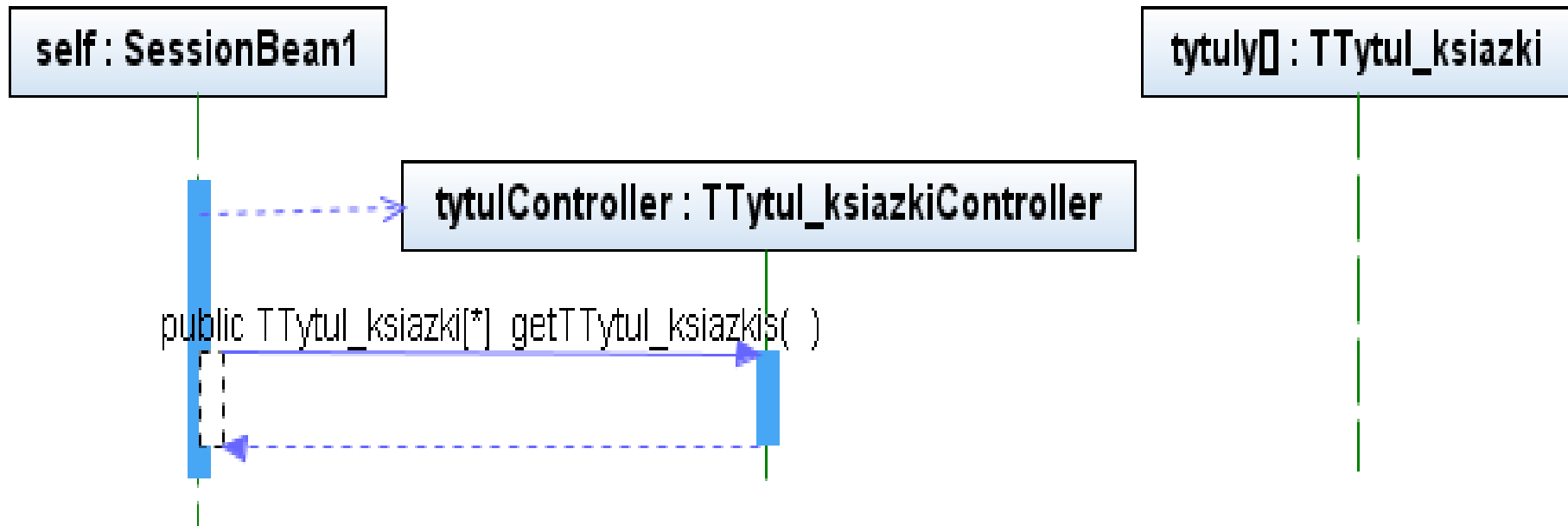
The screenshot shows the NetBeans IDE 6.0.1 interface. The left sidebar displays a project tree for 'WebWypożyczalnia1'. The main editor window shows the 'SessionBean1.java' file. The code is as follows:

```
55  */
56  @Override
57  public void init() {
58      // Perform initializations inherited from our superclass
59      super.init();
60      // Perform application initialization that must complete
61      // *before* managed components are initialized
62      // TODO - add your own initialization code here
63
64      Managed Component Initialization
65
66      // Perform application initialization that must complete
67      // *after* managed components are initialized
68      // TODO - add your own initialization code here
69
70  updateTytuls();
71  }
72  private TTytul_książki tytuly[];
73  public TTytul_książki[] getTytuly() {
74      return tytuly;
75  }
76
77  public void setTytuly(TTytul_książki[] tytuly) {
78      this.tytuly = tytuly;
79  }
80
81  public void updateTytuls() {
82      TTytul_książkiController tytulController = new TTytul_książkiController();
83      tytuly = tytulController.getTTytul_książkis();
84  }
```

The code from line 70 to 84 is highlighted with a red box. The 'Output' window at the bottom shows the following tabs: 'Java DB Database Process', 'GlassFish V2', and 'WebWypożyczalnia1 (run)'. The 'HTTP Monitor' window is also visible at the bottom.

Odczyt tytułów z bazy danych

**public void SessionBean1::updateTytuls()**



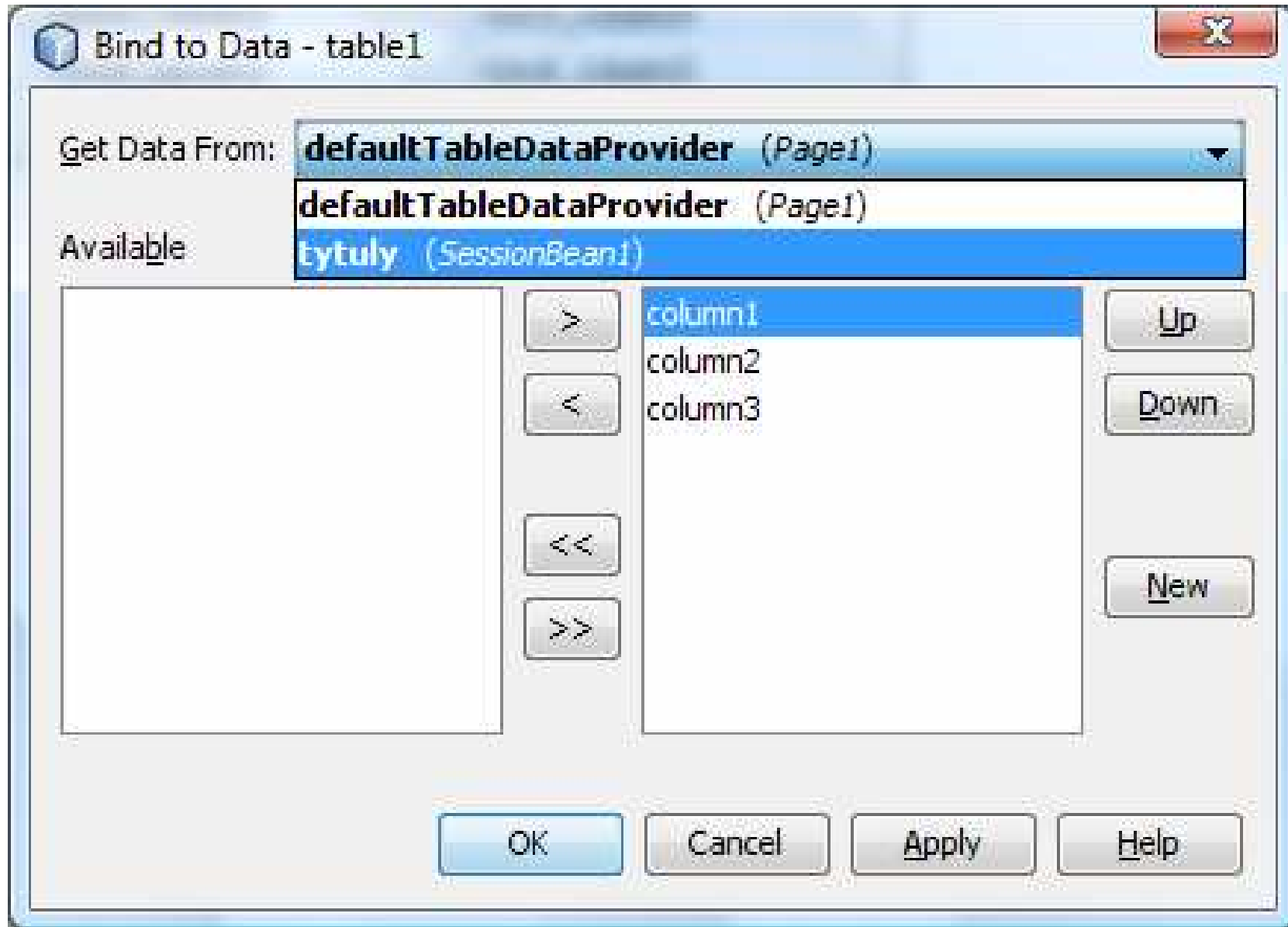
# Łączenie komponentu **Table** z tablicą **TTytul\_ksiazki ksiazki[]** – wybór **Bind to Data**

The screenshot shows the NetBeans IDE 6.0.1 interface. The main workspace displays a design view of a table component. The table has three columns: 'column1', 'column2', and 'column3', and five rows labeled 'row1\_column1' through 'row5\_column1'. A context menu is open over the table, with the 'Bind to Data...' option highlighted. The menu also includes options like 'Edit Java Source', 'Table Layout...', 'Add Row Group', 'Property Bindings...', 'Edit JSP Source', 'Edit Inline', 'Select Parent', 'Edit Event Handler', 'Set Initial Focus', 'Virtual Forms...', 'Snap to Grid', 'Bring to Front', 'Send to Back', 'Customize', 'Cut', 'Copy', 'Paste', 'Delete', 'Preview in Browser...', 'Refresh', 'Target Browser Window', and 'Page Navigation'. The left sidebar shows a project tree for 'WebWypożyczalnia1' and a 'table1 - Navigator' window. The bottom status bar shows the system tray with the time 17:10.

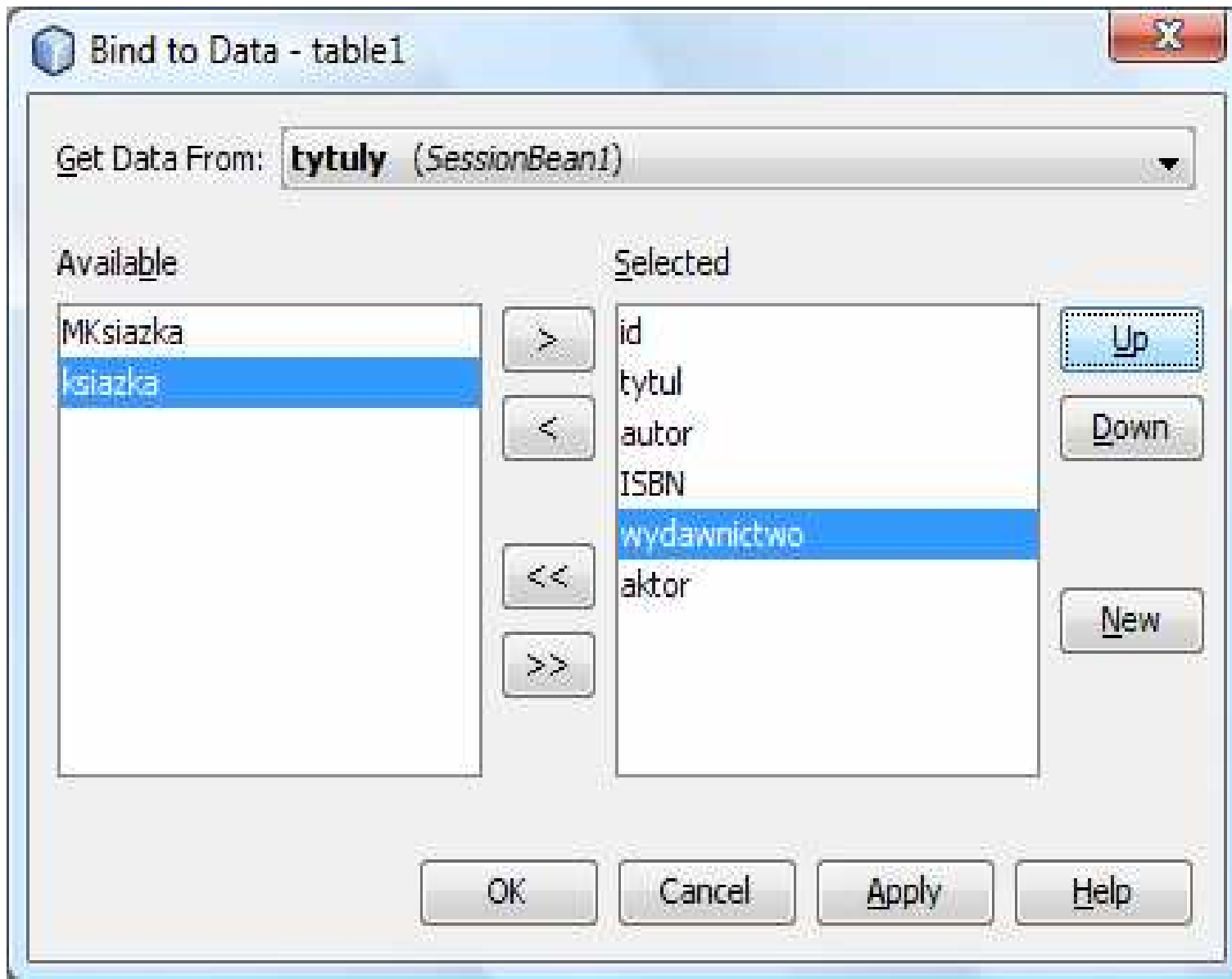
column1	column2	column3
row1_column1	row1_column2	row1_column3
row2_column1	row2_column2	row2_column3
row3_column1	row3_column2	row3_column3
row4_column1	row4_column2	row4_column3
row5_column1	row5_column2	row5_column3



Wybór tablicy **tytuly** w okienku **Bind to Data**



Wybór kolumn (pomijanie kolekcji książka mapującej relację **OneToMany** oraz **MKsiążka** wykorzystywanej przez warstwę biznesową do gromadzenia danych o książkach dla danego tytułu)



# Efekt operacji Bind to Data

The screenshot displays the NetBeans IDE 6.0.1 interface. The main workspace shows a table component in design mode. The table has the following data:

id	tytul	autor	ISBN	wydawnictwo	aktor
123	abc	abc	abc	abc	abc
123	abc	abc	abc	abc	abc
123	abc	abc	abc	abc	abc

The interface includes a Project Explorer on the left showing the project structure for 'WebWypożyczalnia1'. The 'table1 - Navigator' window shows the component hierarchy: Page1 > page1 > html1 > body1 > form1 > table1. The 'Output' window at the bottom shows 'HTTP Monitor'. The 'Palette' on the right lists various UI components like Page Separator, Tab Set, Tab, Page Fragment Box, Page Alert, Property Sheet, Property Sheet Section, Property, Form, and Composite. The 'No Properties' window is also visible.

## Uruchomienie aplikacji

The screenshot shows a Windows Internet Explorer browser window. The address bar displays the URL `http://localhost:8080/WebWypożyczalnia1/`. The menu bar includes "Plik", "Edycja", "Widok", "Ulubione", "Narzędzia", and "Pomoc". The toolbar contains various icons for home, search, and printing. A Norton security bar is visible with the message "Monitorowanie fałszywych witryn jest włączone". The main content area displays a table with the following structure:

Table					
id	tytuł	autor	ISBN	wydawnictwo	aktor
No items found.					

The status bar at the bottom shows "Goto", "Internet | Tryb chroniony: włączony", and a zoom level of "100%".

Wygenerowano tabele w bazie danych **katalog**, założonej wcześniej do utrwalania danych, kolumna **DTYPE** przechowuje informacje o typach mapowanych obiektów powiązanych dziedziczeniem do jednej tabeli

The screenshot shows a database management tool interface. On the left, a tree view displays the database schema for 'katalog'. The 'TEGZEMPLARZ' table is highlighted with a red box. The 'TTYTUL\_KSIAZKI' table is also visible. A foreign key relationship is shown between 'MTYTUL\_KSIAZKI\_ID' in 'TEGZEMPLARZ' and 'ID' in 'TTYTUL\_KSIAZKI'. The main window shows the structure of both tables and a SQL query.

Column	Alias	Table	Output	Sort Type
ID		KRUK.TEGZEMPLARZ	<input checked="" type="checkbox"/>	
DTYPE		KRUK.TEGZEMPLARZ	<input checked="" type="checkbox"/>	
NUMER		KRUK.TEGZEMPLARZ	<input checked="" type="checkbox"/>	
MTYTUL_KSIAZKI_ID		KRUK.TEGZEMPLARZ	<input checked="" type="checkbox"/>	

```
SELECT ALL KRUK.TEGZEMPLARZ.ID,  
          KRUK.TEGZEMPLARZ.DTYPE,  
          KRUK.TEGZEMPLARZ.NUMER,  
          KRUK.TEGZEMPLARZ.MTYTUL_KSIAZKI_ID,  
          KRUK.TEGZEMPLARZ.TERMIN,  
          KRUK.TTYTUL_KSIAZKI.ID,  
          KRUK.TTYTUL_KSIAZKI.DTYPE,  
          KRUK.TTYTUL_KSIAZKI.ISBN,  
          KRUK.TTYTUL_KSIAZKI.TYTUL,  
          KRUK.TTYTUL_KSIAZKI.WYDAWNICTWO,  
          KRUK.TTYTUL_KSIAZKI.AUTOR,  
          KRUK.TTYTUL_KSIAZKI.AKTOR  
FROM KRUK.TEGZEMPLARZ  
INNER JOIN KRUK.TTYTUL_KSIAZKI  
ON KRUK.TEGZEMPLARZ.MTYTUL_KSIAZKI_ID = KRUK.TTYTUL_KSIAZKI.ID
```



## Ponowne wywołanie aplikacji w trybie **Create**

The screenshot shows the NetBeans IDE interface with the following components:

- Projects:** WebWypożyczalnia1
- Files:** Web Pages, Configuration Files, Server Resources
- Classes:** SessionBean1.java, persistence.xml, Start Page, jdbc:derby://localhost:1527/katalog [kruk on KRUK]
- SQL Editor:** `SELECT ALL KRUK.TEGZEMPLARZ.ID,`
- Output:** Java DB Database Process, GlassFish V2, WebWypożyczalnia1 (run)

The output window contains the following log messages:

```
deployed with moduleid = WebWypożyczalnia1
Initializing Sun's JavaServer Faces implementation (1.2_04-b20-p03) for context '/WebWypożyczalnia1'
[TopLink Info]: 2008.04.16 05:39:41.226--ServerSession(10882451)--TopLink, version: Oracle TopLink Essentials - 2.0.1 (Build b09d-fcs (12/06/2007))
[TopLink Info]: 2008.04.16 05:39:41.679--ServerSession(10882451)--file:/E:/Dydaktyka/d1/Komponenty_1/proby/WebWypożyczalnia1/build/web/WEB-INF/lib/Wypożyczalnia1app.jar-Wypoży
[TopLink Info]: 2008.04.16 06:39:41.961--ServerSession(10882451)--file:/E:/Dydaktyka/d1/Komponenty_1/proby/WebWypożyczalnia1/build/web/WEB-INF/lib/Wypożyczalnia1app.jar-Wypoży
deployed with moduleid = WebWypożyczalnia1
Initializing Sun's JavaServer Faces implementation (1.2_04-b20-p03) for context '/WebWypożyczalnia1'
[TopLink Info]: 2008.04.16 07:24:08.837--ServerSession(28940314)--TopLink, version: Oracle TopLink Essentials - 2.0.1 (Build b09d-fcs (12/06/2007))
[TopLink Info]: 2008.04.16 07:24:08.852--ServerSession(28940314)--file:/E:/Dydaktyka/d1/Komponenty_1/proby/WebWypożyczalnia1/build/web/WEB-INF/lib/Wypożyczalnia1app.jar-Wypoży
[TopLink Warning]: 2008.04.16 07:24:08.899--ServerSession(28940314)--Exception [TOPLINK-4002] (Oracle TopLink Essentials - 2.0.1 (Build b09d-fcs (12/06/2007))): oracle.toplink
Internal Exception: java.sql.SQLException: Table/View 'TEGZEMPLARZ' już istnieje w Schema 'KRUK'.
Error Code: -1
Call: CREATE TABLE TEGZEMPLARZ (ID BIGINT NOT NULL, DTYPE VARCHAR(31), NUMER INTEGER, MTYTUL_KSIAZKI_ID BIGINT, TERMIN DATE, PRIMARY KEY (ID))
Query: DataModifyQuery()
[TopLink Warning]: 2008.04.16 07:24:08.915--ServerSession(28940314)--Exception [TOPLINK-4002] (Oracle TopLink Essentials - 2.0.1 (Build b09d-fcs (12/06/2007))): oracle.toplink
Internal Exception: java.sql.SQLException: Table/View 'TTYTUL_KSIAZKI' już istnieje w Schema 'KRUK'.
Error Code: -1
Call: CREATE TABLE TTYTUL_KSIAZKI (ID BIGINT NOT NULL, DTYPE VARCHAR(31), ISBN VARCHAR(255), TYTUL VARCHAR(255), WYDAWNICTWO VARCHAR(255), AUTOR VARCHAR(255), AKTOR VARCHAR(25
Query: DataModifyQuery()
[TopLink Warning]: 2008.04.16 07:24:08.946--ServerSession(28940314)--Exception [TOPLINK-4002] (Oracle TopLink Essentials - 2.0.1 (Build b09d-fcs (12/06/2007))): oracle.toplink
Internal Exception: java.sql.SQLException: Constraint 'TGZEMPLRZMTYTLKSZKD' już istnieje w Schema 'KRUK'.
Error Code: -1
Call: ALTER TABLE TEGZEMPLARZ ADD CONSTRAINT TGZEMPLRZMTYTLKSZKD FOREIGN KEY (MTYTUL_KSIAZKI_ID) REFERENCES TTYTUL_KSIAZKI (ID)
Query: DataModifyQuery()
[TopLink Warning]: 2008.04.16 07:24:08.962--ServerSession(28940314)--Exception [TOPLINK-4002] (Oracle TopLink Essentials - 2.0.1 (Build b09d-fcs (12/06/2007))): oracle.toplink
Internal Exception: java.sql.SQLException: Table/View 'SEQUENCE' już istnieje w Schema 'KRUK'.
Error Code: -1
Call: CREATE TABLE SEQUENCE (SEQ_NAME VARCHAR(50) NOT NULL, SEQ_COUNT DECIMAL, PRIMARY KEY (SEQ_NAME))
Query: DataModifyQuery()
```

# Ponowne uruchamianie aplikacji po założeniu tabel bez wykonywania skryptów SQL typu Create

The screenshot displays the NetBeans IDE 6.0.1 interface. On the left, the 'Projects' and 'Files' view shows a database schema for 'jdbc:derby://localhost:1527/katalog [kruk on KRUK]'. The schema includes tables like 'SEQUENCE', 'TEGZEMPLARZ', and 'TTYTUL\_KSIAZKI'. The 'TTYTUL\_KSIAZKI' table is selected, showing its columns: ID, DTYPE, ISBN, TYTUL, WYDAWNICTWO, and AUTOR.

The main editor area shows the 'Persistence Units' configuration for 'Wypożyczalnia1appPU'. The configuration includes:

- Persistence Unit Name: Wypożyczalnia1appPU
- Persistence Library: TopLink
- JDBC Connection: jdbc:derby://localhost:1527/katalog [kruk on KRUK]
- Table Generation Strategy:  Create  Drop and Create  None
- Include All Entity Classes in "Wypożyczalnia1app" Module
- Include Entity Classes: wypożyczalnia1app.TEgzemplarz, wypożyczalnia1app.TEgzemplarz\_termin, wypożyczalnia1app.TTytul\_ksiazki, wypożyczalnia1app.TTytul\_ksiazki\_na\_kasecie

The 'Output' window at the bottom shows the following log messages:

```
Java DB Database Process GlassFish V2 WebWypożyczalnia1 (run)
deployed with moduleid = WebWypożyczalnia1
Initializing Sun's JavaServer Faces implementation (1.2_04-b20-p03) for context '/WebWypożyczalnia1'
[TopLink Info]: 2008.04.16 07:37:15.077--ServerSession(19868881)--TopLink, version: Oracle TopLink Essentials - 2.0.1 (Build b09d-fcs (12
[TopLink Info]: 2008.04.16 07:37:15.092--ServerSession(19868881)--file:/E:/Dydaktyka/d1/Komponenty_1/proby/WebWypożyczalnia1/build/web/WI
[TopLink Info]: 2008.04.16 07:37:16.122--ServerSession(19868881)--file:/E:/Dydaktyka/d1/Komponenty_1/proby/WebWypożyczalnia1/build/web/WI
```

# 4. Tworzenie warstwy klienta, prezentacji cd

Integrowanie warstwy prezentacji i klienta  
wykonane w technologii **Java Server Faces**

- Wykonanie przypadku użycia „dodaj tytuł”



# Wstawianie komponentów

The screenshot displays the NetBeans IDE 6.0.1 interface for a web application project named 'WebWypożyczalnia1'. The main workspace shows a design view of a JSP page with a grid layout. At the top of the grid is a table with the following data:

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

Below the table, there are two buttons: 'Zapisz tytuły do bazy' and 'Zapisz egzemplarze do bazy'. The form also includes five labels and five text input fields for entering data:

- Podaj tytuł
- Podaj autora
- Podaj ISBN
- Podaj wydawnictwo
- Podaj aktora

The left sidebar shows the project structure, including a 'tytulyaplikacjaPanel - Navigator' with a tree view of components like 'form1', 'panel', 'table1', 'bazaoperacjePanel', and 'tytulyaplikacjaPanel'. The right sidebar shows the 'Palette' with various UI components and the 'Properties' window for the selected component.

The bottom status bar shows the application is running on 'GlassFish V2' and 'WebWypożyczalnia1 (run)'. The 'Output' window is also visible.



# Wstawianie komponentów umieszczanych na komponencie **Grid Panel** z palety **Layout**

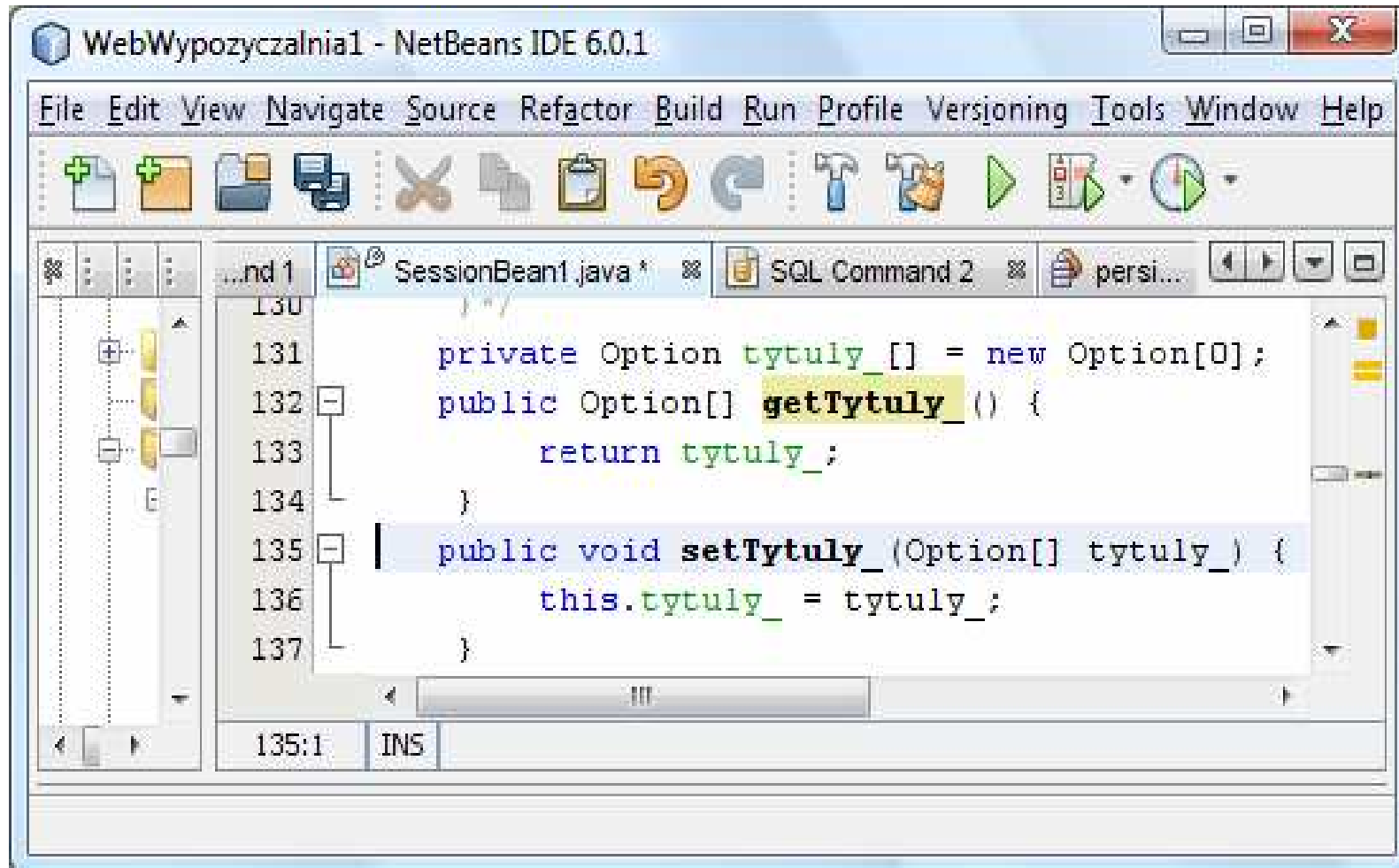
The screenshot displays the NetBeans IDE 6.0.1 interface for a web application named 'WebWypożyczalnia1'. The central design view shows a JSP page with a table and two buttons. The table has the following data:

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

Below the table are two buttons: 'Zapisz tytuły do bazy' and 'Zapisz egzemplarze do bazy'. Under each button are five input fields with labels: 'Podaj tytuł', 'Podaj autora', 'Podaj ISBN', 'Podaj wydawnictwo', and 'Podaj aktora'.

The left sidebar shows the project tree with 'tytulyaplikacjaPanel' selected. The right sidebar shows the 'Palette' with 'Basic' components and the 'Properties' window for 'tytulyaplikacjaPanel' showing 'gridPanel1 (HtmlPanelGrid)'. The bottom status bar shows 'Java DB Database Process', 'GlassFish V2', and 'WebWypożyczalnia1 (run)'.

Dodanie tablicy **Option tytuly\_[]** w klasie sesji **SessionBean1** do wykonania operacji **Bind to Data** dla komponentu **tytuly** typu **DropDownList** w celu wyświetlenia tytułów przechowywanych w aplikacji



The screenshot shows the NetBeans IDE 6.0.1 interface. The main editor window displays the code for SessionBean1.java. The code includes a private array declaration, a getter method, and a setter method. The setter method is currently selected with a mouse cursor.

```
130  
131     private Option tytuly [] = new Option[0];  
132     public Option[] getTytuly_ () {  
133         return tytuly_;  
134     }  
135     public void setTytuly_ (Option[] tytuly_) {  
136         this.tytuly_ = tytuly_;  
137     }
```

The status bar at the bottom indicates the cursor is at line 135, column 1, in Insert (INS) mode.

# Widok komponentów do realizacji przypadku użycia „dodaj tytuł”

The screenshot displays an IDE interface for a web application. The main window shows a design view of a page titled "tytuly". At the top, there is a table with the following data:

id	tytul	autor	ISBN	wydawnictwo	aktor
123	abc	abc	abc	abc	abc
123	abc	abc	abc	abc	abc
123	abc	abc	abc	abc	abc

Below the table are two buttons: "Zapisz tytuły do bazy" and "Zapisz egzemplarze do bazy". Further down are five labels with corresponding input fields: "Podaj tytuł", "Podaj autora", "Podaj ISBN", "Podaj wydawnictwo", and "Podaj aktora". At the bottom is a "Dodaj tytuł" button.

The left sidebar shows a component tree for "tytuly - Navigator". Red boxes labeled "id" and "text" are placed over the tree to indicate the data types of the components. Arrows point from these boxes to the following components in the tree:

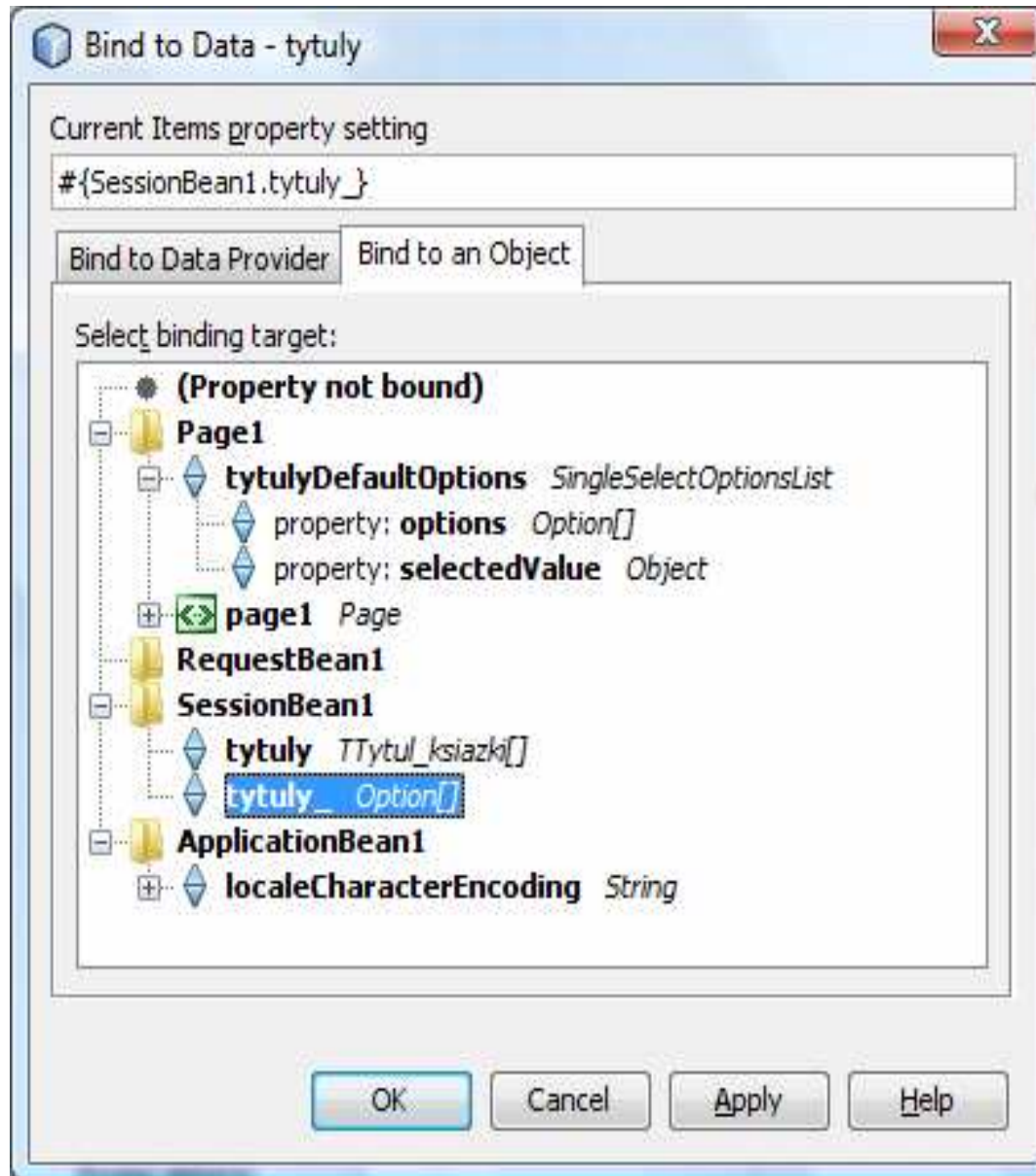
- id**: points to "bazaPanel", "tytul", "autor", "ISBN", "wydawnictwo", and "aktor".
- text**: points to "bazaPanel", "bazaoperacjePanel", "dodajtytulbaza:Zapisz tytuły do bazy", "dodajksiazkabaza:Zapisz egzemplarze do bazy", "tytulyaplikacjaPanel", "label1:Podaj tytuł", "label2:Podaj autora", "label3:Podaj ISBN", "label4:Podaj wydawnictwo", "label5:Podaj aktora", "aktor", and "dodajtytul:Dodaj tytuł".

The right sidebar shows the "Palette" and "Properties" panels. The "Properties" panel for "tytuly" shows the following settings:

- General: id: Page1
- Appearance: Background: [255,255,255], Page Layout: Grid Layout, Style Sheet: /resources/styl...
- Advanced: Response Encoding: UTF-8, Language: [empty]

The bottom right corner shows the "Page1" properties, indicating it is a "Page1 (request)".

Operacja **Bind toData** dla komponentu **tytuly** typu **DropDown List**





## Tworzenie kodu do połączenia warstwy prezentacji z warstwą biznesową za pomocą referencji do klasy typu Fasada - TApplikacja

The screenshot shows an IDE window with the following structure:

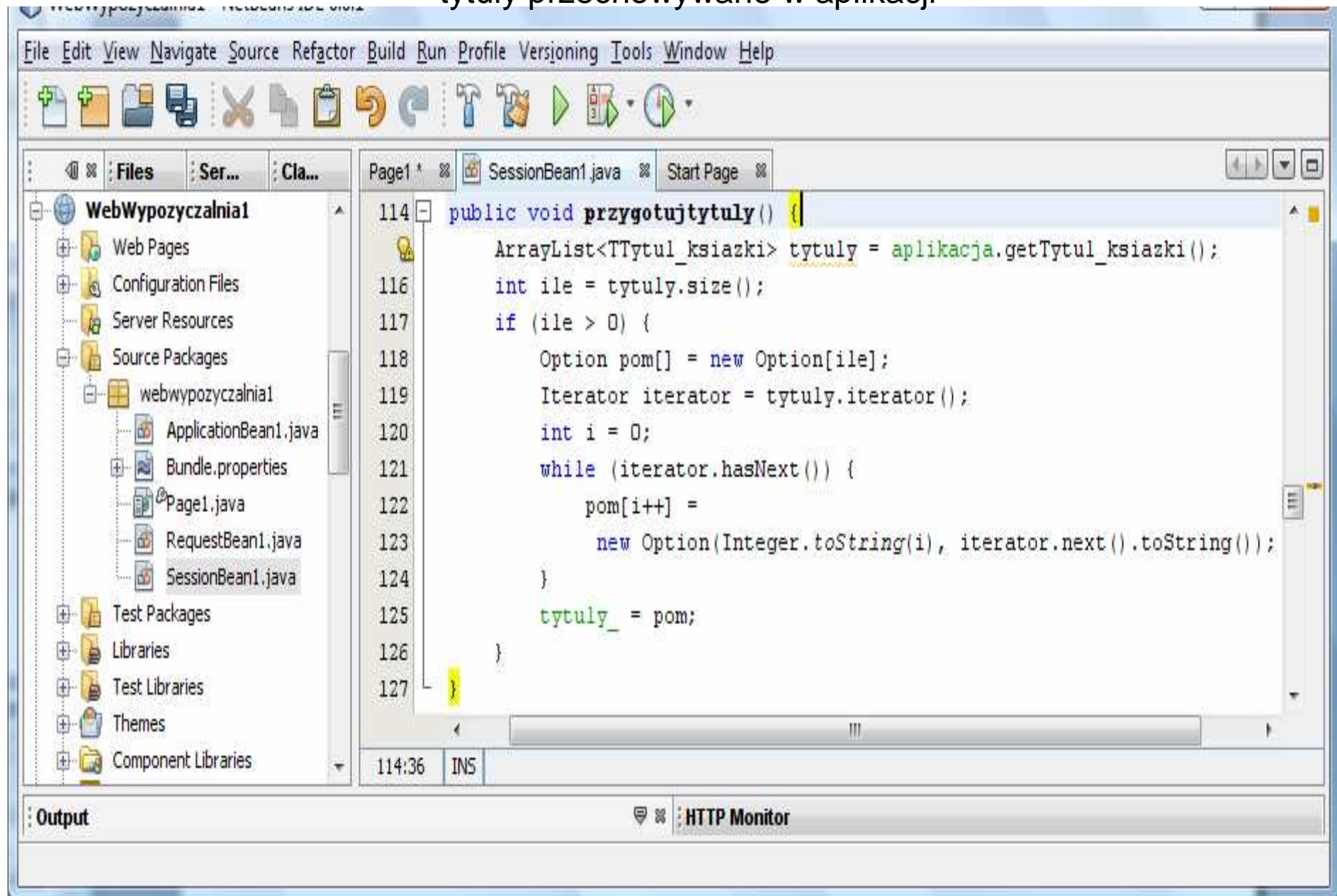
- Project: WebWypożyczalnia
- Source Packages: webwypożyczalnia
- Files: ApplicationBean1.java, Bundle.properties, Page1.java, RequestBean1.java, SessionBean1.java

The main editor displays the code for SessionBean1.java:

```
80     }
81     private TTytul_książki tytuly[];
82     public TTytul_książki[] getTytuly() {
83         return tytuly;
84     }
85
86     public void setTytuly(TTytul_książki[] tytuly) {
87         this.tytuly = tytuly;
88     }
89
90     public void updateTytuls() {
91         TTytul_książkiController tytulController = new TTytul_książkiController();
92         tytuly = tytulController.getTytul_książkis();
93     }
94     private TApplikacja aplikacja = new TApplikacja();
95     public TApplikacja getApplikacja() {
96         return aplikacja;
97     }
98
99     public void setApplikacja(TApplikacja aplikacja) {
100         this.aplikacja = aplikacja;
101     }
```

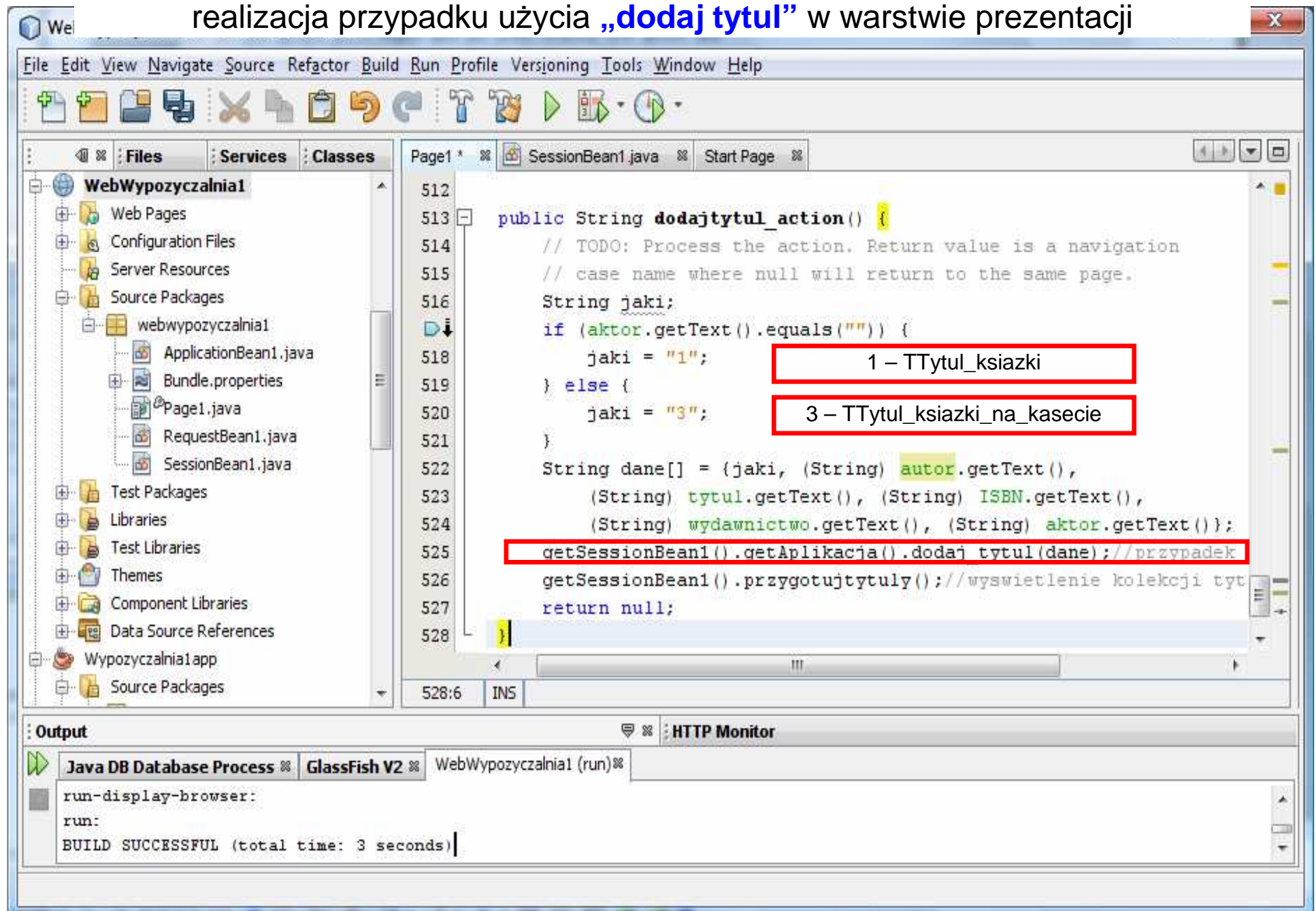
The code from line 94 to 101 is highlighted with a red box, indicating the facade implementation.

Przepisywanie zawartości kolekcji tytułów do tablicy **Option tytuły\_[]** w celu wykonania operacji **Bind to Data** z komponentem **tytuły** typu **DropDown List**, aby wyświetlać tytuły przechowywane w aplikacji

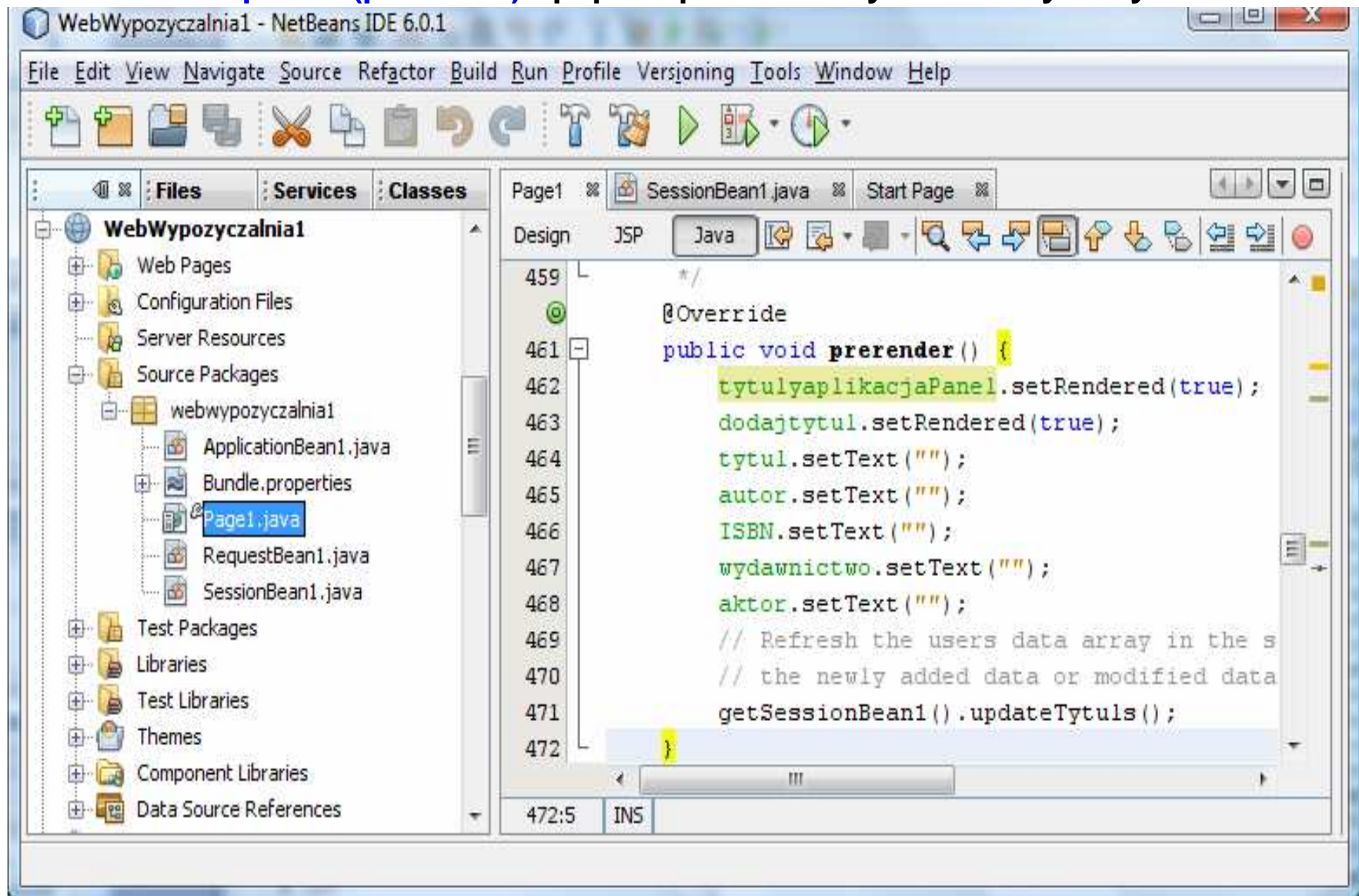




Obsługa zdarzenia klikania w przycisk „dodajtytul” w pliku **Page1.java** – realizacja przypadku użycia „dodaj tytul” w warstwie prezentacji



Ustawienie komponentów w metodzie **prerender** – jest wywoływana w fazie **Response (postback)** np. po zapisaniu danych do bazy danych





# Działanie przypadku użycia po stronie aplikacji, bez użycia bazy danych

The image displays two side-by-side screenshots of a web application interface, likely a library or book management system, running in Internet Explorer. Both screenshots show the same URL: `http://localhost:8080/WebWypożyczalnia1/faces/Page1.jsp`.

The left screenshot shows a search results table titled "Tytuły" with columns: id, tytuł, autor, ISBN, wydawnictwo, aktor. The table contains the text "No items found." Below the table are two buttons: "Zapisz tytuły do bazy" and "Zapisz egzemplarze do baz". At the bottom, there is a form with input fields for "Podaj tytuł", "Podaj autora", "Podaj ISBN", "Podaj wydawnictwo", and "Podaj aktora", along with a "Dodaj tytuł" button.

The right screenshot shows the same interface but with the form fields populated with the value "1". Additionally, a dropdown menu is open, showing a list of items with the following text: "Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1". The dropdown is currently displaying the first item, which is highlighted in blue.

## Tworzenie kodu do zapisu danych o tytułach w bazie danych

The image displays two screenshots of the NetBeans IDE 6.0.1. The top screenshot shows the `SessionBean1.java` file with the following code:

```
101 public void zapisz_tytuly_do_bazy() {
102     TTytul_książki newTTytul_książki;
103     // Add the new Entity to the database using UserController
104     Iterator it = getAplikacja().getTytul_książki().iterator();
105     TTytul_książkiController Tytul_książkiController =
106         new TTytul_książkiController();
107     while (it.hasNext()) {
108         newTTytul_książki = (TTytul_książki) it.next();
109         if (!Tytul_książkiController.findTTytul_książkis(newTTytul_książki))
110             Tytul_książkiController.addTTytul_książkis(newTTytul_książki);
111     }
112 }
```

Two annotations are present in the code:

- `//1` points to the `newTTytul_książki = (TTytul_książki) it.next();` line (line 108).
- `//2` points to the `Tytul_książkiController.addTTytul_książkis(newTTytul_książki);` line (line 110).

The bottom screenshot shows the `Page1` JSP file with the following code:

```
530 public String dodajtytulbaza_action() {
531     // TODO: Replace with your code
532     getSessionBean1().zapisz_tytuly_do_bazy();
533     return null;
534 }
```

The `Page1` tab is highlighted in the top-left corner of the IDE window.

**Nowa metoda w klasie TTytul\_książkiController do sprawdzania, czy obiekt TTytul\_książki w relacji wiele do jeden zostanie zapisany ponownie naruszając więzy integralności referencyjnej tabeli TTytul\_książki**

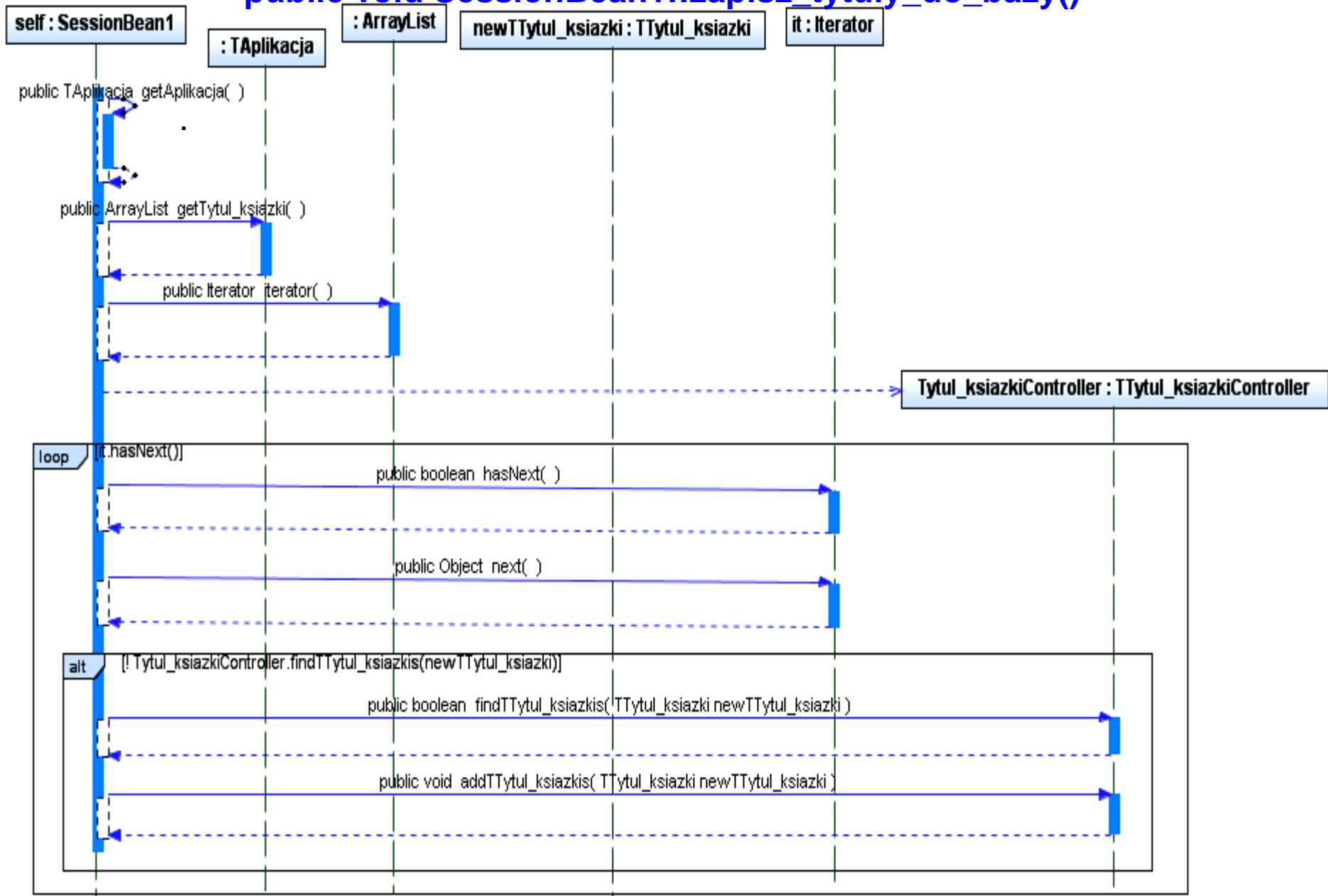
```
public boolean findTTytul_książkis(TTytul_książki TTytul_książki)
{
    EntityManager em = getEntityManager();
    try {
        return em.find(TTytul_książki.class, TTytul_książki.getId())!=null;
    }
    finally {
        em.close();
    }
}
```

**Uzupełniono kod konstruktora klasy TTytul\_książki– wstawia klucz główny równy 0 do danej, która jeszcze nie została zapisana do bazy danych. Wartość 0 klucza głównego jest wartością nie należącą do dziedziny wartości kluczy głównych – wtedy można przekazać taki obiekt do metody find w klasie EntityManager**

```
public TTytul_książki ()
    { id = new Long (-1);    // id=null; w lab7_8 dla wersji 2, slajd
    }
```

# Zapis danych do bazy tytułów metodą `add` obiektu `TTytul_książkiController`

## `public void SessionBean1::zapisz_tytuly_do_bazy()`





# Uruchomienie aplikacji

The screenshot shows a web browser window displaying a web application. The address bar shows the URL `http://localhost:8080/Wel`. The browser's toolbar includes a search bar with "Live Search" and various icons. A Norton security bar is visible at the top, indicating "Monitorowanie fałszywych witryn jest wł...".

The main content area features a table titled "Tytuły" with the following data:

id	tytuł	autor	ISBN	wydawnictwo	aktor
2	1	1	1	1	
3	1	1	1	1	1

Below the table are two buttons: "Zapisz tytuły do bazy" and "Zapisz egzemplarze do baz".

At the bottom, there is a form for adding a new title with the following fields:

- Podaj tytuł
- Podaj autora
- Podaj ISBN
- Podaj wydawnictwo
- Podaj aktora

A dropdown menu is open, showing the following options:

- Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1
- Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1**
- Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1

A "Dodaj tytuł" button is located at the bottom left of the form area.

The browser's status bar at the bottom indicates "Internet | Tryb chroniony: włączony" and a zoom level of "100%".

Widok bazy danych w zakładce **Services** – wykonanie operacji **ViewData** dostępnej z wyskakującego menu po naciśnięciu tabeli **TTytul\_ksiazki** prawym klawiszem myszy

The screenshot shows the NetBeans IDE 6.0.1 interface. On the left, the 'Databases' tree is expanded to show the 'TTYTUL\_KSIAZKI' table. The table's columns are listed: ID, DTYPE, ISBN, TYTUL, WYDAWNI..., AUTOR, and AKTOR. The main editor window displays the SQL command: `select * from KRUK.TTYTUL_KSIAZKI`. Below the editor, a data grid shows the results of the query. The grid has a header row with columns: ID, DTYPE, ISBN, TYTUL, WYDAWNI..., AUTOR, and AKTOR. The data rows are:

ID	DTYPE	ISBN	TYTUL	WYDAWNI...	AUTOR	AKTOR
2	TTytul_ksiazki	1	1	1	1	NULL
3	TTytul_ksiazki_na_kasecie	1	1	1	1	1
152	TTytul_ksiazki	2	2	2	2	NULL
852	TTytul_ksiazki					NULL

# 5. Tworzenie warstwy klienta, prezentacji cd

Integrowanie warstwy prezentacji i klienta  
wykonane w technologii **Java Server Faces**

- Wykonanie przypadku użycia „dodaj  
egzemplarz”

# Przygotowanie komponentów dla przypadku użycia „dodaj egzemplarz”

The screenshot displays the NetBeans IDE 6.0.1 interface for a project named 'WebWypożyczalnia1'. The 'panel - Navigator' on the left shows a hierarchical structure of components under 'form1', including 'panel', 'bazaPanel', 'bazaoperacjePanel', and 'aplikacjaoperacjePanel'. The main design area shows two data tables: 'Tytuły' and 'Książki'. Below the tables are buttons for 'Zapisz tytuły do bazy', 'Zapisz egzemplarze do bazy', 'Dodaj tytuł', and 'Dodaj egzemplarz'. At the bottom, there are input fields for 'Podaj tytuł', 'Podaj autora', 'Podaj ISBN', 'Podaj wydawnictwo', 'Podaj aktora', 'Podaj numer', and 'Podaj termin', along with dropdown menus and a 'Dodaj' button.

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

column1	column2	column3
row1_column1	row1_column2	row1_column3
row2_column1	row2_column2	row2_column3
row3_column1	row3_column2	row3_column3
row4_column1	row4_column2	row4_column3
row5_column1	row5_column2	row5_column3

Buttons: Zapisz tytuły do bazy, Zapisz egzemplarze do bazy, Dodaj tytuł, Dodaj egzemplarz

Form fields: Podaj tytuł, Podaj autora, Podaj ISBN, Podaj wydawnictwo, Podaj aktora, Podaj numer, Podaj termin



# Tworzenie kodu do „bindowania” komponentu **Table** z danymi z tabeli **TEgzemplarz** z bazy danych za pomocą tablicy **TEgzemplarz ksiazki [ ]**

```
Page1  SessionBean1.java  Start Page  TEgzemplarzController.java
141 private TEgzemplarz ksiazki[];
142 public TEgzemplarz[] getKsiazki() {
143     return ksiazki;
144 }
145 public void setKsiazki(TEgzemplarz[] ksiazki) {
146     this.ksiazki = ksiazki;
147 }
148 public void updateKsiazkis() {
149     TEgzemplarzController egzController = new TEgzemplarzController();
150     ksiazki = egzController.getTEgzemplarzs();
151 }
```

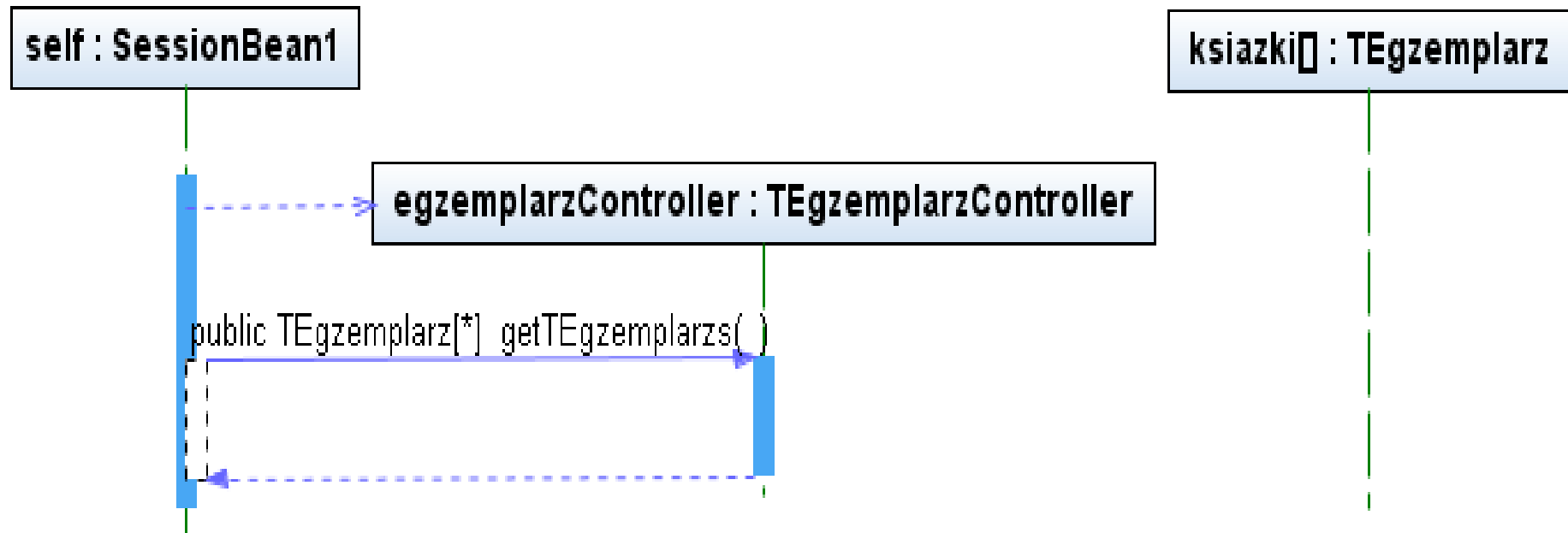
Output: Java DB Database Process | GlassFish V2 | WebWypożyczalnia1 (dist)

```
Page1  SessionBean1.java *  Start Page  TEgzemplarzController.java
66 Managed Component Initialization
77 // Perform application initialization that must complete
78 // *after* managed components are initialized
79 // TODO - add your own initialization code here
80 updateTytuls();
81 updateKsiazkis();
```

Output: Java DB Database Process | GlassFish V2 | WebWypożyczalnia1 (dist)

# Odczyt książek z bazy danych

## `public void SessionBean1::updateKsiazkis()`



# Wstawianie nowych komponentów rozpinanych na zagnieżdżonych komponentach typu **Layout – Grid Panel**

The screenshot shows an IDE window titled "Web" with a menu bar (File, Edit, View, Navigate, Source, Refactor, Build, Run, Profile, Versioning, Tools, Window, Help) and a toolbar. The main workspace displays a design view with two tables and form elements.

**Tytuły**

id	tytul	autor	ISBN	wydawnictwo	aktor
123	abc	abc	abc	abc	abc
123	abc	abc	abc	abc	abc
123	abc	abc	abc	abc	abc

**Książki**

id	numer	termin	tytul_ksiazki
123	123	Thu Apr 17 13:16:29 CEST 2008	Tytul: abc Autor: abc ISBN: abc Wydawnictwo: abc
123	123	Thu Apr 17 13:16:29 CEST 2008	Tytul: abc Autor: abc ISBN: abc Wydawnictwo: abc
123	123	Thu Apr 17 13:16:29 CEST 2008	Tytul: abc Autor: abc ISBN: abc Wydawnictwo: abc

**Buttons:** Zapisz tytuły do bazy, Zapisz egzemplarze do bazy, Dodaj tytuł, Dodaj egzemplarz.

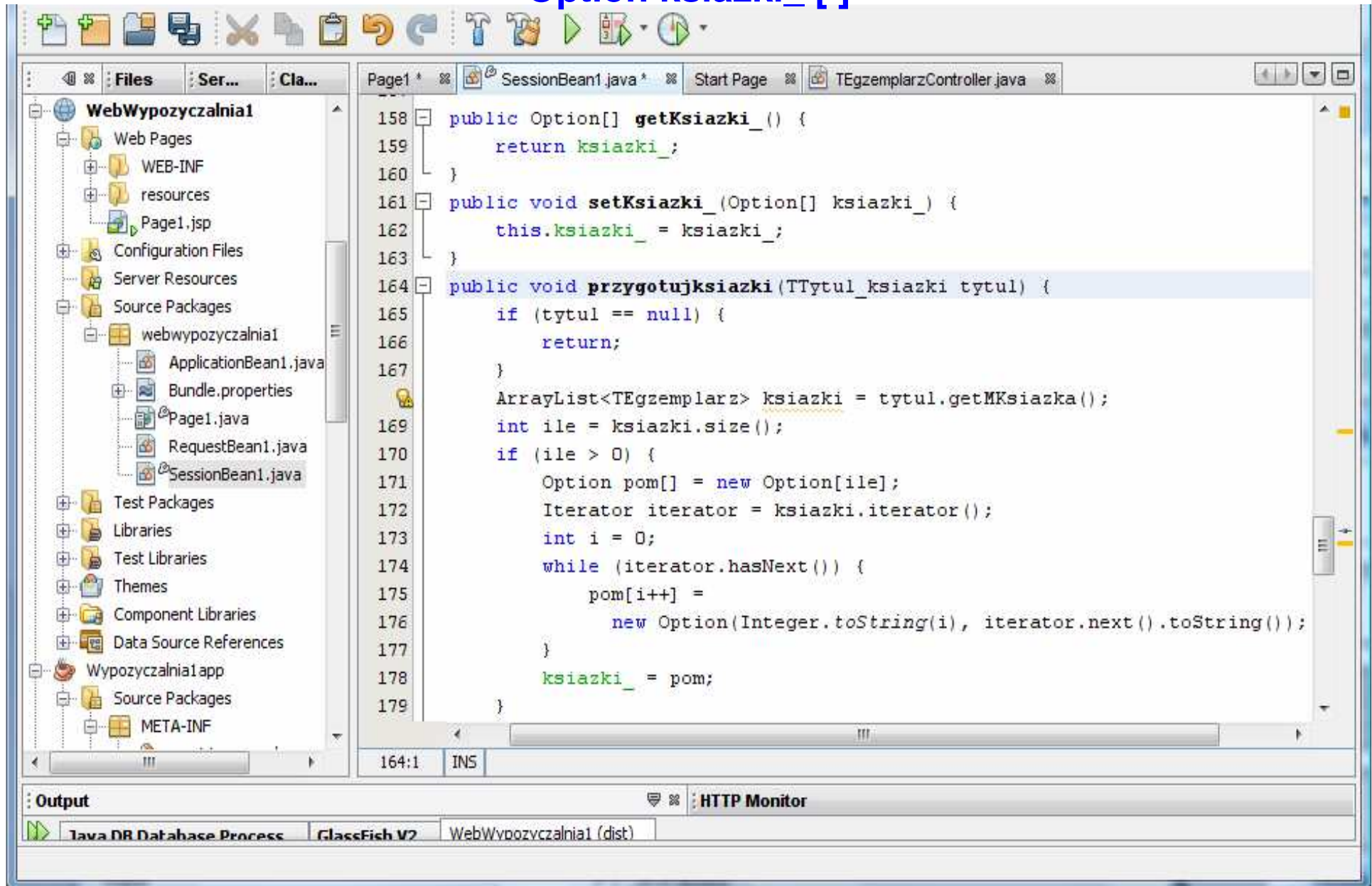
**Form Fields:**

- Podaj tytuł:
- Podaj autora:
- Podaj ISBN:
- Podaj wydawnictwo:
- Podaj aktora:
- Podaj numer:
- Podaj termin:
- Dropdown: a...
- Dropdown: Item 1

**Output:** GlassFish V2, Java DB Database Process, WebWypożyczalnia1 (run)



Wstawienie kodu do wyświetlania danych o egzemplarzach przechowywanych w aplikacji w elemencie **ksiazki** typu **DropDown List** za pośrednictwem tablicy **Option ksiazki\_ [ ]**



The screenshot shows an IDE window with the following content:

- Project Explorer (Left):** Shows a project named "WebWypożyczalnia1" with a package "webwypożyczalnia1" containing files like "ApplicationBean1.java", "Page1.java", "RequestBean1.java", and "SessionBean1.java".
- Code Editor (Center):** Displays the code for "SessionBean1.java". The code is as follows:

```
158 public Option[] getKsiazki_() {
159     return ksiazki_;
160 }
161 public void setKsiazki_(Option[] ksiazki_) {
162     this.ksiazki_ = ksiazki_;
163 }
164 public void przygotujksiazki(TTytul_ksiazki tytul) {
165     if (tytul == null) {
166         return;
167     }
168     ArrayList<TEgzemplarz> ksiazki = tytul.getMKsiazka();
169     int ile = ksiazki.size();
170     if (ile > 0) {
171         Option pom[] = new Option[ile];
172         Iterator iterator = ksiazki.iterator();
173         int i = 0;
174         while (iterator.hasNext()) {
175             pom[i++] =
176                 new Option(Integer.toString(i), iterator.next().toString());
177         }
178         ksiazki_ = pom;
179     }
}
```
- Output Console (Bottom):** Shows the status of the application: "Java DB Database Process", "GlassFish V2", and "WebWypożyczalnia1 (dist)".



# Wykonanie operacji **Bind to Data** między elementem **ksiązki** typu **DropDown List** oraz tablicą **Option** **ksiązki\_ [ ]**

The screenshot shows an IDE window with a 'Bind to Data - książki' dialog box open. The dialog is configured to bind a 'DropDown List' widget to the 'ksiązki' property of a 'SessionBean1' object. The background shows a web page design with a table of books and a form with input fields.

**Table**

id	nun
123	123
123	123
123	123

**Form**

Podaj nu  
Podaj te

**Bind to Data - książki**

Current Items property setting  
#{SessionBean1.książki\_}

Bind to Data Provider Bind to an Object

Select binding target:

- (Property not bound)
- Page1
  - ksiązkiDefaultOptions SingleSelectOptionsList
    - property: options Option[]
    - property: selectedValue Object
  - page1 Page
  - RequestBean1
  - SessionBean1
    - aplikacja TApplikacja
    - ksiązki Option[]
    - tytuly TTytul\_ksiązki[]
    - tytuly\_ Option[]
    - ksiązki TEgzemplarz[]
  - ApplicationBean1
    - localeCharacterEncoding String

OK Cancel Apply Help

# Efekt operacji **Bind to Data** – okienko **Navigator** przedstawia strukturę paneli i komponentów

The screenshot displays an IDE window titled 'Web' with a 'Navigator' panel on the left and a design view on the right. The 'Navigator' panel shows a tree structure for a web application named 'ksiazki'. The design view shows two data tables and a form with input fields and buttons.

**ksiazki - Navigator**

- form1
  - panel
    - bazaPanel
      - bazatytyly
      - bazaksiążki
    - bazaoperacjePanel
      - dodajtytulbaza:Zapisz tytuły do ba...
      - dodajksiazkabaza:Zapisz egzemplar...
    - aplikacjadanePanel
      - tytulyaplikacjaPanel
        - label1:Podaj tytuł
        - tytuł
        - label2:Podaj autora
        - autor
        - label3:Podaj ISBN
        - ISBN
        - label4:Podaj wydawnictwo
        - wydawnictwo
        - label5:Podaj aktora
        - aktor
      - ksiazkiaplikacjadane
        - label6:Podaj numer
        - numer
        - label7:Podaj termin
        - termin
    - aplikacjaoperacjePanel
      - tytuly
      - ksiazki

**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

**Książki**

id	numer	termin	tytuł_ksiazki
123	123	Thu Apr 17 13:20:50 CEST 2008	Tytuł: abc Autor: abc ISBN: abc Wydawnictwo: abc
123	123	Thu Apr 17 13:20:50 CEST 2008	Tytuł: abc Autor: abc ISBN: abc Wydawnictwo: abc
123	123	Thu Apr 17 13:20:50 CEST 2008	Tytuł: abc Autor: abc ISBN: abc Wydawnictwo: abc

**Zapisz tytuły do bazy** **Zapisz egzemplarze do bazy**

**Podaj tytuł**

**Podaj autora**

**Podaj ISBN**

**Podaj wydawnictwo**

**Podaj aktora**

**Podaj numer**  **Podaj termin**

**Dodaj tytuł** **Dodaj egzemplarz**

Wstawianie kodu dla przypadku użycia „dodaj egzemplarz” w warstwie prezentacji – wybór tytułu następuje w elemencie **tytuly** typu **DropDown List** przez wybór pozycji z listy (`tytuly.getSelected()`), natomiast dane o numerze i ewentualnie terminie podawane są z pól tekstowych **numer** i **termin**

```
...java | SQL Command 6 | TApplikacja.java | Page1 * | ApplicationBean1.java | Start Page |
725 public String dodajksiazke_action() {
726     // TODO: Process the action. Return value is a navigation
727     // case name where null will return to the same page.
728     String jaki,jaka;
729     if (termin.getText().equals(""))
730     { jaka = "0"; }
731     else
732     { jaka = "1"; } //Informacja dla fabryki - jaki tytuł wyszukać
733     int nr = Integer.parseInt((String) tytuly.getSelected());
734     if (nr==0) return null;
735     Option pom[]=getSessionBean1().getTytuly_();
736     String pom1=pom[nr-1].getLabel();
737     String pom2[]=pom1.split(" ");
738     if (pom2.length==8) //Informacja dla fabryki - jaki egzemplarz wykonać
739     { pom1="";
740       jaki="0"; }
741     else
742     { pom1=pom2[9];
743       jaki="2"; }
744     String dane1[]= {jaki, (String) pom2[5], pom1};
745     String dane2[]= {jaka, (String) numer.getText(), (String) termin.getText()};
746     getSessionBean1().przygotujksiazki(
747         getSessionBean1().getApplikacja().dodaj_ksiazke(dane1, dane2));
748     return null;
749 }
```

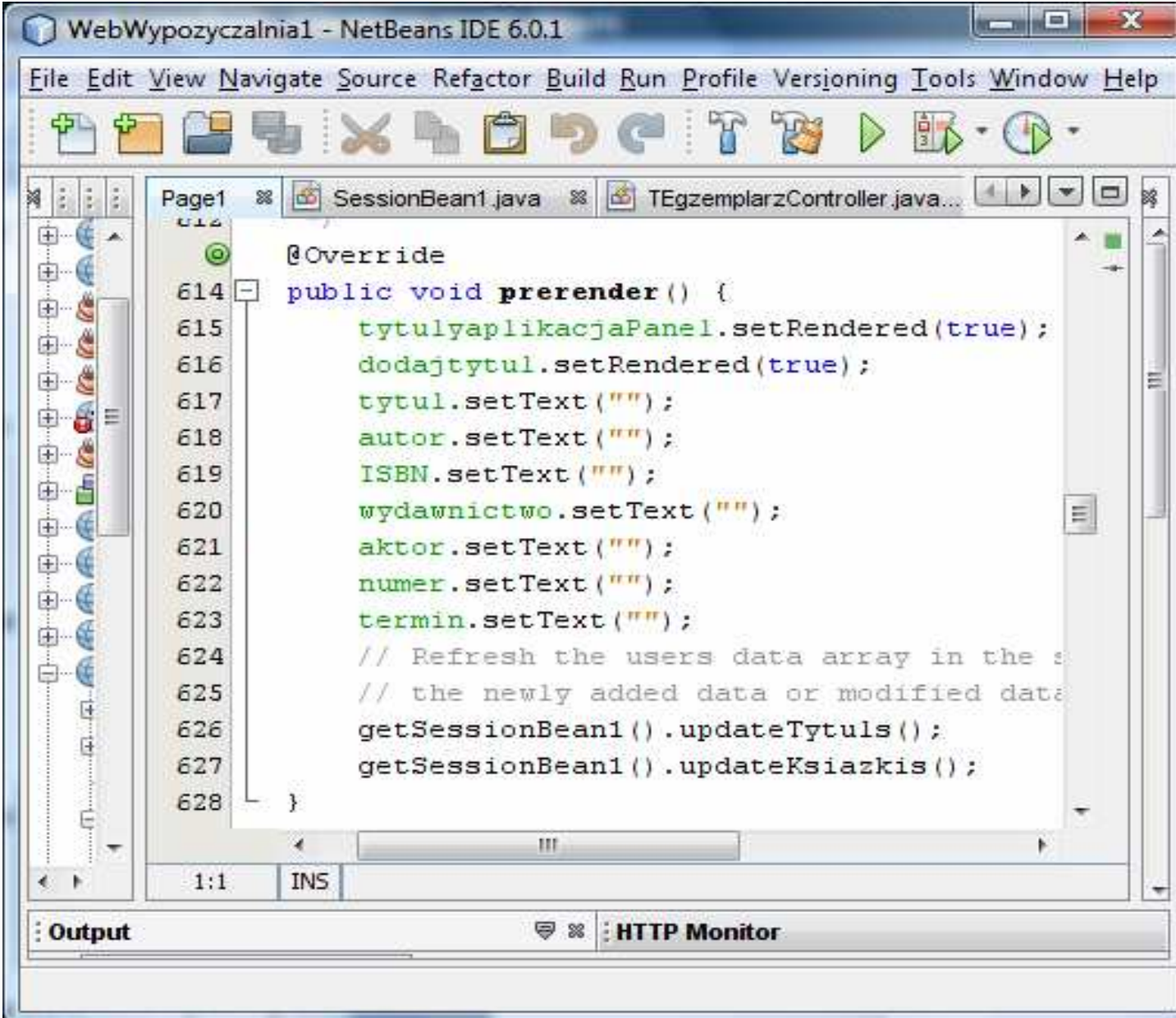
Wybór typu klasy egzemplarz:  
0 – TEgzemplarz  
1 – TEgzemplarz\_termin

Pobranie z listy rozwijanej typu DropDown informacji o tytule książki, dla którego należy dodać książkę

Wybór typu klasy tytuł do wyszukiwania:  
0 – TTytuł\_ksiazki (4 dane – 8 łańcuchów w łańcuchu tytułu)  
2 – TTytuł\_ksiazki\_na\_kasce (5 danych – 10 łańcuchów w łańcuchu tytułu)



Uzupełnienie kodu metody **prerender** w klasie **Page1.java** - – jest wywoływana w **fazie Response (postback)** np. po zapisaniu danych do bazy danych



The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** WebWypożyczalnia1 - NetBeans IDE 6.0.1
- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Build, Run, Profile, Versioning, Tools, Window, Help
- Toolbar:** Standard IDE icons for file operations and execution.
- Project Explorer:** Shows a tree view of the project structure on the left.
- Editor:** Displays the code for the `prerender` method in `Page1.java`. The code is as follows:

```
614 @Override
615 public void prerender() {
616     tytułaplikacjaPanel.setRendered(true);
617     dodajtytuł.setRendered(true);
618     tytuł.setText("");
619     autor.setText("");
620     ISBN.setText("");
621     wydawnictwo.setText("");
622     aktor.setText("");
623     numer.setText("");
624     termin.setText("");
625     // Refresh the users data array in the s
626     // the newly added data or modified data
627     getSessionBean1().updateTytuls();
628     getSessionBean1().updateKsiazkis();
629 }
```
- Status Bar:** Shows "1:1" and "INS" (Insert mode).
- Output Window:** Located at the bottom, with "HTTP Monitor" selected.



http://localhost:8080/WebWypożyczalnia1/faces/Page1.jsp - Windows Internet Explorer

http://localhost:8080/WebWypożyczalnia1/faces/Page1.jsp

http://localhost:8080/WebWypożyczalnia1/faces/...

Norton

Monitorowanie fałszywych witryn jest włączone

Opcje

Tytuły					
id	tytuł	autor	ISBN	wydawnictwo	aktor
2	1	1	1	1	
3	1	1	1	1	1

Książki			
id	numer	termin	tytuł_książki
No items found.			

Zapisz tytuły do bazy

Zapisz egzemplarze do baz

Podaj tytuł

Podaj autora

Podaj ISBN

Podaj wydawnictwo

Podaj aktora

Podaj numer

Podaj termin

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

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2 Numer: 2 termin: Tue Apr 22 14:52:59 CEST 2008

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

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

Dodaj tytuł

Dodaj egzemplarz

Internet | Tryb chroniony: włączony

100%

http://localhost:8080/WebWypozyczalnia1/faces/Page1.jsp - Windows Internet Explorer

http://localhost:8080/WebWypozyczalnia1/faces/Page1.jsp

http://localhost:8080/WebWypozyczalnia1/faces/...

Norton

Monitorowanie fałszywych witryn jest włączone

Opcje

Tytuły					
id	tytuł	autor	ISBN	wydawnictwo	aktor
2	1	1	1	1	
3	1	1	1	1	1

Książki			
id	numer	termin	tytuł_książki
No items found.			

Zapisz tytuły do bazy

Zapisz egzemplarze do baz

Podaj tytuł

Podaj autora

Podaj ISBN

Podaj wydawnictwo

Podaj aktora

Podaj numer

Podaj termin

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

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2 Numer: 2 termin: Tue Apr 22 14:52:59 CEST 2008

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2 Numer: 2 termin: Tue Apr 22 14:52:59 CEST 2008

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

Dodaj tytuł

Dodaj egzemplarz

Internet | Tryb chroniony: włączony

100%

## Aktualizacja danych w aplikacji pobieranych z bazy danych za pomocą tablic **TTytul\_książki** `tytuly[ ]` oraz **TEgzemplarz** `książki[ ]`

The image shows two windows from the NetBeans IDE. The left window displays the code for `SessionBean1.java`, and the right window displays the code for `EgzemplarzController.java`. Both windows are titled "WebWypożyczalnia1 - NetBeans IDE 6.5".

In the `SessionBean1.java` window, the `updateAplikacja()` method is highlighted with a red box. The code in this method is as follows:

```
119 public void updateAplikacja() {
120     for (int i = 0; i < tytuly.length; i++)
121         aplikacja.getTytul_książki().add(tytuly[i]);
122     Iterator it = aplikacja.getTytul_książki().iterator();
123     while (it.hasNext()) {
124         TTytul_książki tytul = (TTytul_książki) it.next();
125         for (int j = 0; j < książki.length; j++) {
126             if (książki[j].getMTytul_książki().equals(tytul))
127                 tytul.getMKsiążka().add(książki[j]);
128         }
129     }
130 }
```

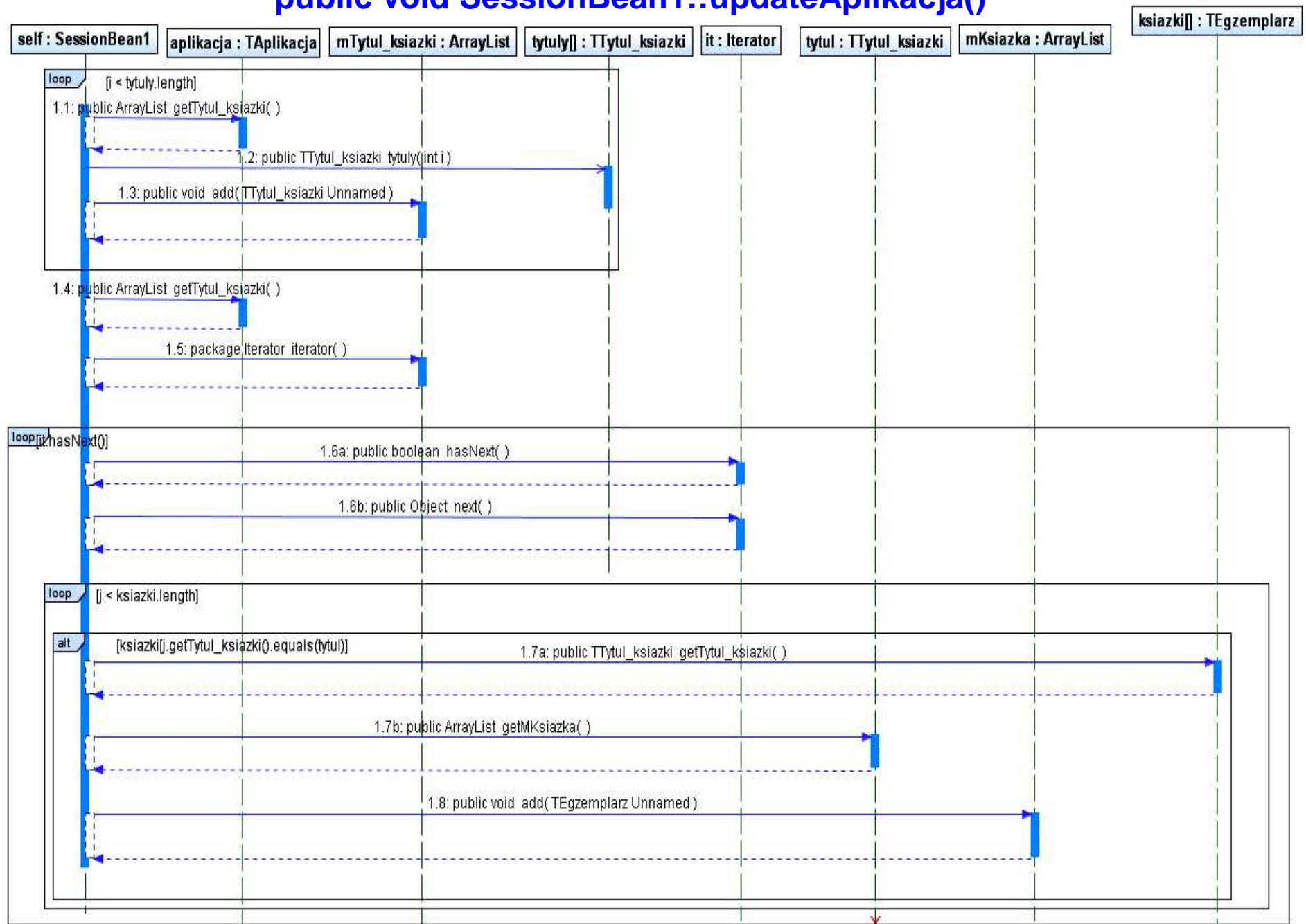
In the `EgzemplarzController.java` window, the `init()` method is visible, with the `przygotujtytuly();` call highlighted in blue. The code in this method is as follows:

```
60 public void init() {
61     // Perform initiali
62     super.init();
63     // Perform applicat
64     // *before* managed
65     // TODO - add your
66
67     Managed Component
68     // Perform applicat
69     // *after* managed
70     // TODO - add your
71     updateTytuls();
72     updateKsiążkis();
73     updateAplikacja();
74     przygotujtytuly();
75 }
```



# Odczyt danych z bazy tytułów oraz książek za pomocą tablic encji tytuły i książki

## public void SessionBean1::updateAplikacja()





## Wykonanie kodu zapisującego dane o egzemplarzach przechowywanych w aplikacji do bazy danych

The image displays two screenshots of the NetBeans IDE 6.0.1 interface, illustrating the implementation of a database saving function for book records.

**Top Screenshot:** Shows the `SessionBean1.java` file. The method `zapisz_książki_do_bazy()` is defined, which iterates through book titles and their corresponding records, creating `TEgzemplarz` objects and saving them to the database using `TEgzemplarzController`. The code is as follows:

```
182 public void zapisz_książki_do_bazy() {
183     TEgzemplarz newTEgzemplarz;
184     TTYtul_książki newTTYtul_książki;
185     // Add the new Entity to the database using UserController
186     Iterator it = getAplikacja().getTytul_książki().iterator();
187     TEgzemplarzController egzemplarzController =
188         new TEgzemplarzController();
189     while (it.hasNext()) {
190         newTTYtul_książki = (TTYtul_książki) it.next();
191         Iterator it_ = newTTYtul_książki.getMKsiążka().iterator();
192         while (it_.hasNext()) {
193             newTEgzemplarz = (TEgzemplarz) it_.next();
194             if (!egzemplarzController.findTEgzemplarz(newTEgzemplarz))
195                 egzemplarzController.addTEgzemplarz(newTEgzemplarz);
196         }
197     }
198 }
```

**Bottom Screenshot:** Shows the `Page1` file, which contains the `dodajksiążkabaza_action()` method. This method calls the `zapisz_książki_do_bazy()` method from the `SessionBean1` object. The code is as follows:

```
692 public String dodajksiążkabaza_action() {
693     // TODO: Replace with your code
694     getSessionBean1().zapisz_książki_do_bazy();
695     return null;
696 }
```

**Nowa metoda w klasie TEgzemplarzController do sprawdzania, czy obiekt TEgzemplarz w relacji wiele do jeden zostanie zapisany ponownie naruszając więzy integralności referencyjnej tabeli TEgzemplarz**

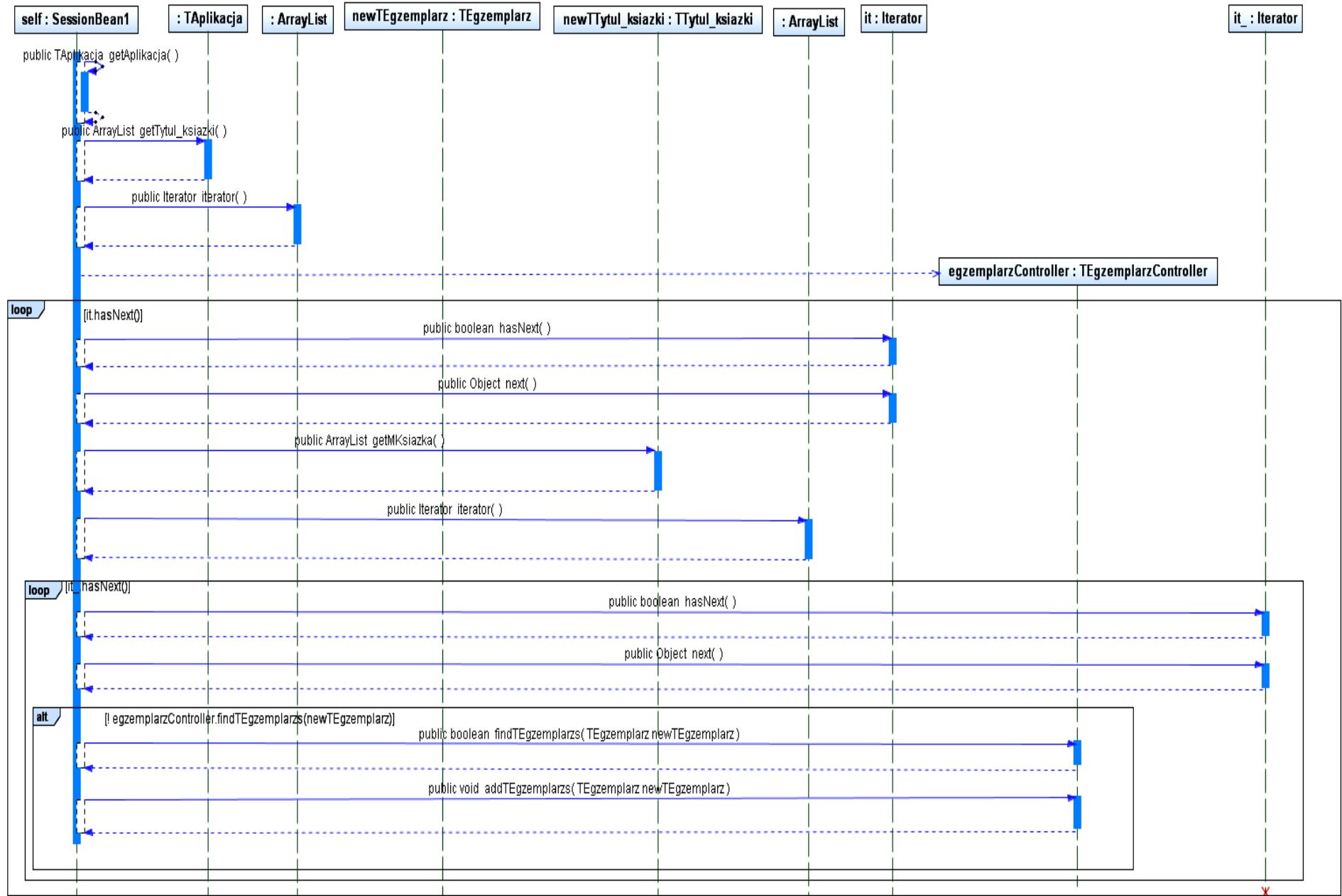
```
public boolean findTEgzemplarzs(TEgzemplarz TEgzemplarz)
{
    EntityManager em = getEntityManager();
    try {
        return em.find(TEgzemplarz.class, TEgzemplarz.getId())!null; }
    finally {
        em.close();
    }
}
```

**Uzupełniono kod konstruktora klasy TEgzemplarz – wstawia klucz główny równy 0 do danej, która jeszcze nie została zapisana do bazy danych. Wartość 0 klucza głównego jest wartością nie należącą do dziedziny wartości kluczy głównych- wtedy można przekazać taki obiekt do metody find w klasie EntityManager**

```
public TEgzemplarz() {
    id = new Long(-1); // id=null; dla wersji 2 w lab.7_8
}
```

# Zapis danych do bazy książek metodą add obiektu TEgzemplarzController

## public void SessionBean1::zapisz\_książki\_do\_bazy()



# Uruchomienie aplikacji

http://localhost:8080/WebWypożyczalnia1/ - Windows Internet Explorer

http://localhost:8080/WebWypożyczalnia1/ Live Search

Norton™ Monitorowanie fałszywych witryn jest włączone Opcje

Tytuły					
id	tytuł	autor	ISBN	wydawnictwo	aktor
2	1	1	1	1	
3	1	1	1	1	1

Książki			
id	numer	termin	tytuł_książki
No items found.			

Zapisz tytuły do bazy

Zapisz egzemplarze do baz

Podaj tytuł

Podaj autora

Podaj ISBN

Podaj wydawnictwo

Podaj aktora

Podaj numer

Podaj termin

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

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

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

Gotowe Internet | Tryb chroniony: włączony 100%



# Wstawianie danych z aplikacji do tabeli **TEgzemplarz** w bazie danych (1)

The screenshot shows a web browser window with the following content:

**Tytuły**

id	tytuł	autor	ISBN	wydawnictwo	aktor
2	1	1	1	1	
3	1	1	1	1	1

**Książki**

id	numer	termin	tytuł_książki
52	1		Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1
53	2	Sat Apr 19 00:00:00 CEST 2008	Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1

Buttons:

Form fields:

Podaj tytuł:

Podaj autora:

Podaj ISBN:

Podaj wydawnictwo:

Podaj aktora:

Podaj numer:

Podaj termin:

Dropdown menu (open):

- Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1
- Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1
- Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1

Buttons:

Status bar: Gotowe | Internet | Tryb chroniony: włączony | 100%

## Wstawianie danych z aplikacji do tabeli **TEgzemplarz** w bazie danych (2)

The screenshot shows a web browser window displaying a library management application. The address bar shows the URL: `http://localhost:8080/WebWypożyczalnia1/faces/Page1.jsp`. The page contains two data tables and several form elements.

**Tytuły**

id	tytul	autor	ISBN	wydawnictwo	aktor
2	1	1	1	1	
3	1	1	1	1	1

**Książki**

id	numer	termin	tytul_książki
52	1		Tytul: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1
53	2	Sat Apr 19 00:00:00 CEST 2008	Tytul: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1

Buttons:

Form fields for adding data:

- Podaj tytuł:
- Podaj autora:
- Podaj ISBN:
- Podaj wydawnictwo:
- Podaj aktora:
- Podaj numer:
- Podaj termin:

Buttons:

Status bar: Gotowe | Internet | Tryb chroniony: włączony | 100%

## Wstawianie danych z aplikacji do tabeli **TEgzemplarz** w bazie danych (3)

http://localhost:8080/WebWypożyczalnia1/faces/Page1.jsp - Windows Internet Explorer

http://localhost:8080/WebWypożyczalnia1/faces/Page1.jsp

Monitorowanie fałszywych witryn jest włączone

Tytuły					
id	tytuł	autor	ISBN	wydawnictwo	aktor
2	1	1	1	1	
3	1	1	1	1	1

Książki			
id	numer	termin	tytuł_książki
52	1		Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1
53	2	Sat Apr 19 00:00:00 CEST 2008	Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1
102	1	Fri Apr 18 00:00:00 CEST 2008	Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1

Zapisz tytuły do bazy

Zapisz egzemplarze do baz

Podaj tytuł

Podaj autora

Podaj ISBN

Podaj wydawnictwo

Podaj aktora

Podaj numer

Podaj termin

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

Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1 Numer: 1 termin: Fri Apr 18 15:15:33 CEST 2008

Dodaj tytuł

Dodaj egzemplarz

Gotowe

Internet | Tryb chroniony: włączony

100%



Widok bazy danych w zakładce **Services** – wykonanie operacji **ViewData** dostępnej z wyskakującego menu po naciśnięciu tabeli **TEgzemplarz** prawym klawiszem myszy

The screenshot shows an IDE interface with a database schema on the left and a query result on the right. The schema tree on the left shows a database with two schemas: 'jdbc:derby://localhost:1527/Zoo [kruk on KRUK]' and 'jdbc:derby://localhost:1527/katalog [kruk on KRUK]'. Under the 'katalog' schema, the 'TEGZEMPLARZ' table is selected. The table's columns are: ID, DTYPE, NUMER, MTYTUL\_K..., and TERMIN. The query result on the right shows the following data:

ID	DTYPE	NUMER	MTYTUL_K...	TERMIN
52	TEgzemplarz	1	2	NULL
53	TEgzemplarz...	2	2	2008-04-19
102	TEgzemplarz...	1	3	2008-04-18
202	TEgzemplarz...	1	152	2008-04-18
252	TEgzemplarz...	3	3	2008-04-21
302	TEgzemplarz...	3	2	2008-04-19
352	TEgzemplarz	2	152	NULL
402	TEgzemplarz	4	152	NULL
452	TEgzemplarz	5	152	NULL
453	TEgzemplarz	4	2	NULL
502	TEgzemplarz	2	3	NULL
552	TEgzemplarz	4	3	NULL
602	TEgzemplarz	5	3	NULL
652	TEgzemplarz	6	3	NULL
653	TEgzemplarz	8	3	NULL
702	TEgzemplarz	9	3	NULL
752	TEgzemplarz	10	2	NULL
802	TEgzemplarz	12	2	NULL
902	TEgzemplarz	11	3	NULL
952	TEgzemplarz	7	3	NULL
1002	TEgzemplarz	13	3	NULL
1052	TEgzemplarz	14	3	NULL
1102	TEgzemplarz	15	3	NULL
1152	TEgzemplarz	16	3	NULL
1202	TEgzemplarz	17	3	NULL
1252	TEgzemplarz	18	3	NULL
1302	TEgzemplarz	19	3	NULL
1303	TEgzemplarz	20	3	NULL
1353	TEgzemplarz	10	3	NULL