Projekt INP002017
Instrukcja 2

Autor

Dr inż. Zofia Kruczkiewicz

I. Czynności wykonane zgodnie z harmonogramem grupy w tygodniach 1-15

Tabela 2. Przebieg realizacji każdego z projektów (tabela 1)

Opis realizacji projektu					
Nr tygodni Daty	Termin przekazania prac	Spotkanie	Uwagi dotyczące realizacji zadań przez każdą z dwóch podgrup zespołu	Ocena cząstkowa	Zadania Scrum Master
1-tydzień 23.02.18-2.03.18 2-tydzień 2.03.18-9.03.18	Wynik prac należy wysłać za pomocą poczty elektronicznej do dnia 7.03.2018. Wynik prac zostanie oceniony przez prowadzącego zajęcia. Uwaga: należy wysłać wersję końcową projektu UML z ewentualnie uzupełnionym diagramem wymagań	Sprint planning meeting (45 min)	• Zajęcia organizacyjne (podział na grupy i podgrupy, przydzielenie ról projektowych, uzyskanie dostępu do wymaganych narzędzi) • User Stories - Opracowanie modelu biznesowego „świata rezczywistego”systemu • Sprint Backlog (formy pośrednie: Product Backlog), Sprint planining) - zdefiniowanie wymagań funkcjonalnych i niefunkcjonalnych wynikających z otrzymanego fragmentu opisu „świata rzeczywistego” w postaci diagramu wymagań wg instrukcji: Instrukcja 3 - definiowanie wymagań Podczas spotkania 9.03.2018 należy zaprezentować wyniki prac, czyli „Opis świata rzeczywistego” i diagram wymagań oprogramowania.		Współdziałanie z wykonawcami

3- tydzień 4- tydzień 9.03.2018- 23.03.2018	Konieczne wysłanie projektu UML z dodanym diagramem przypadków użycia do dnia 19.03.18 Diagram klas (wersja początkowa) powinien być dostarczony do dnia 22.03.18	Sprint planning meeting (45 min)	Należy wykonać: 1) diagram przypadków użycia wg instrukcji: Instrukcja 4 - specyfikacja wymagań za pomocą diagramu przypadków użycia. ze specyfikacją poszczególnych przypadków użycia 2) Diagramy klas i sekwencji wg instrukcji do lab5-7: Instrukcja - część 1 Instrukcja - część 2 Diagram klas UML zostanie przedstawiony na wykładzie w dniu 20.03.2018.		Eliminacja redundancji w projekcie. Udział w pracach projektowych
5- tydzień 6- tydzień 23.03.2018- 6.04.2018	Projekt UML oraz projekt z kodem Javy należy wysłać za pomocą poczty elektronicznej do dnia 2.04.2018.	Sprint planning meeting (45 min)	1) Projektowanie i implementacja oprogramowania w oparciu o wzorce projektowe – diagramy UML: rozwijanie diagramu klas, diagramy sekwencji i aktywności wybranych 3 prostych przypadków użycia reprezentujących projekt warstwy biznesowej projektu oraz implementacja projektu na platformie Java SE. Diagramy klas i sekwencji wg instrukcji do lab5-7: Instrukcja - część 1 Instrukcja - część 2		Eliminacja redundancji w projekcie. Udział w pracach projektowych i implementacji

7- tydzień 8- tydzień 6.04.2018- 20.04.2018	Kod programu w wersji SE i EE należy wysłać do dnia 16.04.18	Sprint planning meeting (45 min)	1) Tworzenie interfejsu graficznego użytkownika na platformie Java SE do programu zawierającego implementację warstwy biznesowej 2) Przekształcenie aplikacji na aplikację działającą na platformie Java EE, zawierającego warstwę klienta typu desktopowego 3) Instrukcja 1 - opisująca przebieg prac podczas 1- 8 tygodnia		Eliminacja redundancji w projekcie. Udział w pracach projektowych i implementacji

9-tydzień 10-tydzień 11- tydzień 12- tydzień 20.04.2018-18.05.2018	Kod programu wg 1 należy wysłać do dnia 4.05.2018, drugi wg p.2 do 14.05.2018.	Sprint planning meeting (45 min)	1) Rozwijanie oprogramowania: projektowanie i implementacja oprogramowania w oparciu o wzorce projektowe – diagramy UML: rozwijanie diagramu klas, diagramy sekwencji i aktywności wybranych 2 (minimum) złożonych przypadków użycia reprezentujących projekt warstwy biznesowej projektu oraz implementacja projektu na platformie Java EE 2) Rozwijanie aplikacji desktopowej umożliwiającej dostęp do dodanej logiki biznesowej, działającej na platformie Java EE Uwaga: działania wg instrukcji dodanej do tego etapu projektu 1) Sprawdzanie poprawności oprogramowania wg Instrukcji do lab 10 oraz Instrukcji do lab 11.		Eliminacja redundancji w projekcie. Udział w pracach projektowych i implementacji
13- tydzień 14- tydzień 15- tydzień 18.05.2018-8.06.2018	Kod programu należy wysłać do dnia 1.05.2018, drugi wg p.2 do 4.06.2018	Sprint planning meeting (45 min)	1) Dodanie warstwy integracji z bazą danych opartą na technologii JPA (Java Persistence API) do programu oparta na automatycznym generowaniu kontrolerów typu Session Bean For Entity Class **Uwaga: działania wg bieżącej instrukcji do tego etapu projektu** 1) Uzupełnie aplikacji desktopowej do wywołanie operacji na bazie danych w technologii ORM. 2) Zaliczenie projektu		Eliminacja redundancji w projekcie. Udział w pracach projektowych i implementacji _

II. Kontynuacja programu wykonanego wg instrukcji Projekt_INP002017_1 (wg materiałów do wykładu 7 Wyklad_INP002017_7_2.pdf – p.6-8)

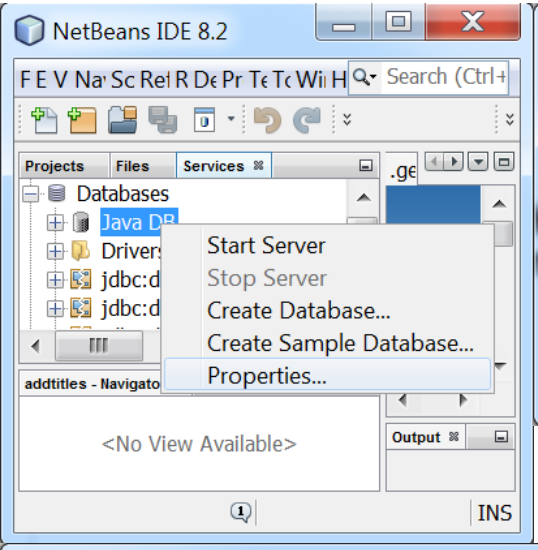
1. **Warstwa zasobów (EIS) - baza danych w systemie baz danych Derby**
2. **Utworzenie obiektowego modelu danych do utrwalania ORM**
3. **Warstwa integracji. Zastosowanie wzorca projektowego typu *Domain Store* w technologii JPA (Java Persistence) na platformie Java EE**

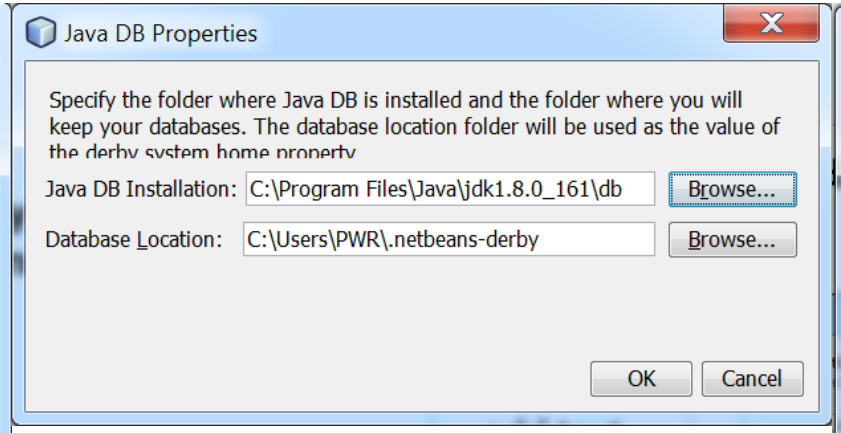
Zagadnienia

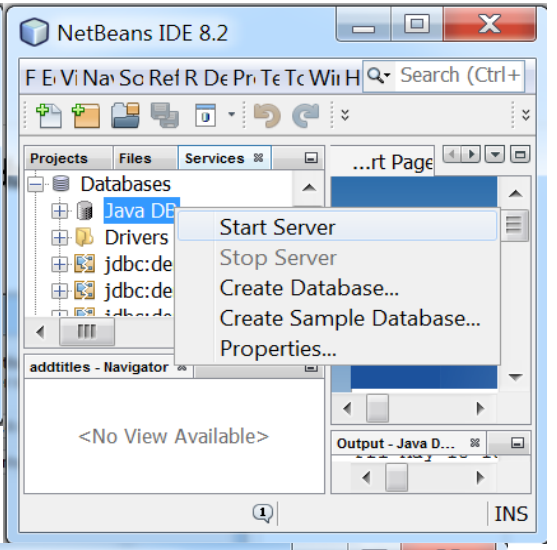
1. **Warstwa zasobów (EIS)- baza danych w systemie baz danych Derby**

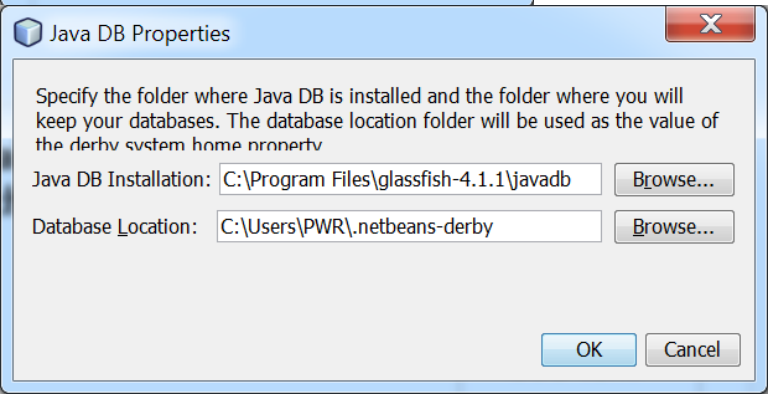
Przed założeniem bazy danych należy sprawdzić, czy silnik bazy danych Java DB uruchamia się

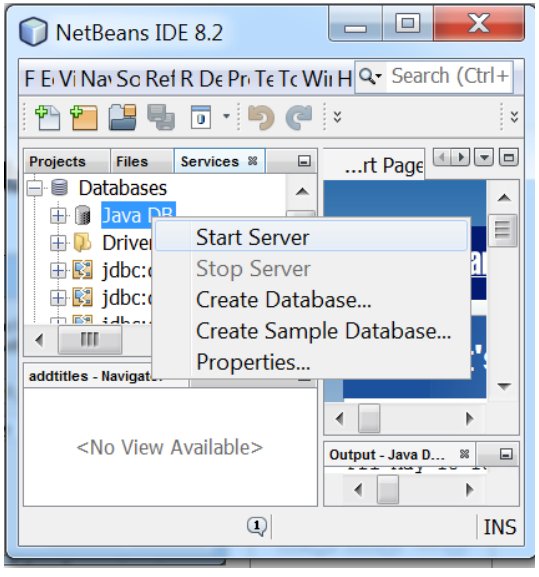
1. Należy w zakładce **Services** rozwinąć pozycję **Databases** i kliknąć na pozycję **Start Server.**
2. Jeżeli server **Java DB** nie wystartuje (przejście do pozycji **Stop Server**), należy wybrać pozycję **Properties**, wybrać pole **Java DB Instalation** i nacisnąć na klawisz **Browse...** Wtedy można wybrać instalację javadb w **C:\Program Files\glasfish-4.1.1\javadb** i ponowić uruchomienie silnika Java DB. Proba powinna być udana

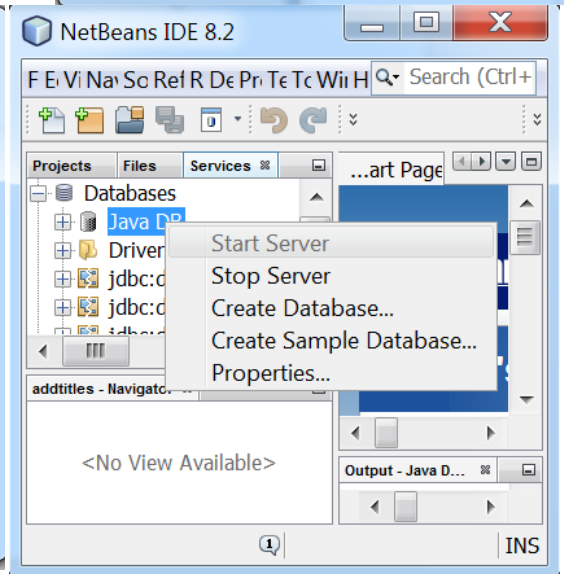




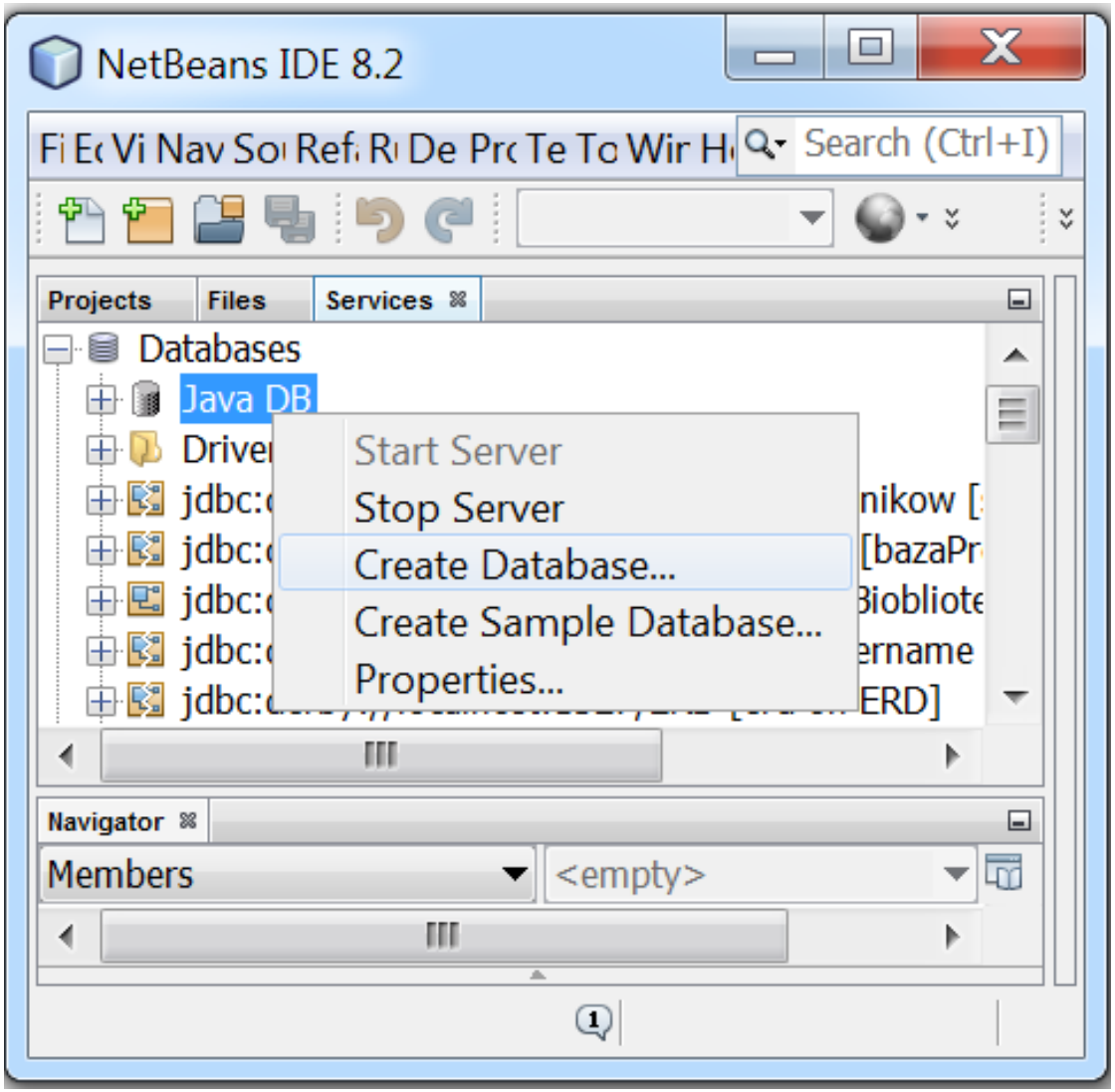




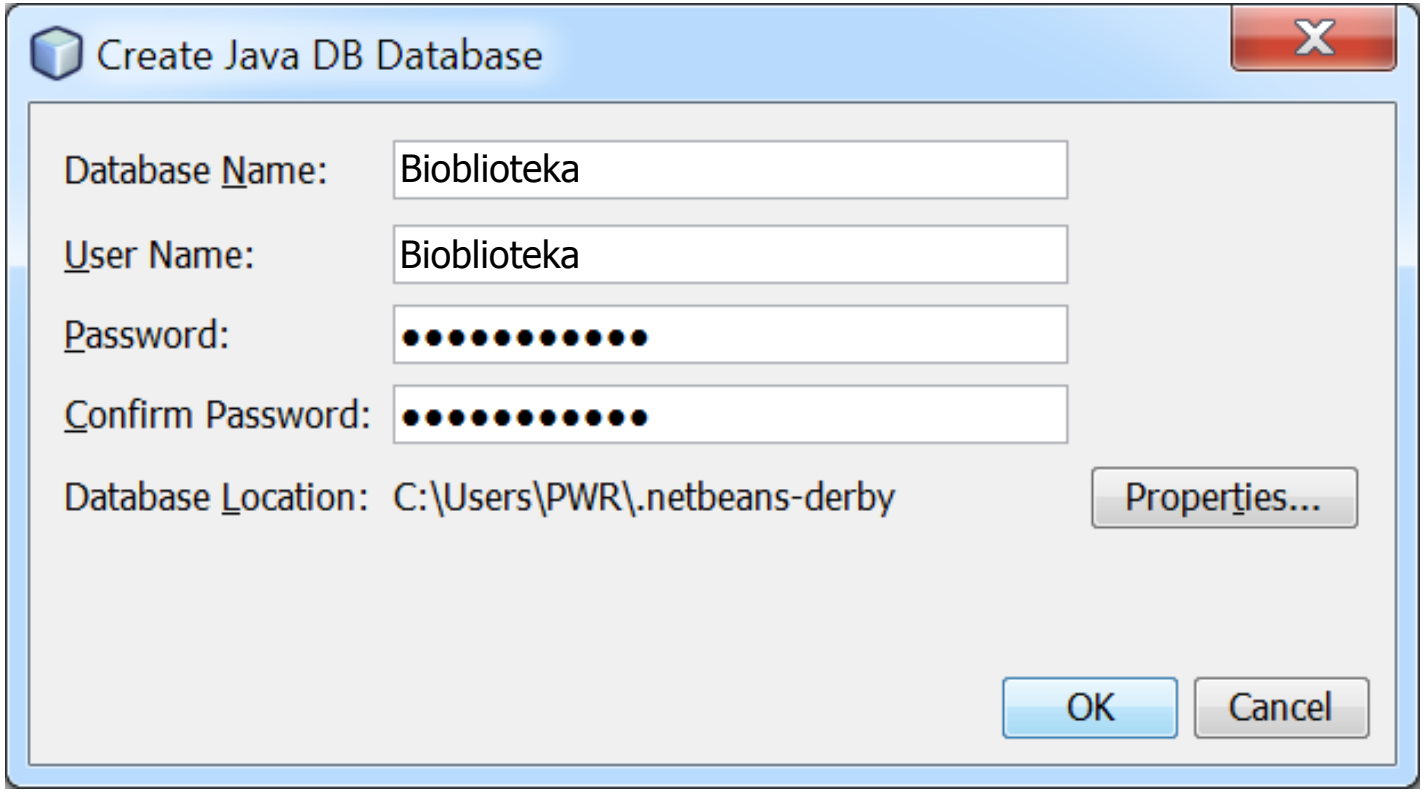




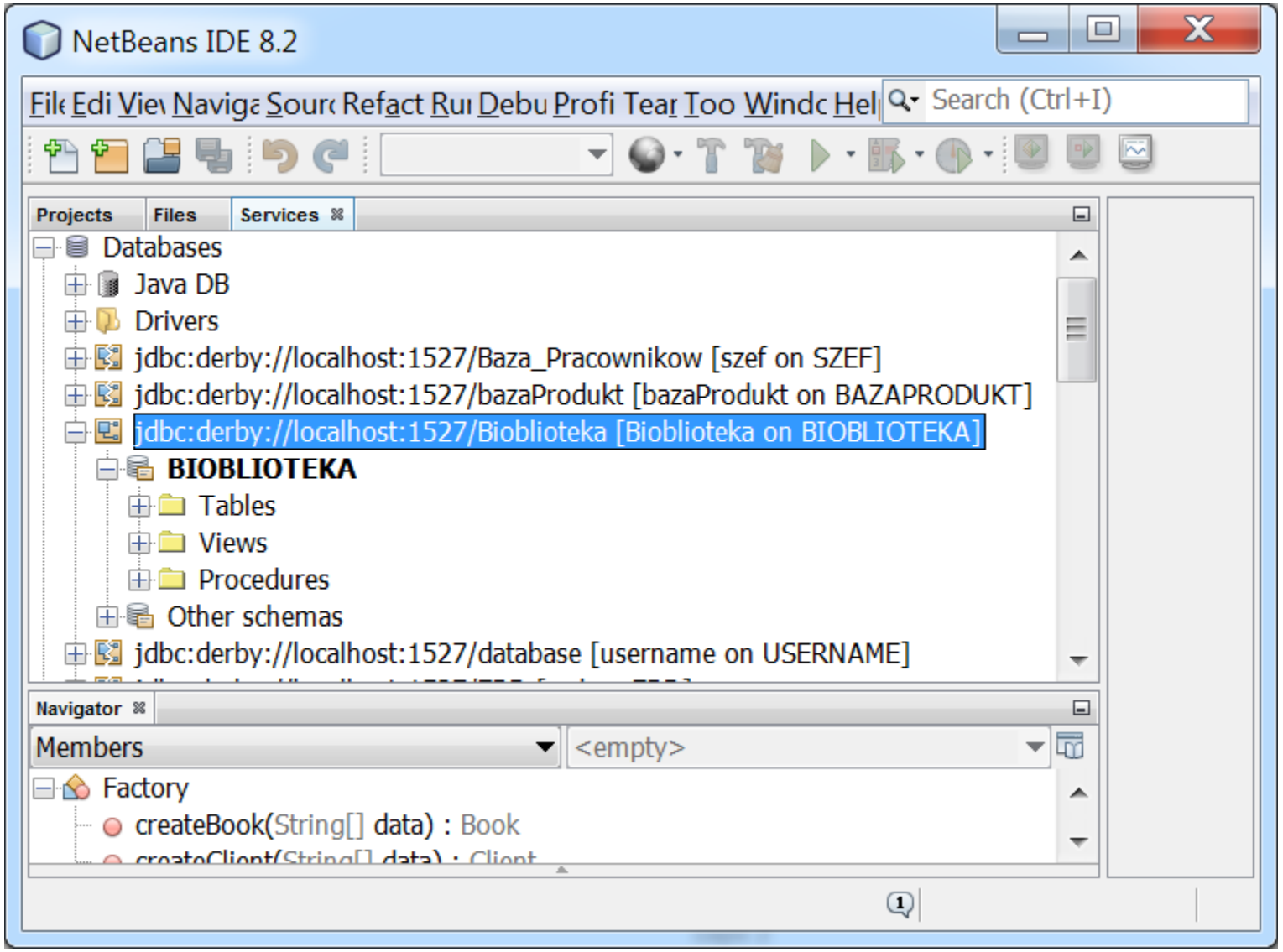
1. Zakładanie pustej bazy danych dla systemu baz danych Derby



2. Zakładanie pustej bazy danych Bioblioteka w systemie baz danych Derby



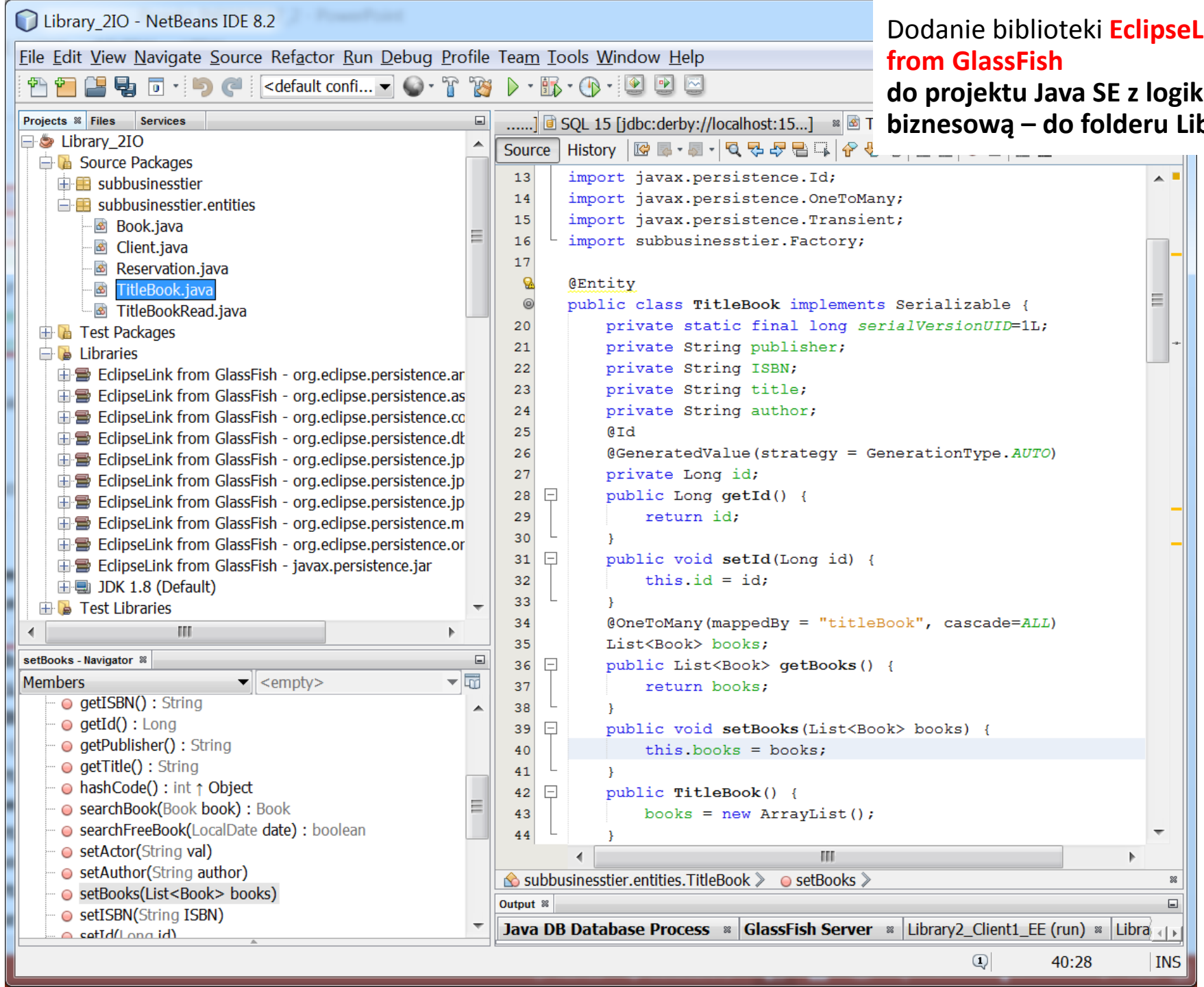
3. Utworzona pusta baza danych. W celu połączenia z bazą danych należy kliknąć na nazwę bazy danych i wybrać pozycję Connect...



Zagadnienia

1. **Warstwa zasobów (EIS)- baza danych w systemie baz danych Derby**
2. **Utworzenie obiektowego modelu danych do utrwalania ORM**

Dodanie biblioteki **EclipseLink from GlassFish**
do projektu Java SE z logiką biznesową – do folderu Libraries

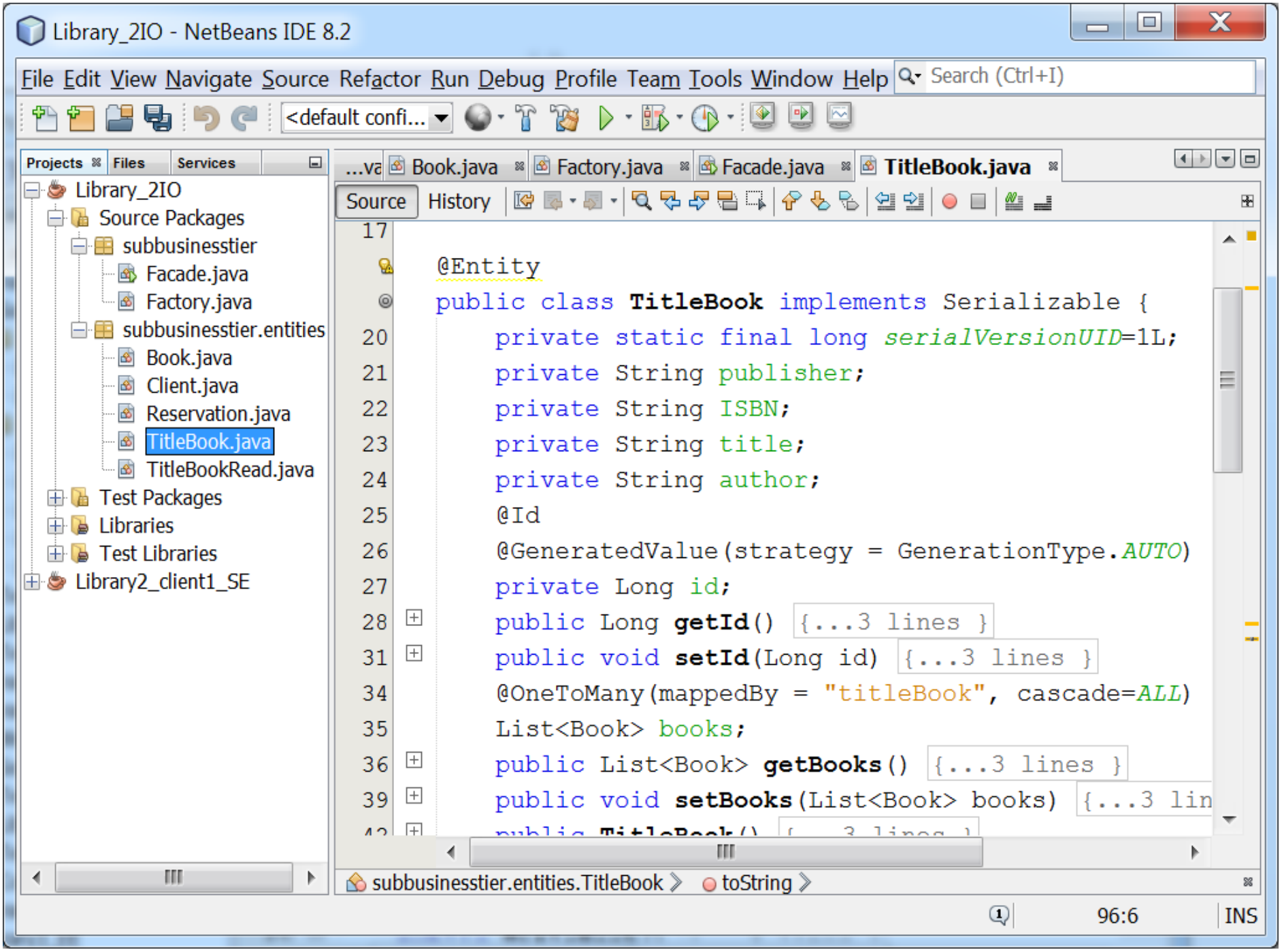


```java
@Entity
public class TitleBook implements Serializable {
    private static final long serialVersionUID=1L;
    private String publisher;
    private String ISBN;
    private String title;
    private String author;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    public Long getId() {...3 lines }
    public void setId(Long id) {...3 lines }
    @OneToMany(mappedBy = "titleBook", cascade=ALL)
    List<Book> books;
    public List<Book> getBooks() {...3 lines }
    public void setBooks(List<Book> books) {...3 lines }
    public TitleBook() {...3 lines }
    public String getPublisher() {...3 lines }
    public void setPublisher(String publisher) {...3 lines }
    public String getISBN() {...3 lines }
    public void setISBN(String ISBN) {...3 lines }
    public String getTitle() {...3 lines }
    public void setTitle(String title) {...3 lines }
    public String getAuthor() {...3 lines }
    public void setAuthor(String author) {...3 lines }
    public String getActor() {...3 lines }
    public void setActor(String val) {...2 lines }
```

Zmiana typu klas danych na typ @Entity w projekcie warstwy biznesowej – dodano adnotacje, nowy atrybut Id. Należy zestandaryzować nazwy metod dostępu do składowych klasy typu Entity.

```
package subbusinesstier.entities;

import ...

@Entity
public class TitleBookRead extends TitleBook {
private static final long serialVersionUID=1L;
    private String actor;

    @Override
    public String getActor() {...3 lines }

    @Override
    public void setActor(String actor) {...3 lines }

    @Override
    public String toString() {...5 lines }

}
```

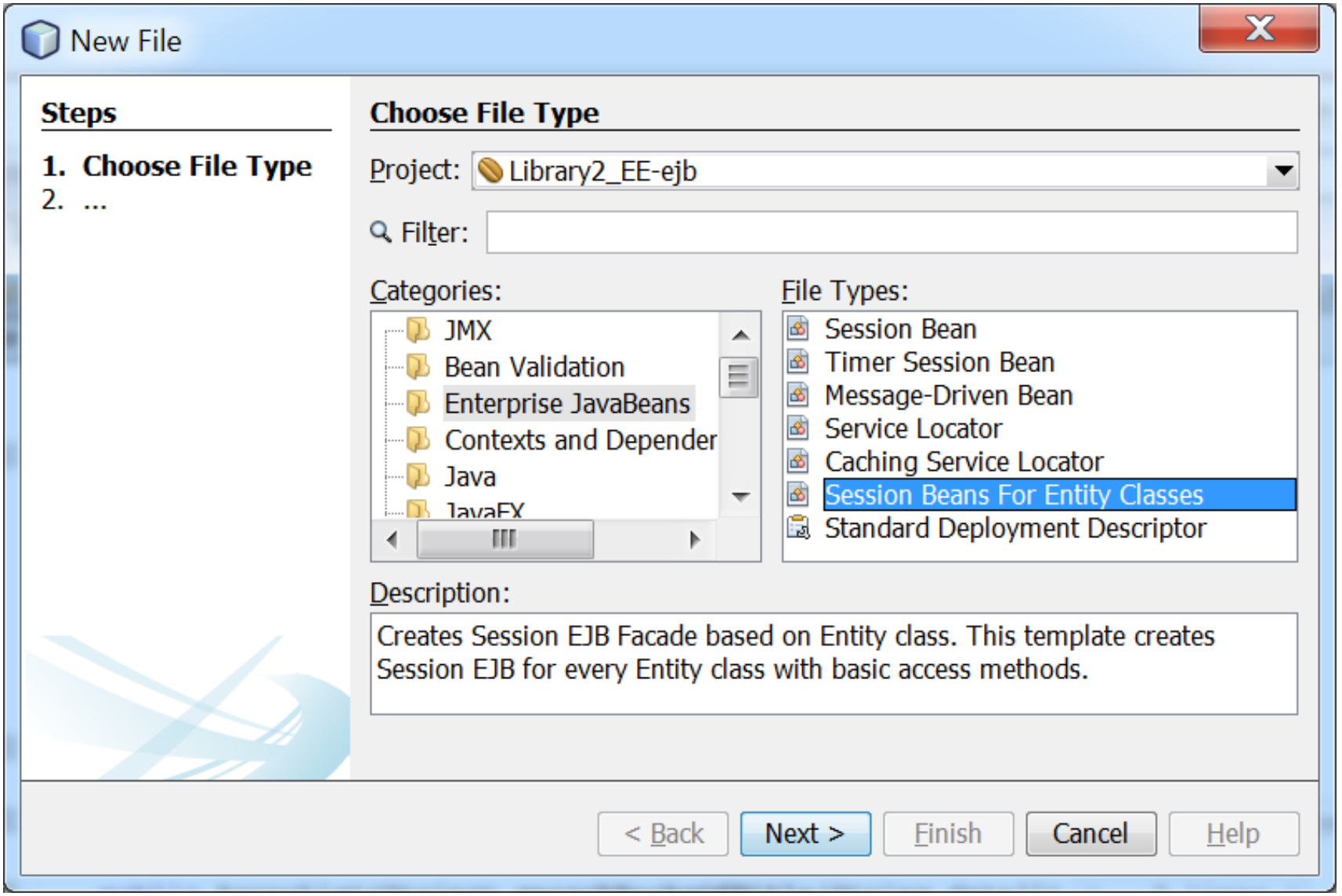
```java
@Entity
public class Book implements Serializable {
private static final long serialVersionUID=1L;
    private int number;
    @ManyToOne
    private TitleBook titleBook;
   // @OneToMany (mappedBy="book")
    @Transient
    private List<Reservation> reservations;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    public Long getId() {...3 lines }
    public void setId(Long id) {...3 lines }

    public List<Reservation> getReservations() {...3 lines }
    public void setReservations(List<Reservation> reservations) {...3 lines }
```


Library_2IO - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default confi...
Projects
Files
Services
Library_2IO
Source Packages
subbusinesstier
Facade.java
Factory.java
subbusinesstier.entities
Book.java
Client.java
Reservation.java
TitleBook.java
TitleBookRead.java
Test Packages
Libraries
Test Libraries
Library2_client1_SE
Book.java
Factory.java
Facade.java
TitleBook.java
Source
History
17
@Entity
public class TitleBook implements Serializable {
20 private static final long serialVersionUID=1L;
21 private String publisher;
22 private String ISBN;
23 private String title;
24 private String author;
25 @Id
26 @GeneratedValue(strategy = GenerationType.AUTO)
27 private Long id;
28 public Long getId() {...3 lines }
31 public void setId(Long id) {...3 lines }
34 @OneToMany(mappedBy = "titleBook", cascade=ALL)
35 List<Book> books;
36 public List<Book> getBooks() {...3 lines }
39 public void setBooks(List<Book> books) {...3 lin
subbusinesstier.entities.TitleBook
toString
96:6
INS

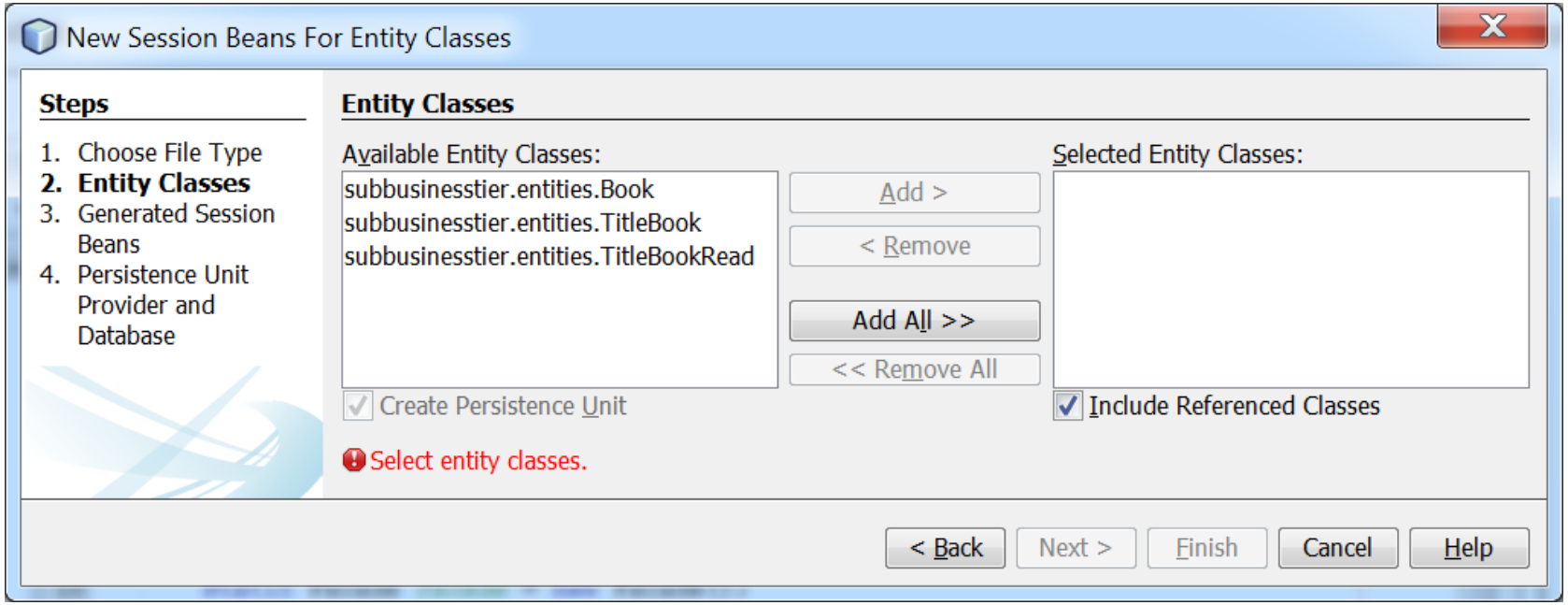
Zagadnienia

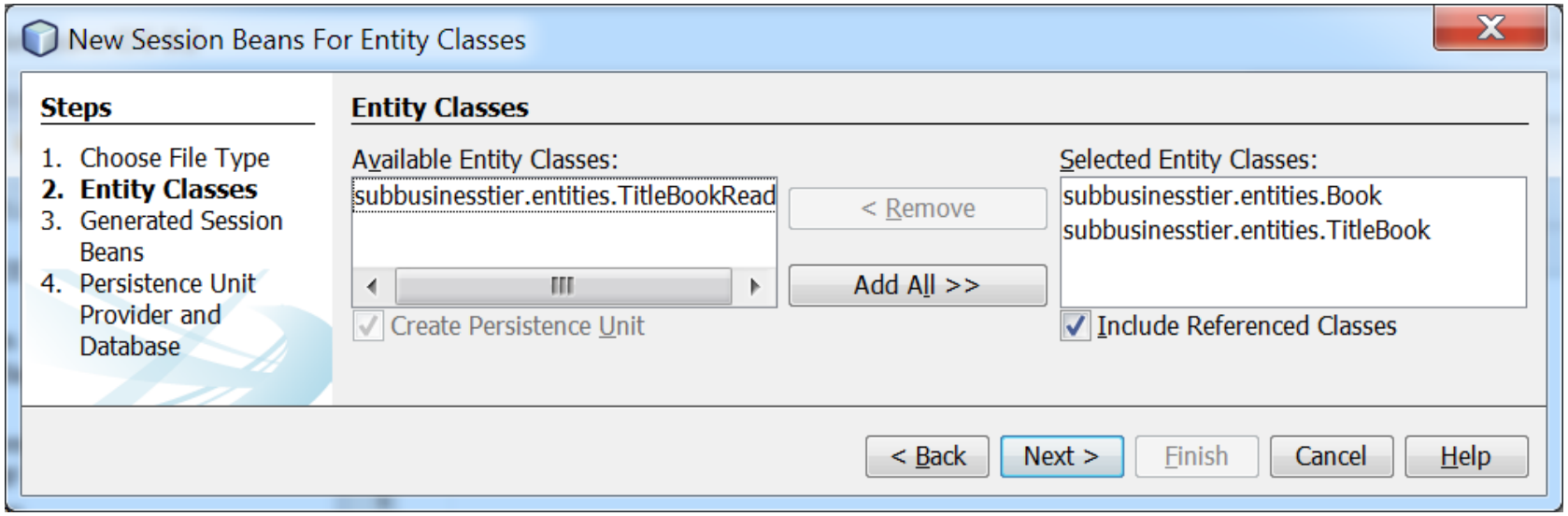
1. Warstwa zasobów (EIS)- baza danych w systemie baz danych Derby
2. Utworzenie obiektowego modelu danych do utrwalania ORM
3. Warstwa integracji. Zastosowanie wzorca projektowego typu *Domain Store* w technologii JPA (Java Persistence) na platformie Java EE

1. Należy w projekcie typu moduł EJB dodać komponenty typu Session Bean for Entity Class - po kliknięciu na nazwę projektu w zakładce Projects wybrać New/Other../Enterprise JavaBeans/Session Beans For Entity Classes nacisnąć na klawisz Next

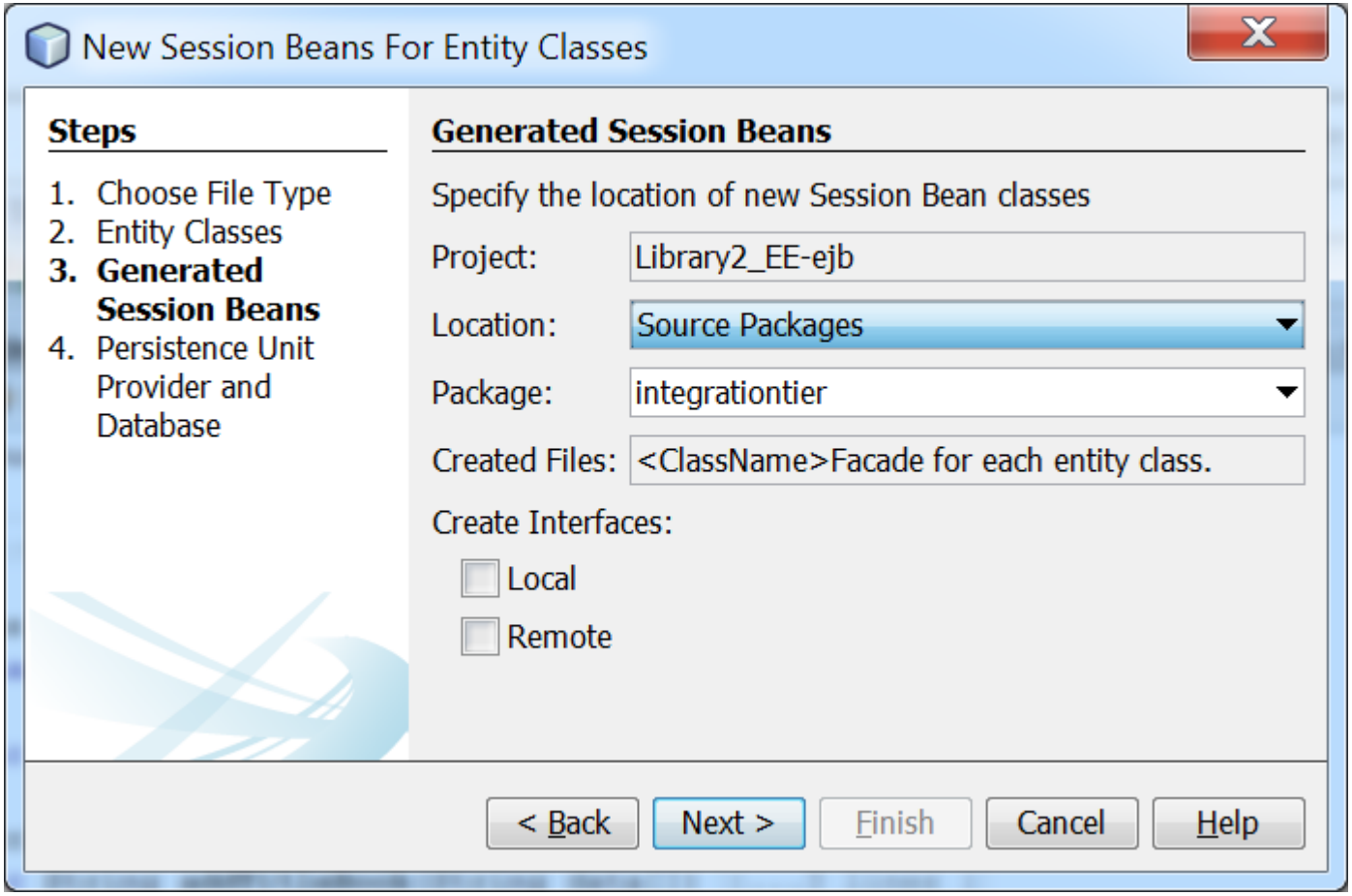


2. Należy wybrać z listy encje TitileBook i Book i kliknąć na klawisz Next

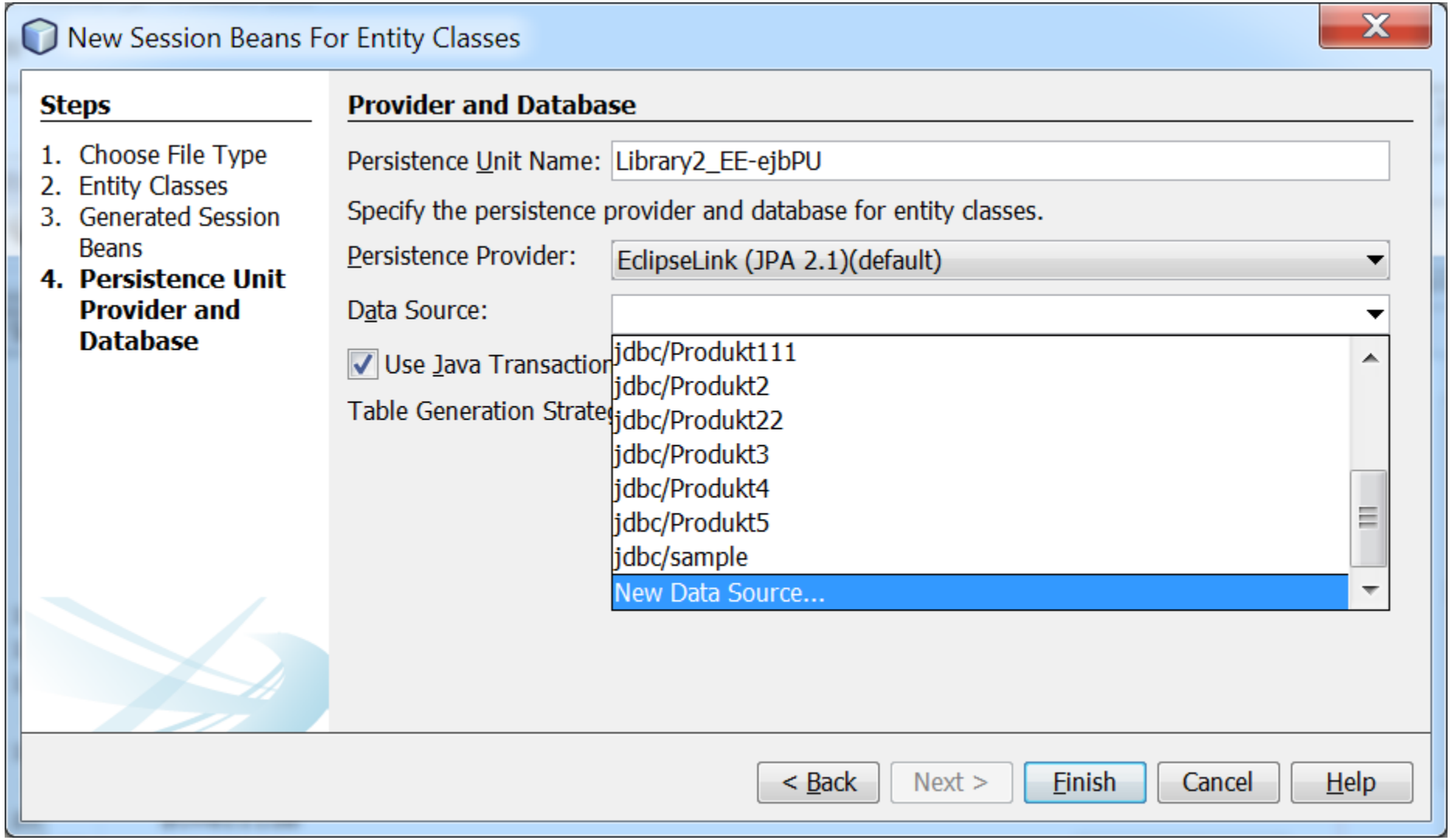




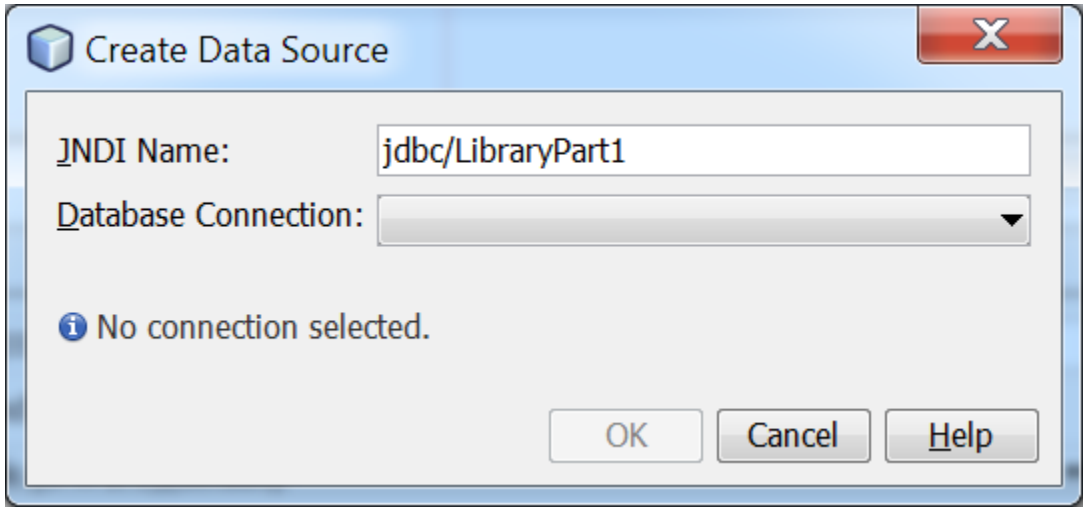
3. W nowym pakiecie np integrationtier zostaną wygenerowane komponenty po naciśnięciu na klawisz Next

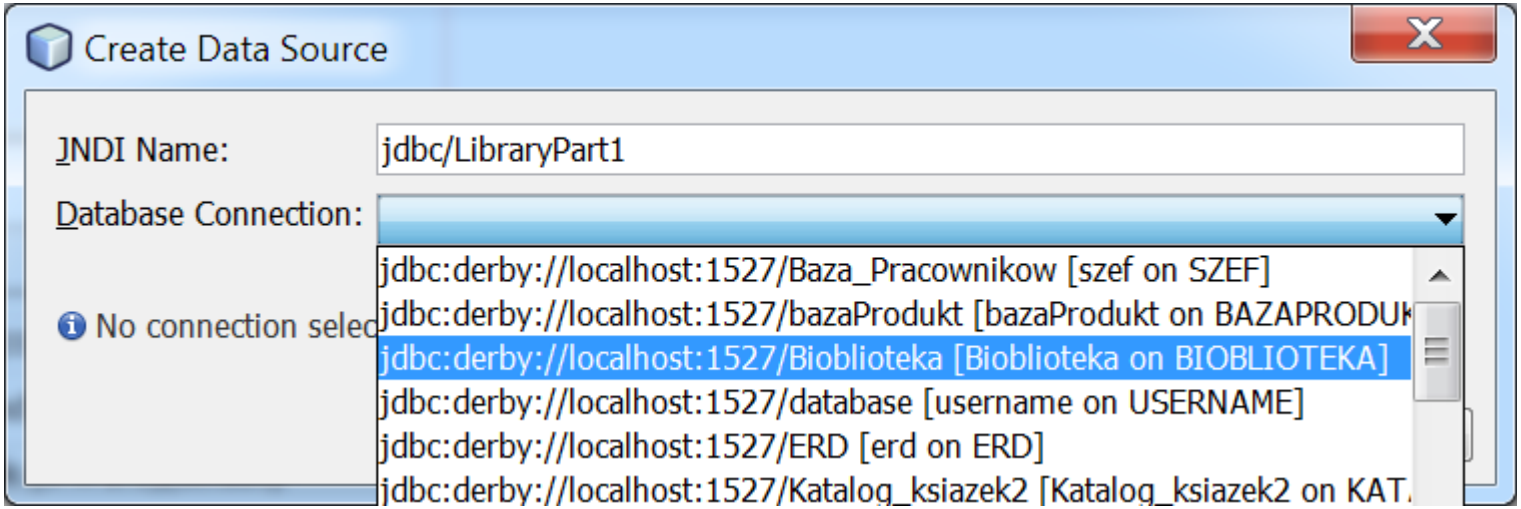


4. W kolejnym kroku należy utworzyć połączenie typu JNDI z bazą danych Biblioteka – należy z listy wybrać pozycję New Data Source...

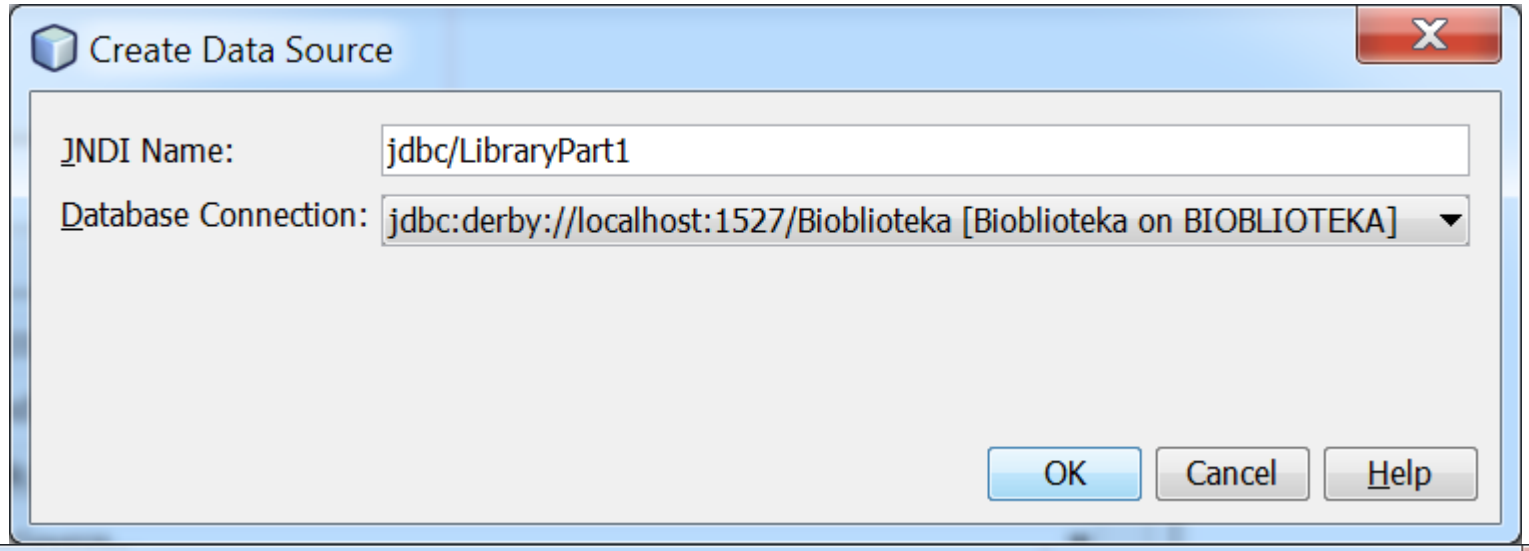


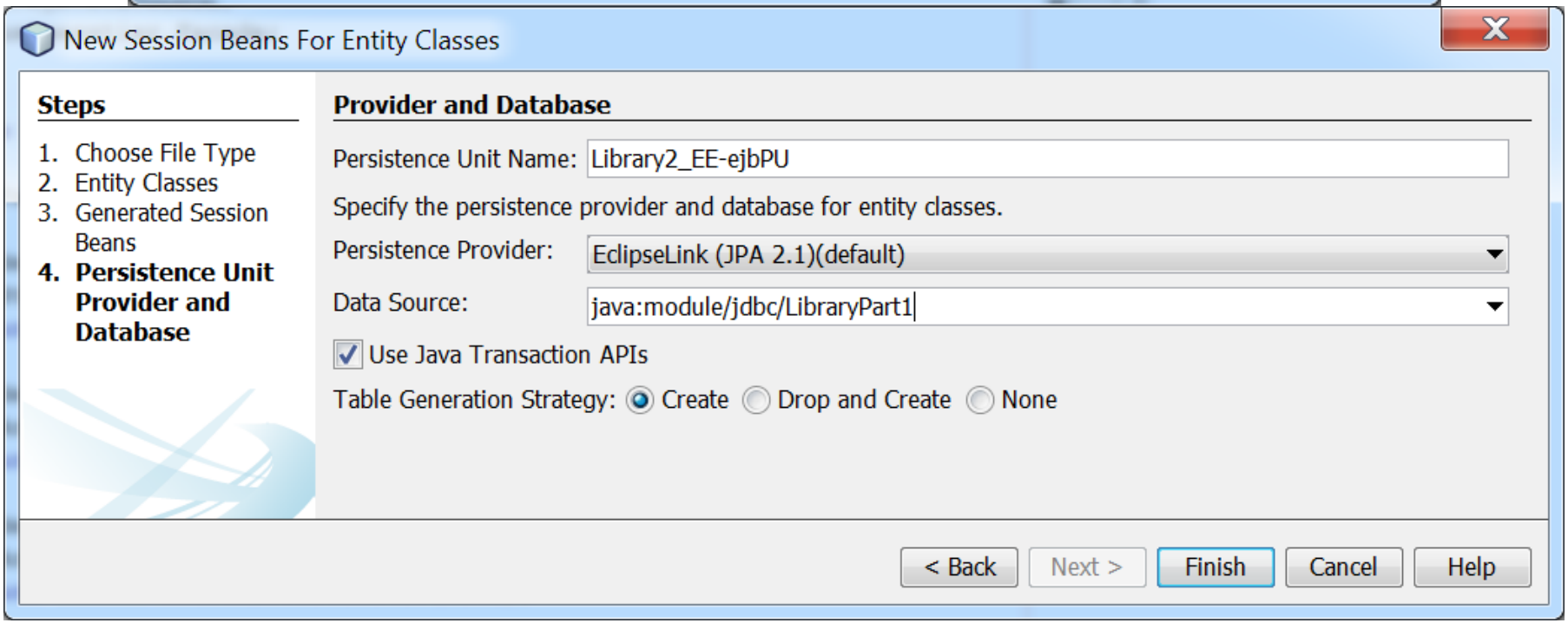
5. Wypełnić pole JNDI Name, a z listy Database Connection należy wybrać bazę danych Bioblioteka, założoną wcześniej



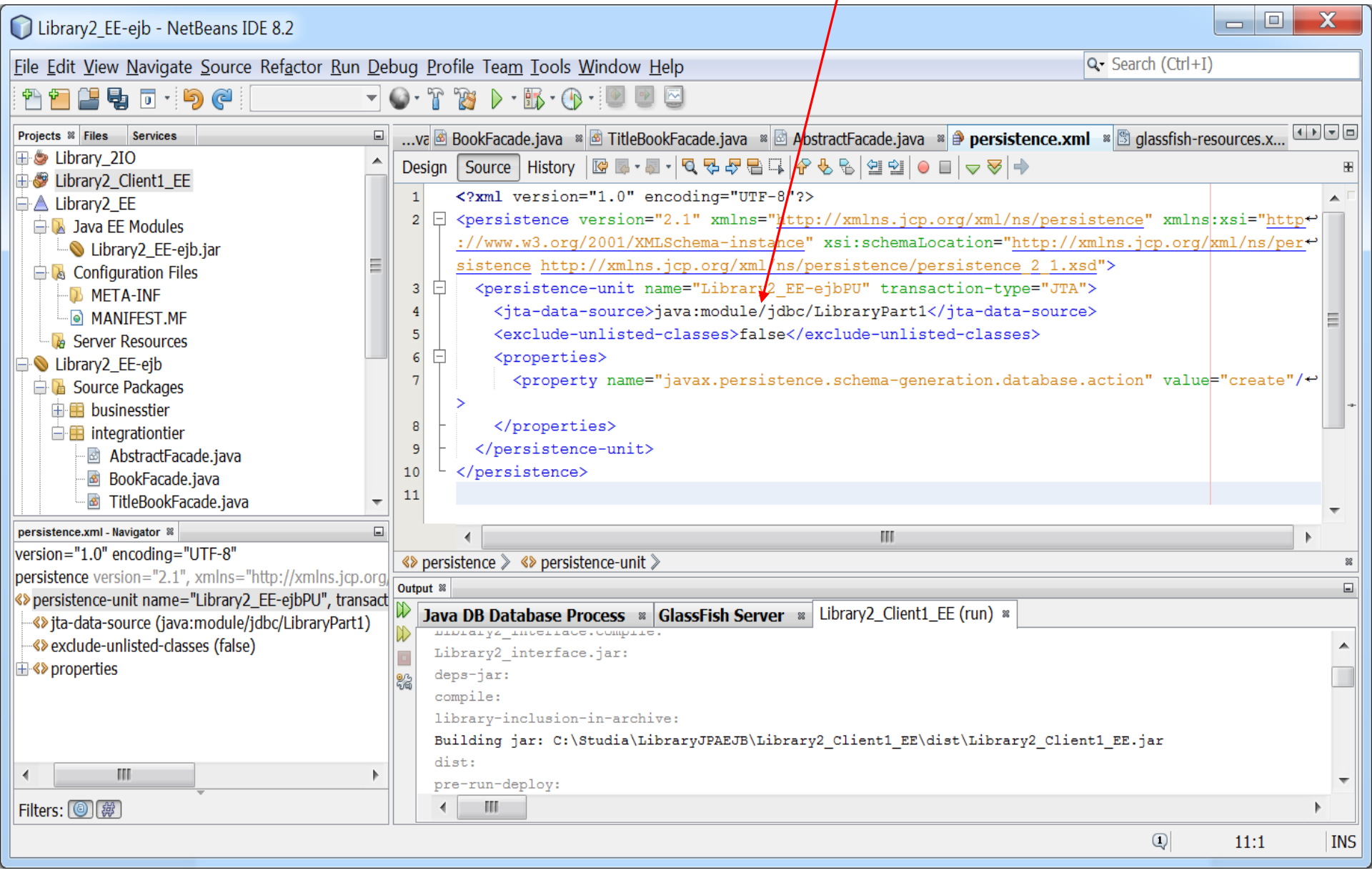


6. Widok po utworzeniu połączenia JNDI i kliknięcia na OK, w polu Data Source połączenie ma zasięg modułu – należy zmodyfikkować to połączenie na globalne (verte)

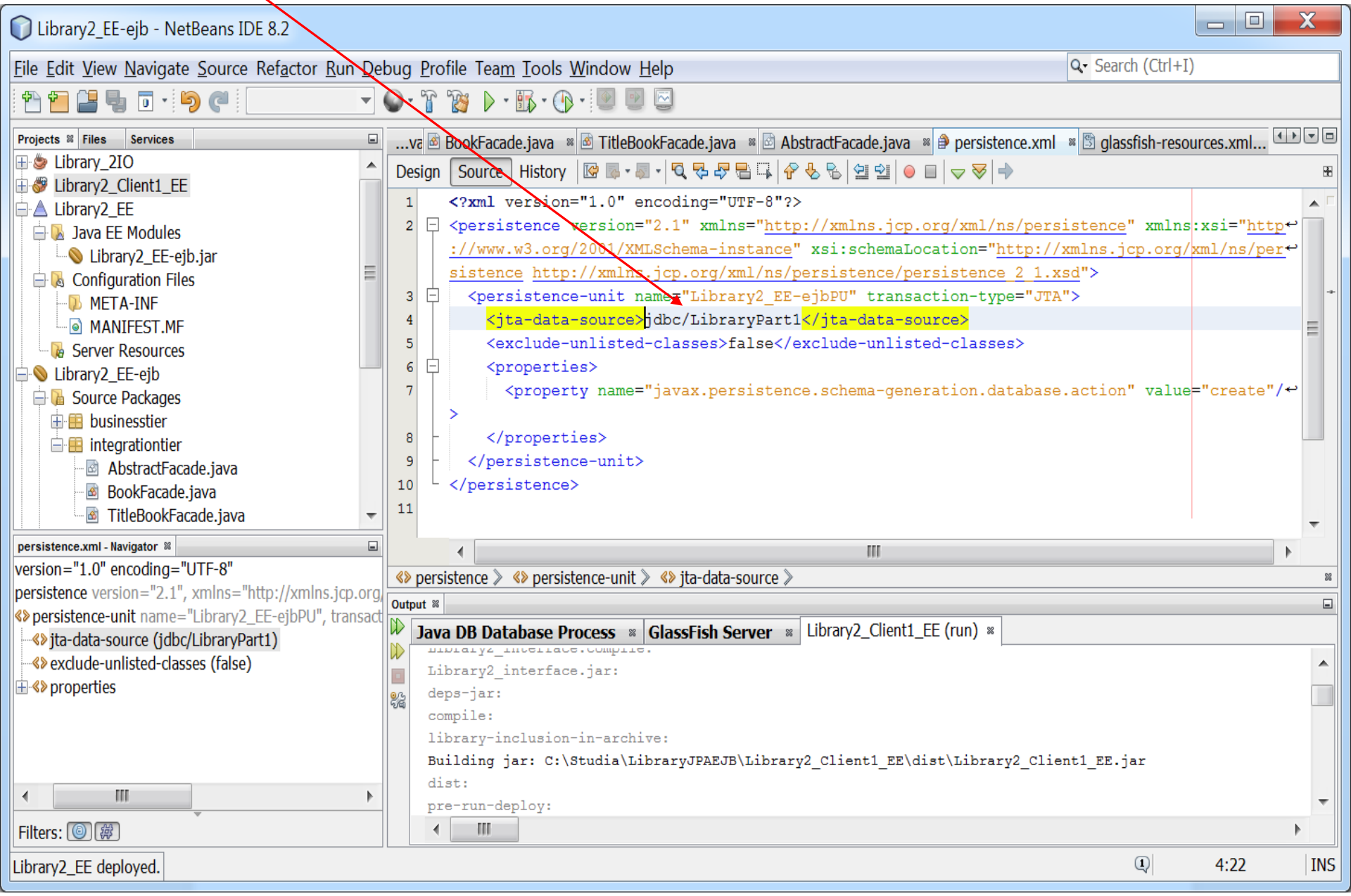




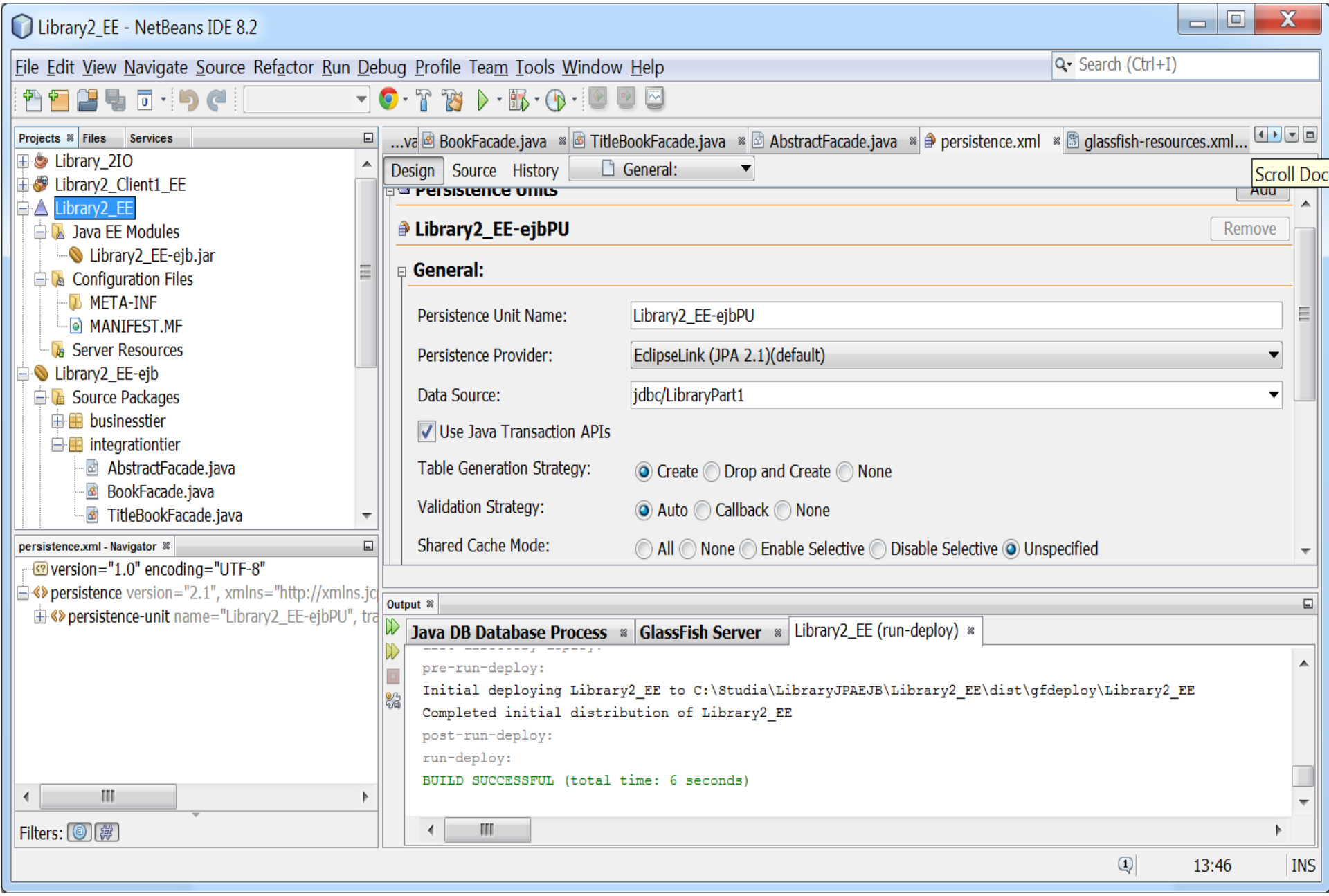
7. Należy wybrać zakładkę Source i w kodzie źródłowym pliku persistence.xml zmodyfikować wartość znacznika <jta-data-source> na jdbc/LibraryPart1



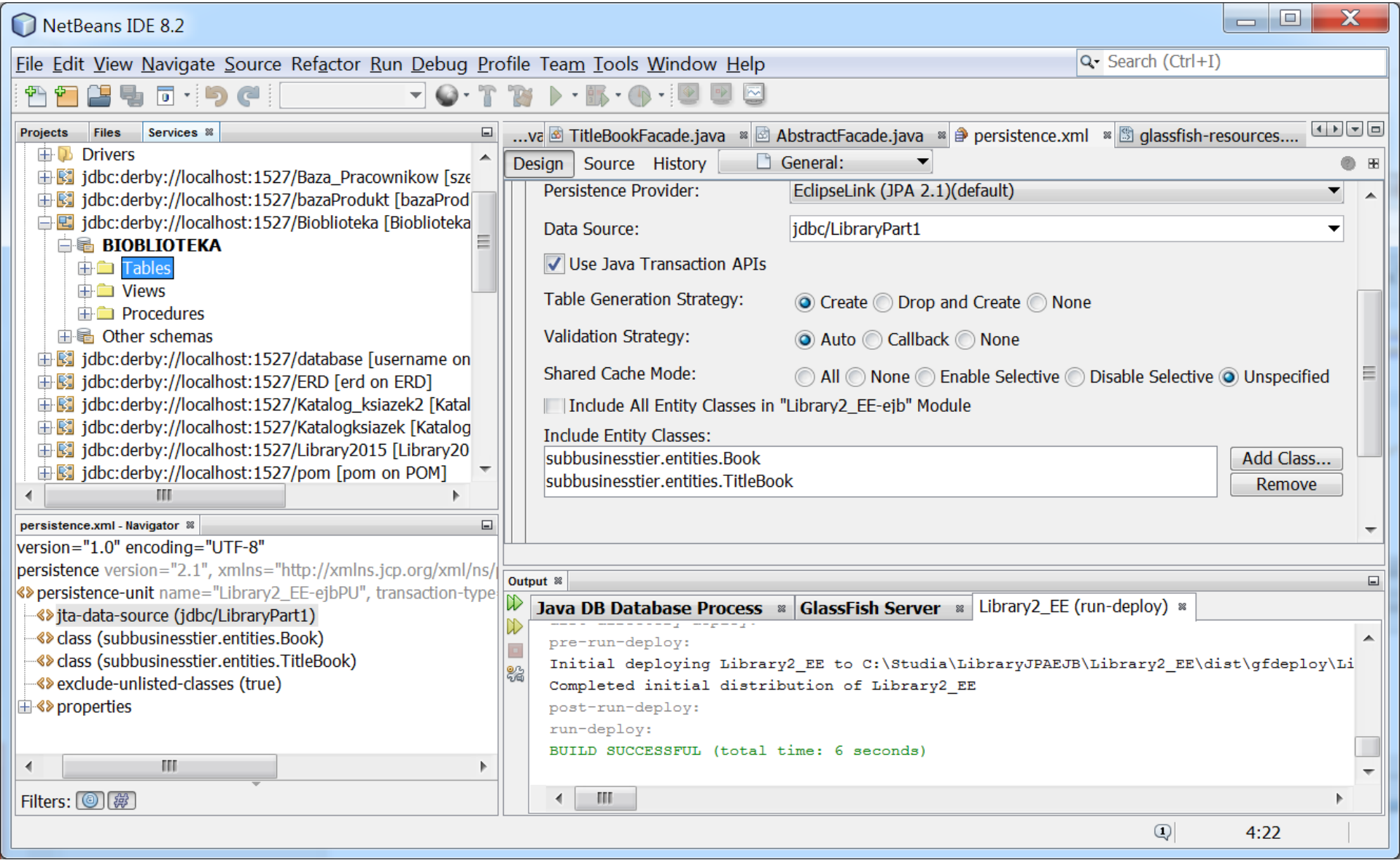
8. Rezultat


Library2_EE-ejb - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Files Services
Library_2IO
Library2_Client1_EE
Library2_EE
Java EE Modules
Library2_EE-ejb.jar
Configuration Files
META-INF
MANIFEST.MF
Server Resources
Library2_EE-ejb
Source Packages
businesstier
integrationtier
AbstractFacade.java
BookFacade.java
TitleBookFacade.java
persistence.xml - Navigator
version="1.0" encoding="UTF-8"
persistence version="2.1", xmlns="http://xmlns.jcp.org/
persistence-unit name="Library2_EE-ejbPU", transact
jta-data-source (jdbc/LibraryPart1)
exclude-unlisted-classes (false)
properties
Filters:
...va BookFacade.java TitleBookFacade.java AbstractFacade.java persistence.xml glassfish-resources.xml...
Design Source History
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
3 <persistence-unit name="Library2_EE-ejbPU" transaction-type="JTA">
4 <jta-data-source>jdbc/LibraryPart1</jta-data-source>
5 <exclude-unlisted-classes>false</exclude-unlisted-classes>
6 <properties>
7 <property name="javax.persistence.schema-generation.database.action" value="create"/>
8 </properties>
9 </persistence-unit>
10 </persistence>
11
persistence > persistence-unit > jta-data-source >
Output
Java DB Database Process GlassFish Server Library2_Client1_EE (run)
Library2_interface.jar:
deps-jar:
compile:
library-inclusion-in-archive:
Building jar: C:\Studia\LibraryJPAEJB\Library2_Client1_EE\dist\Library2_Client1_EE.jar
dist:
pre-run-deploy:
Library2_EE deployed.
4:22
INS

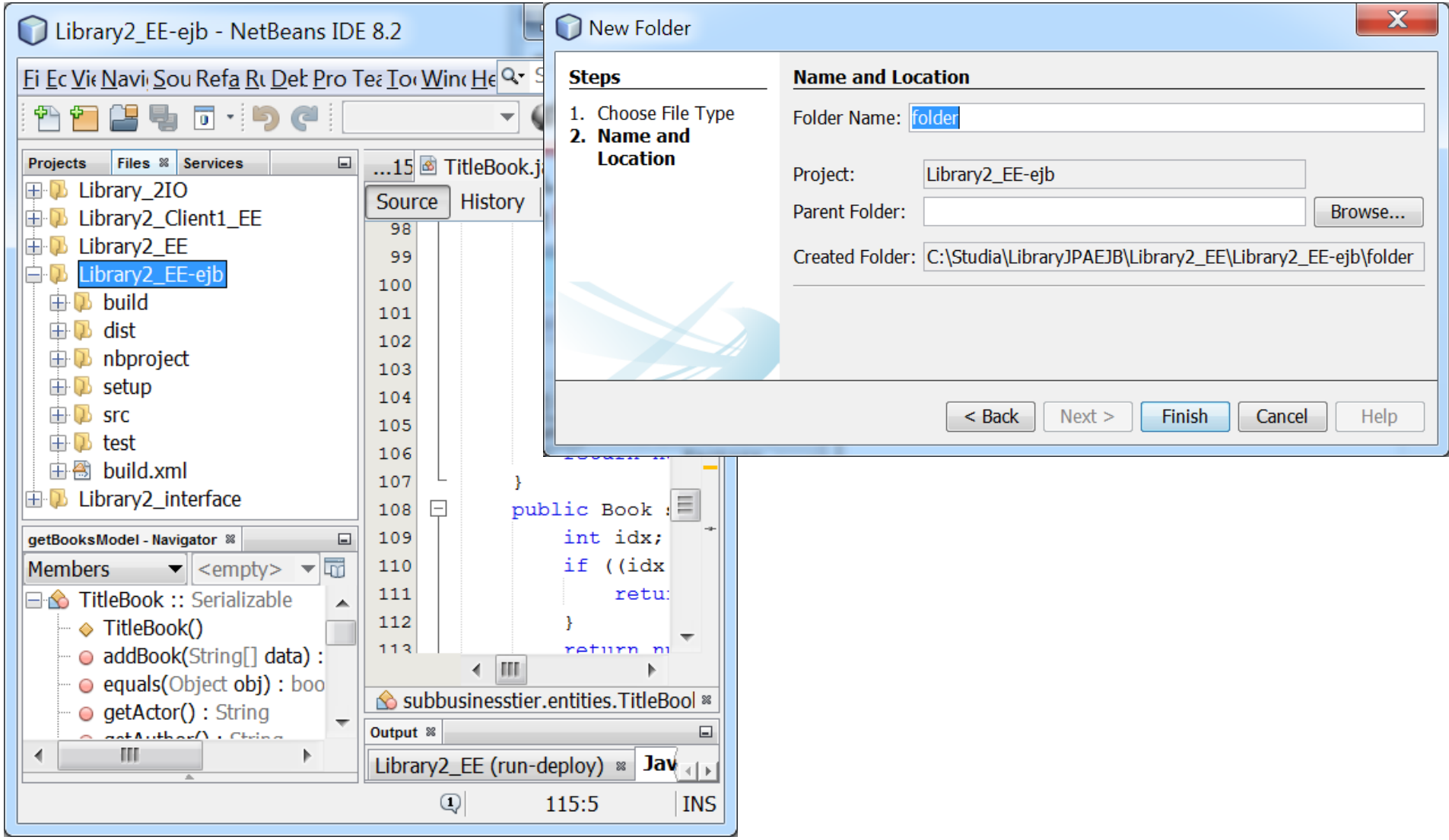
9. Widok pliku persistence.xml po przejściu do widoku Design



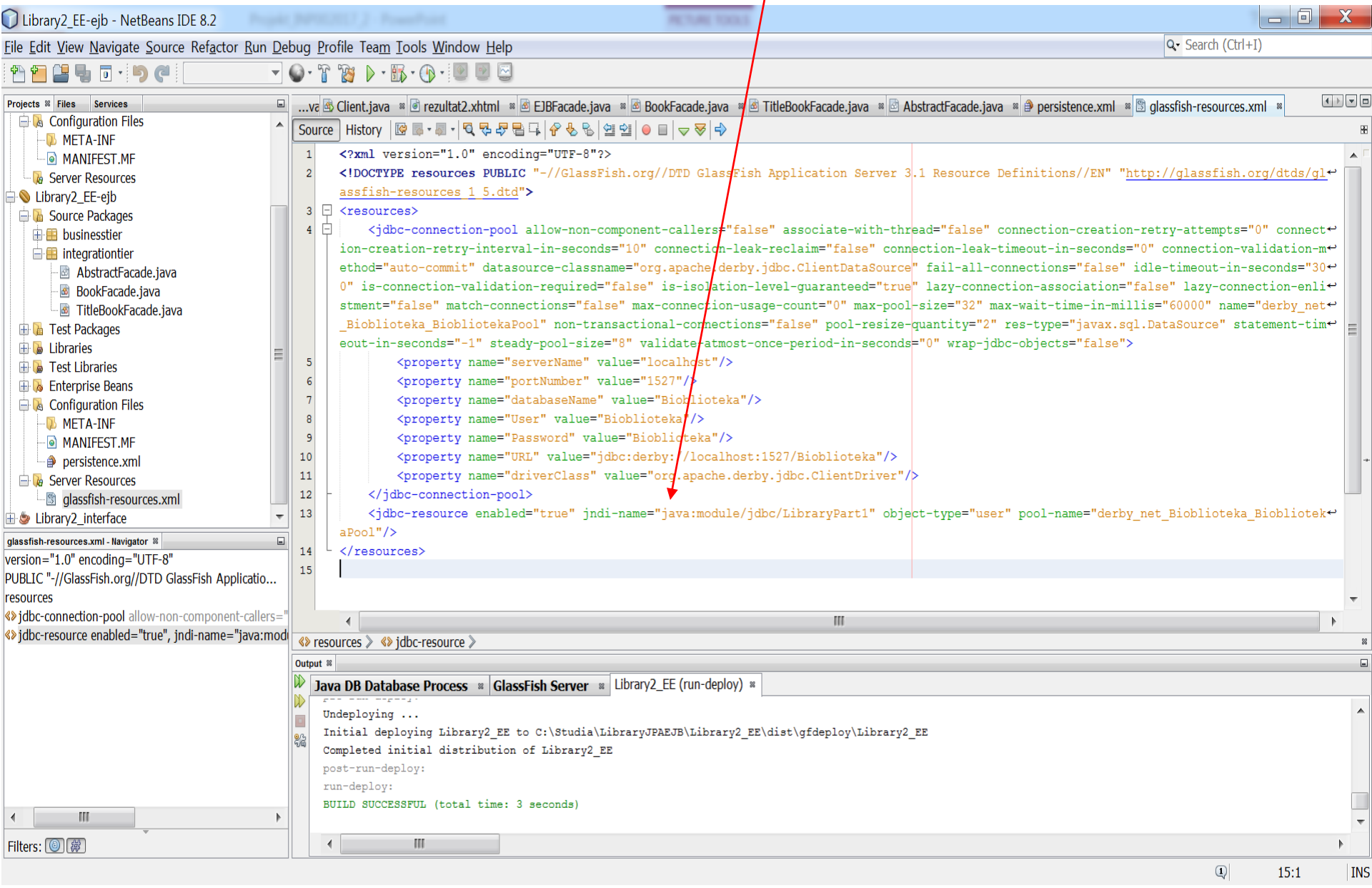
10. Należy usunąć zaznaczenie z pola Include All Entity Classes... i wybrać z listy Add Class... encje TitleBook i Book. Spowoduje to w kolejnych krokach wygenerowanie jedynie dwóch tabel w bazie danych: TitleBook i Book. Tabela TitleBook będzie zawierała dane obiektów typu TitleBook i TitleBookRead



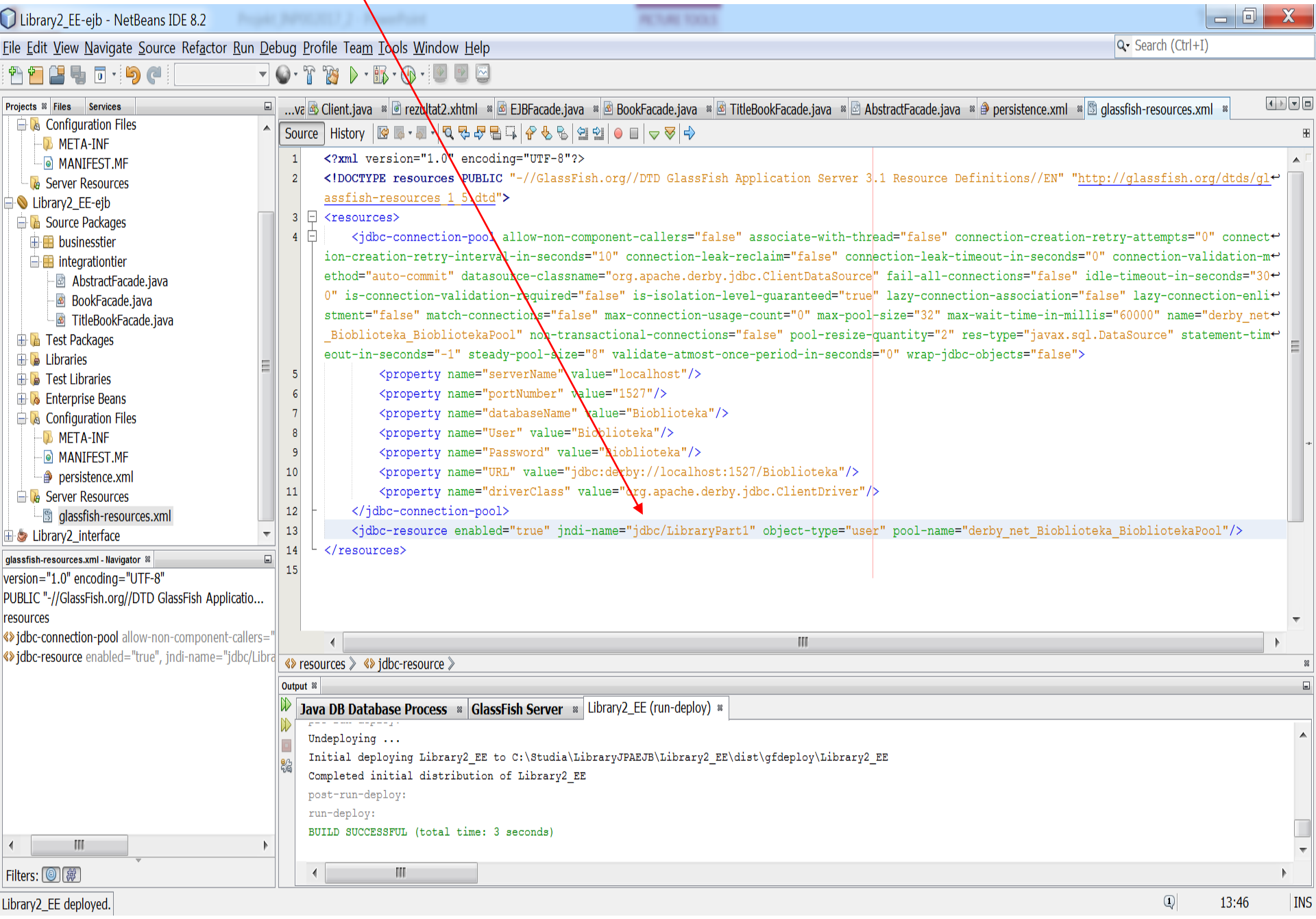
11. Należy „przesunąć” plik glassfish-resources.xml z folderu Configuration Files/Meta-INF do folderu Server Resources. Przed wykonaniem należy w zakładce Files sprawdzić, czy istnieje folder setup w module Library2_EE-ejb. Jeśli nie, należy go utworzyć (po zaznaczeniu projektu należy wybrać z listy pozycję New/Folder i założyć folder o nazwie setup).



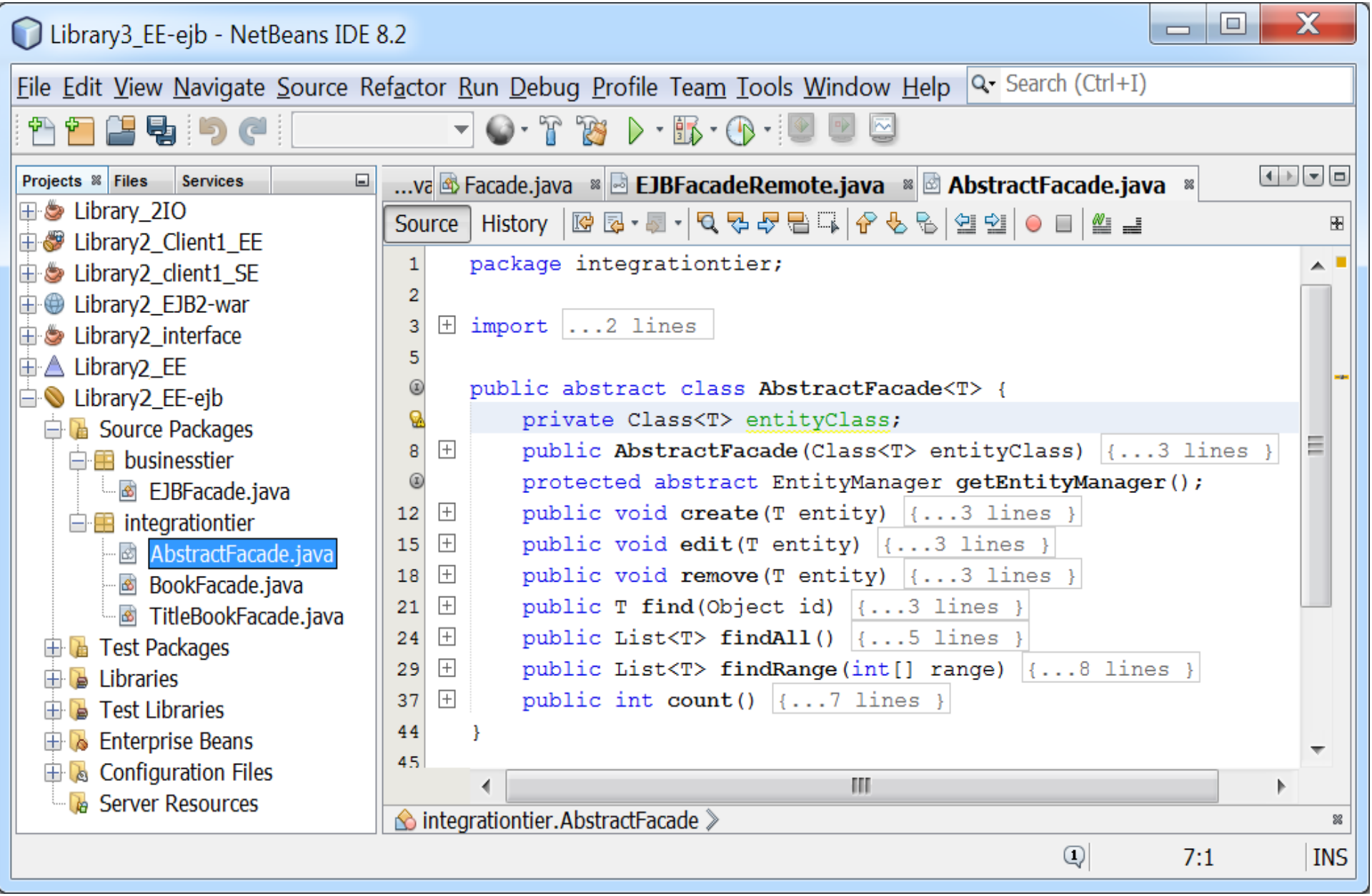
12. Należy „przesunąć” plik glassfish-resotces.xml z folderu Configuration Files/Meta-INF do folderu Server Resources. Następnie, należy zmienić atrybut jndi-name w znaczniku <jdbc-resources na jdbc/LibraryPart1



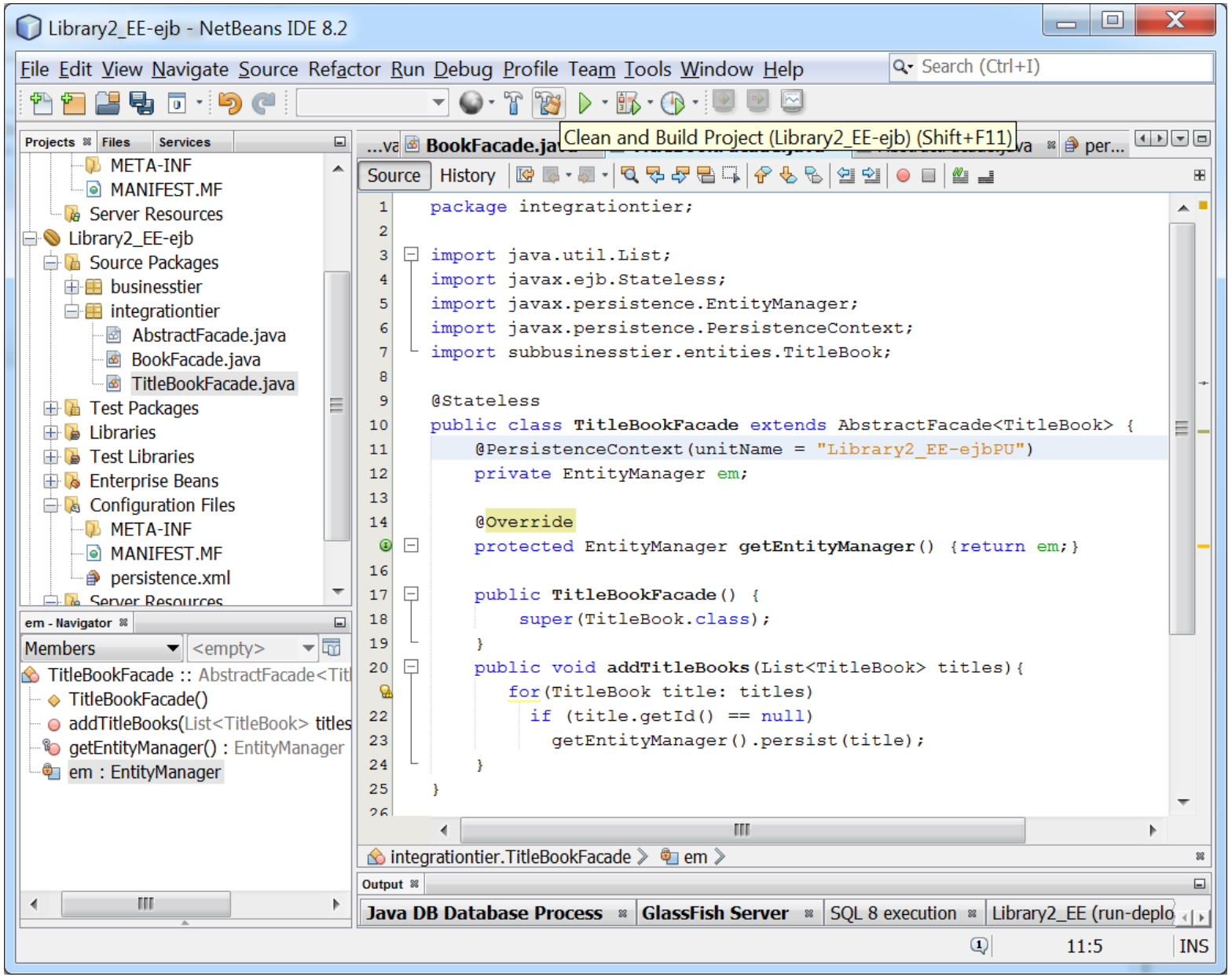
13. Wynik po zmianie



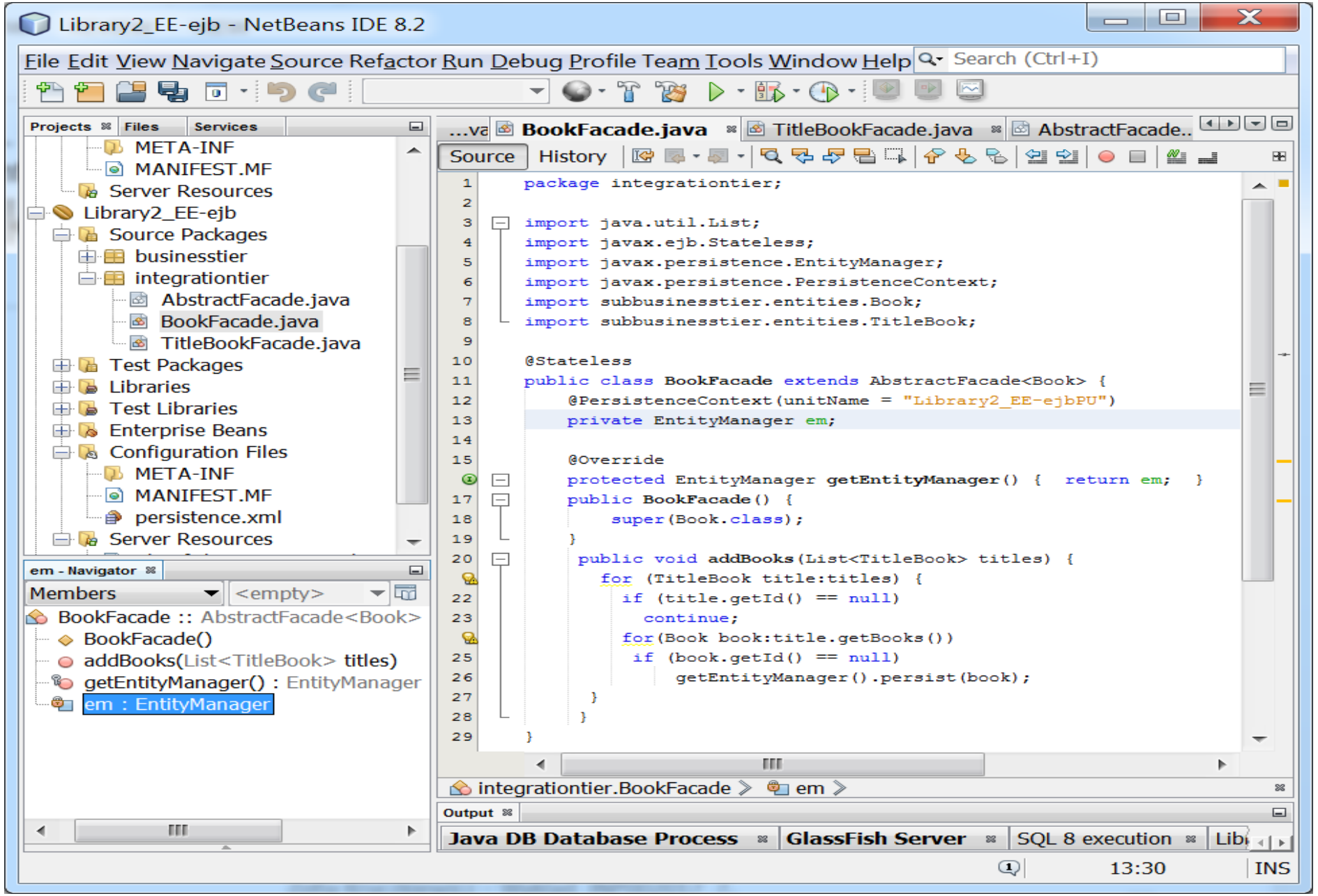
14. Klasa abstrakcyjna generyczna AbstractFacade z definicją uniwersalnych metod obsługujących transakcje JPA - parametr T może być zastąpiony każdą z klas typu Entity

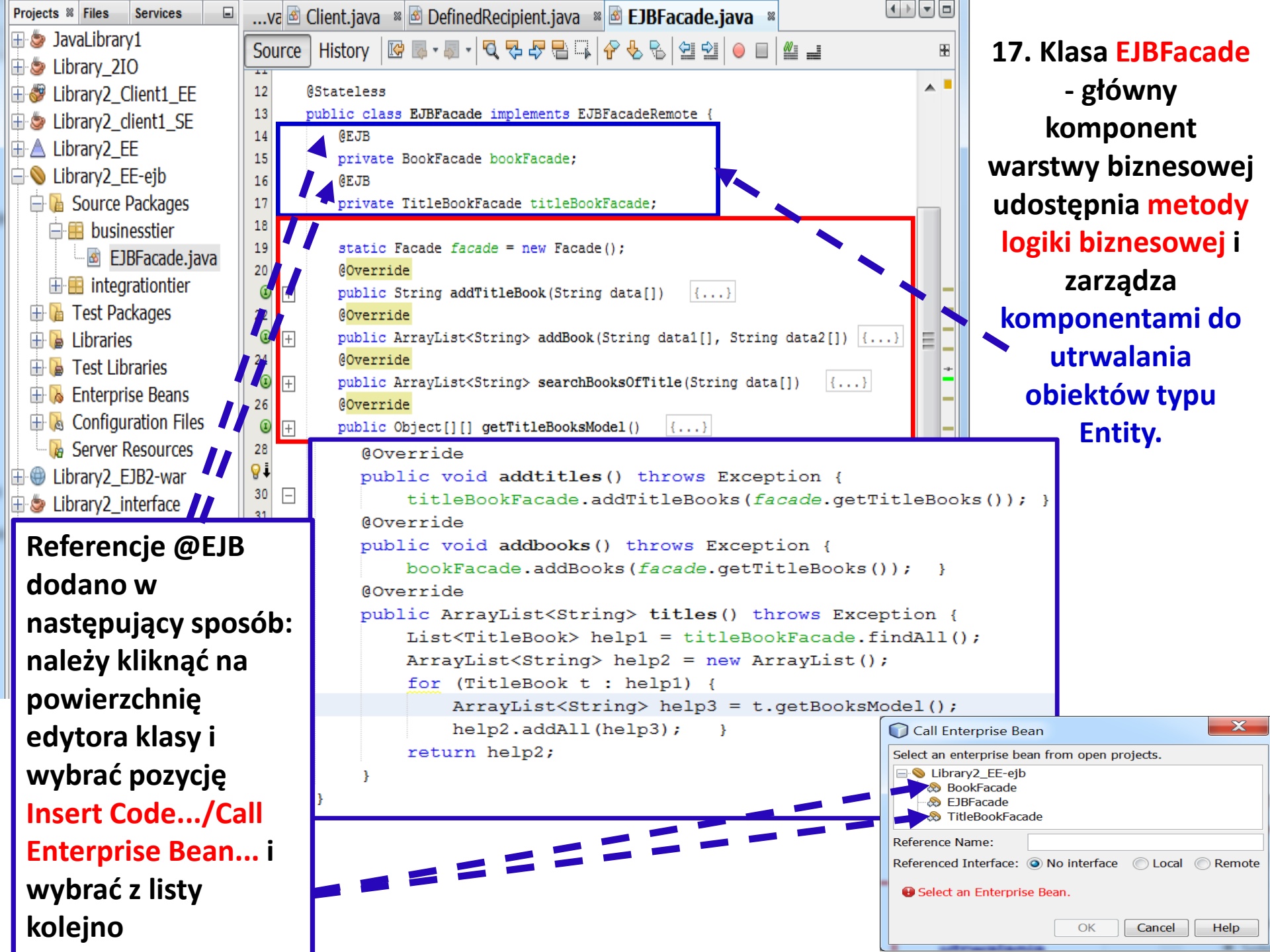


15. Klasa TitleBookFacade implementująca klasę AbstractFacade - kontroler typu Session Bean do utrwalania obiektów typu TitleBook i TitleBookRead. Dodano metodę addTitleBooks.



16. Klasa BookFacade implementująca klasę AbstractFacade - kontroler typu Session Bean do utrwalania obiektów typu Book. Dodano metodę addBooks

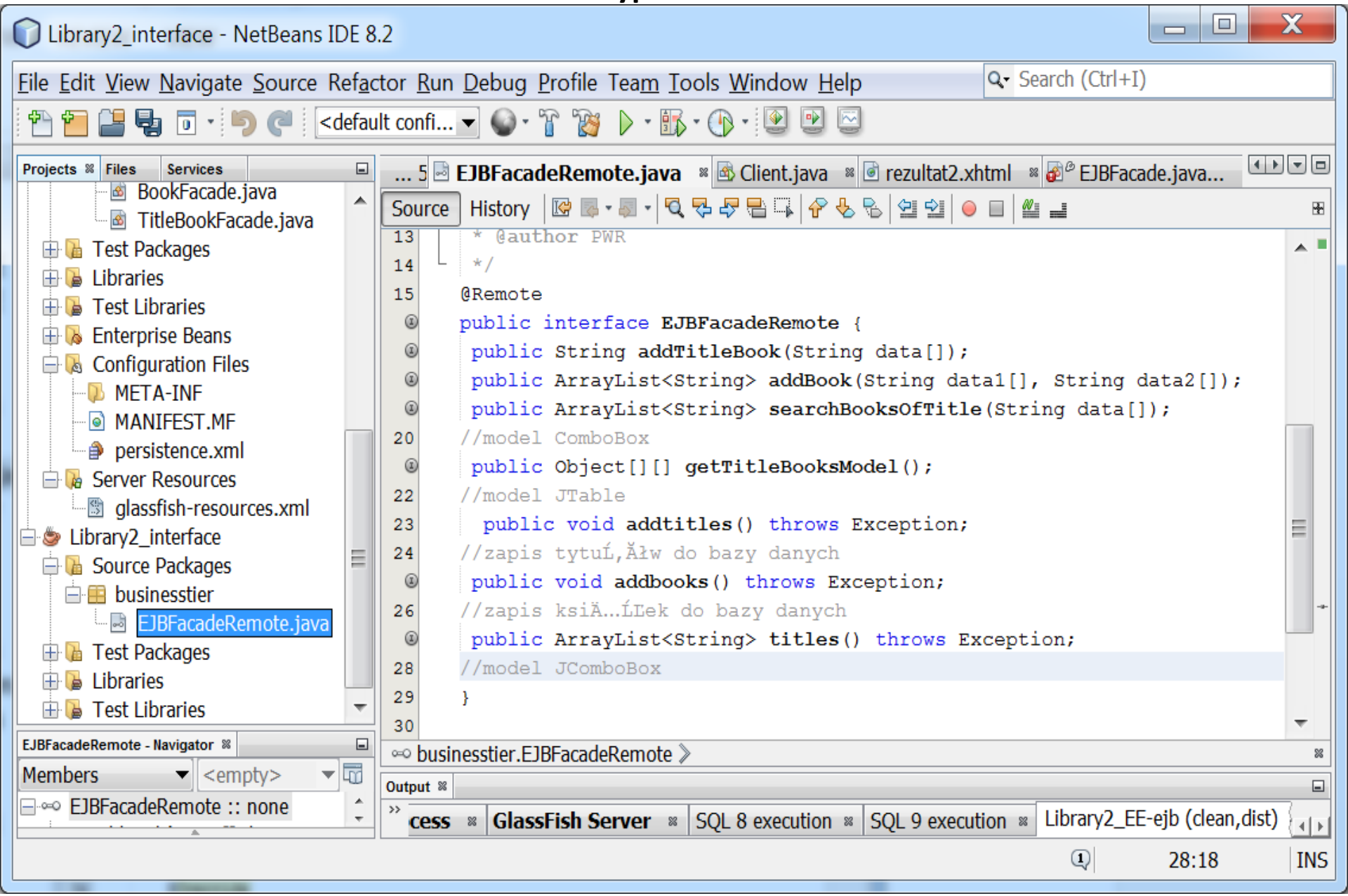




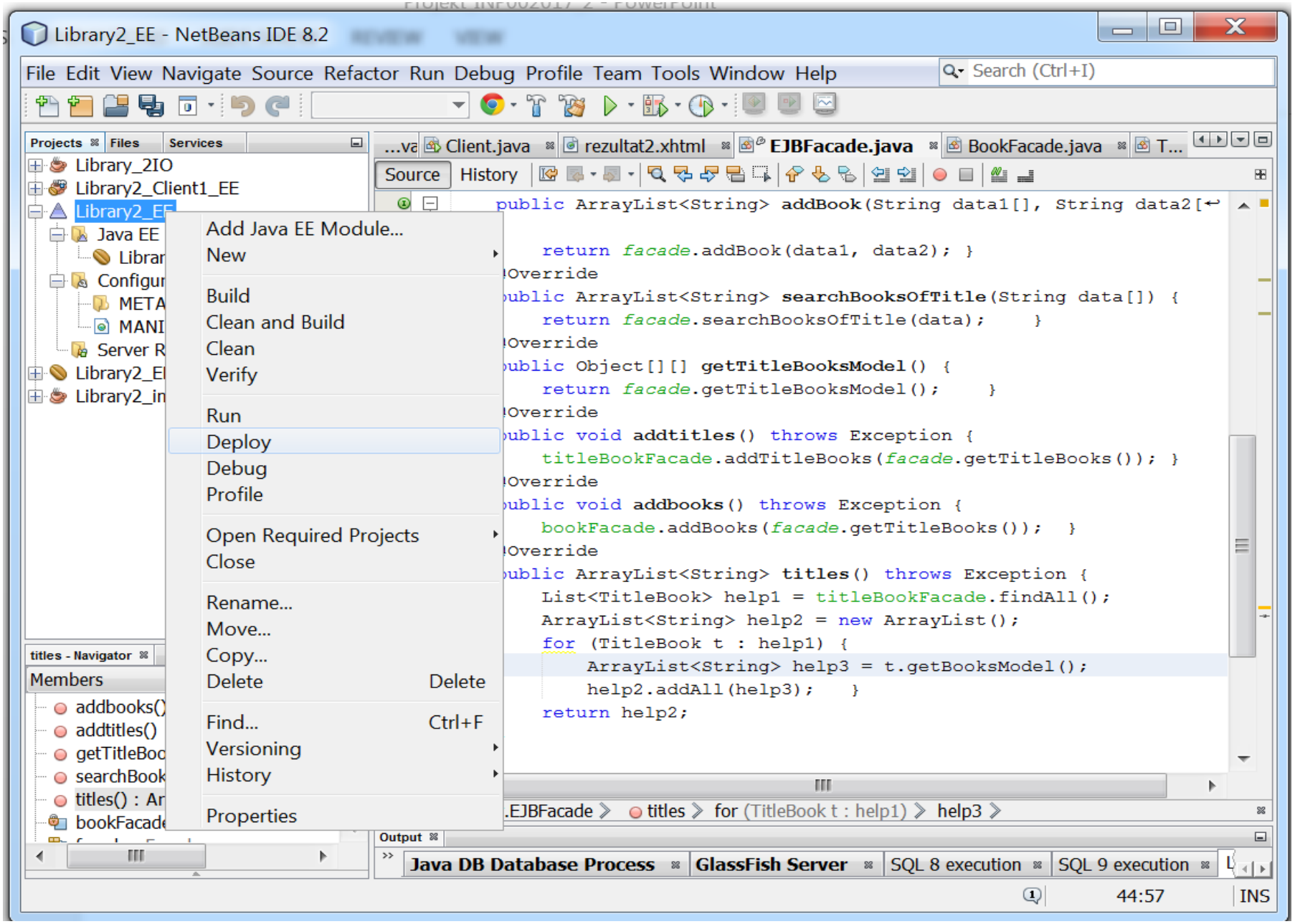
17. Klasa EJBFacade - główny komponent warstwy biznesowej udostępnia metody logiki biznesowej i zarządza komponentami do utrwalania obiektów typu Entity.

Referencje @EJB dodano w następujący sposób: należy kliknąć na powierzchnię edytora klasy i wybrać pozycję Insert Code.../Call Enterprise Bean... i wybrać z listy kolejno

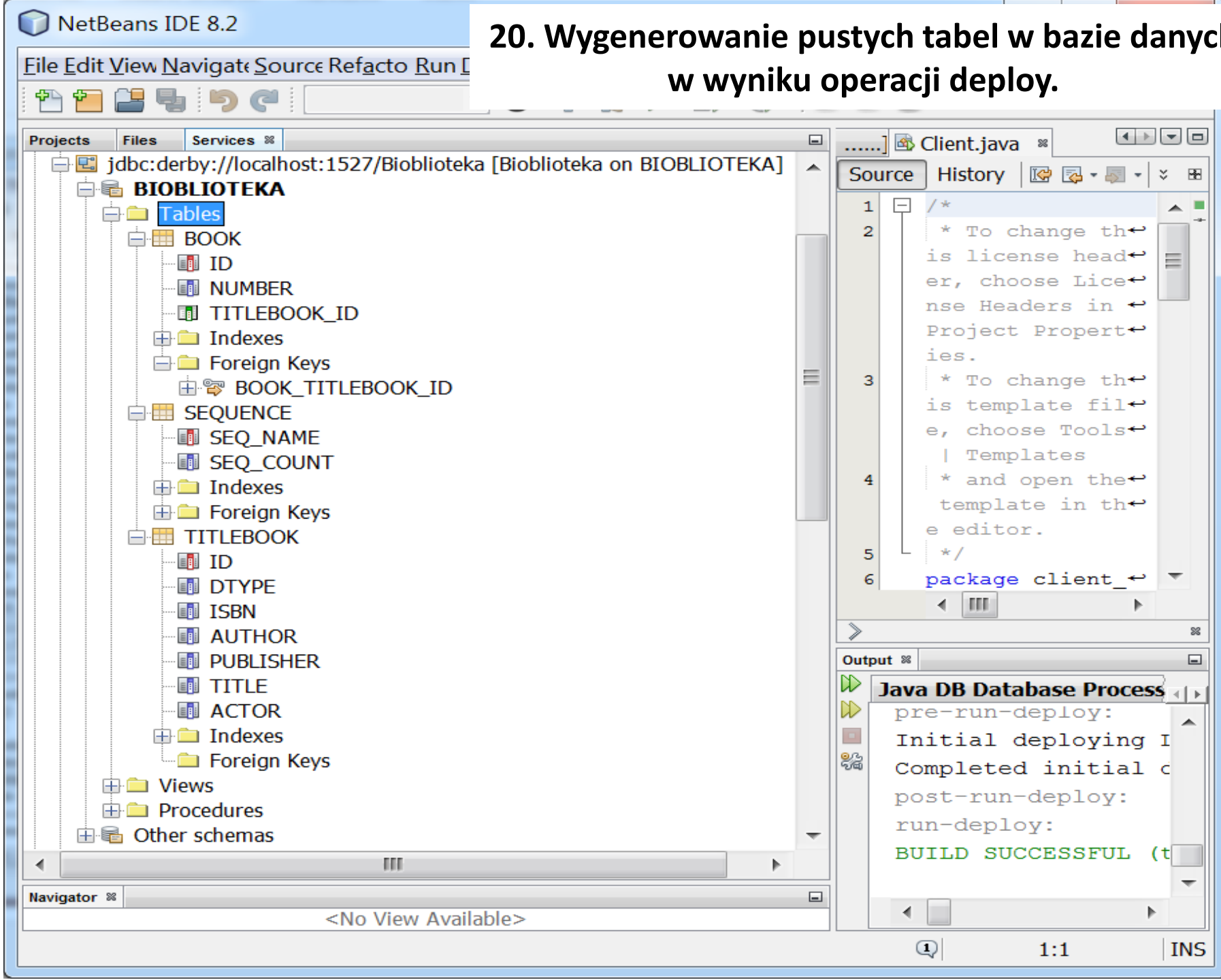
18. Uzupełnienie deklaracji metod o zdalnym dostępie do wybranych metod komponentów do utrwalania obiektów typu Entity w interfejsie EJBFacadeRemote komponentu Session Bean typu EJBFacade

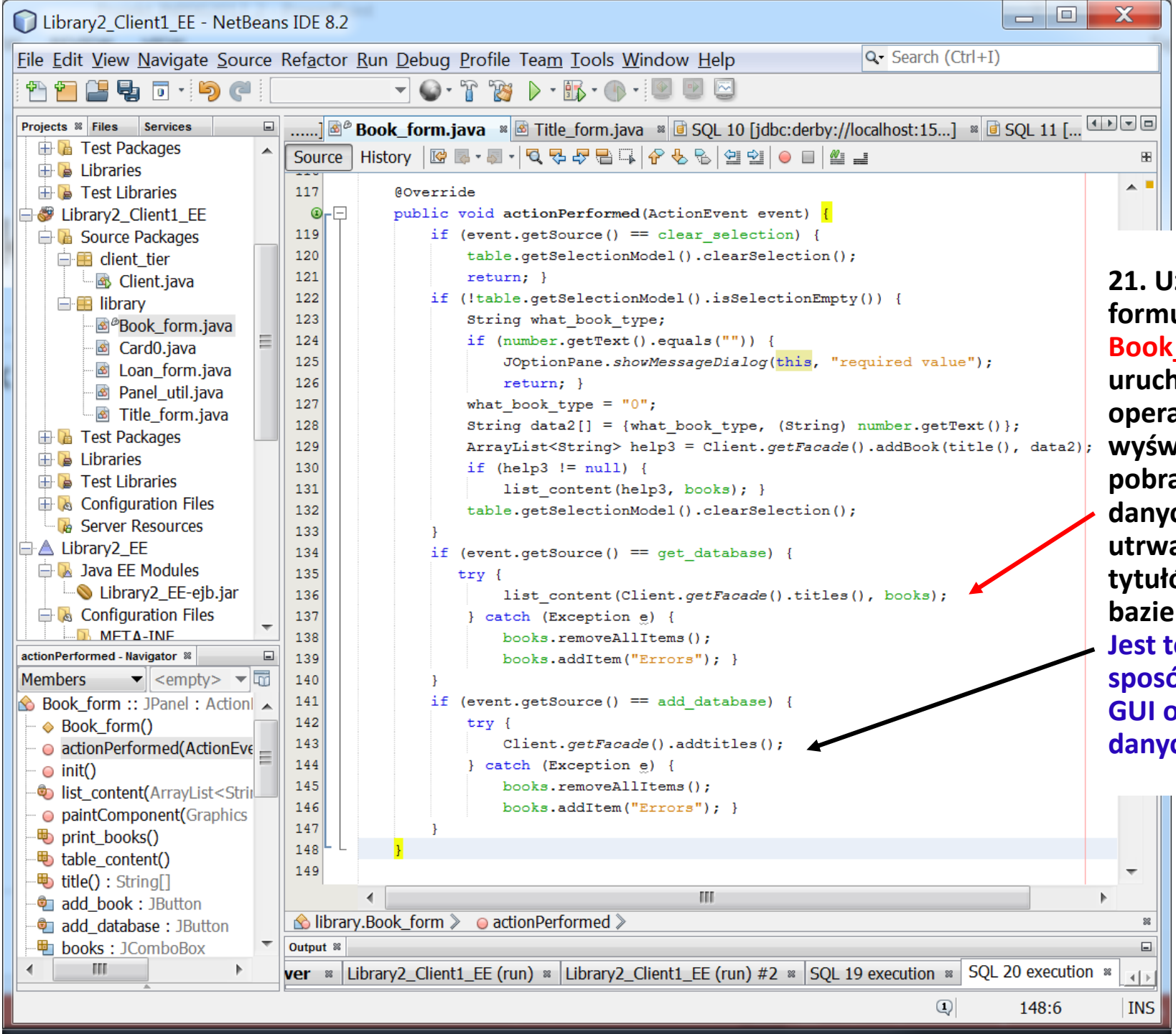


19. Uruchomienie części serwerowej aplikacji typu EE – deploy (po wcześniejszym przeprowadzonym procesie Build na projektach składowych)



20. Wygenerowanie pustych tabel w bazie danych w wyniku operacji deploy.

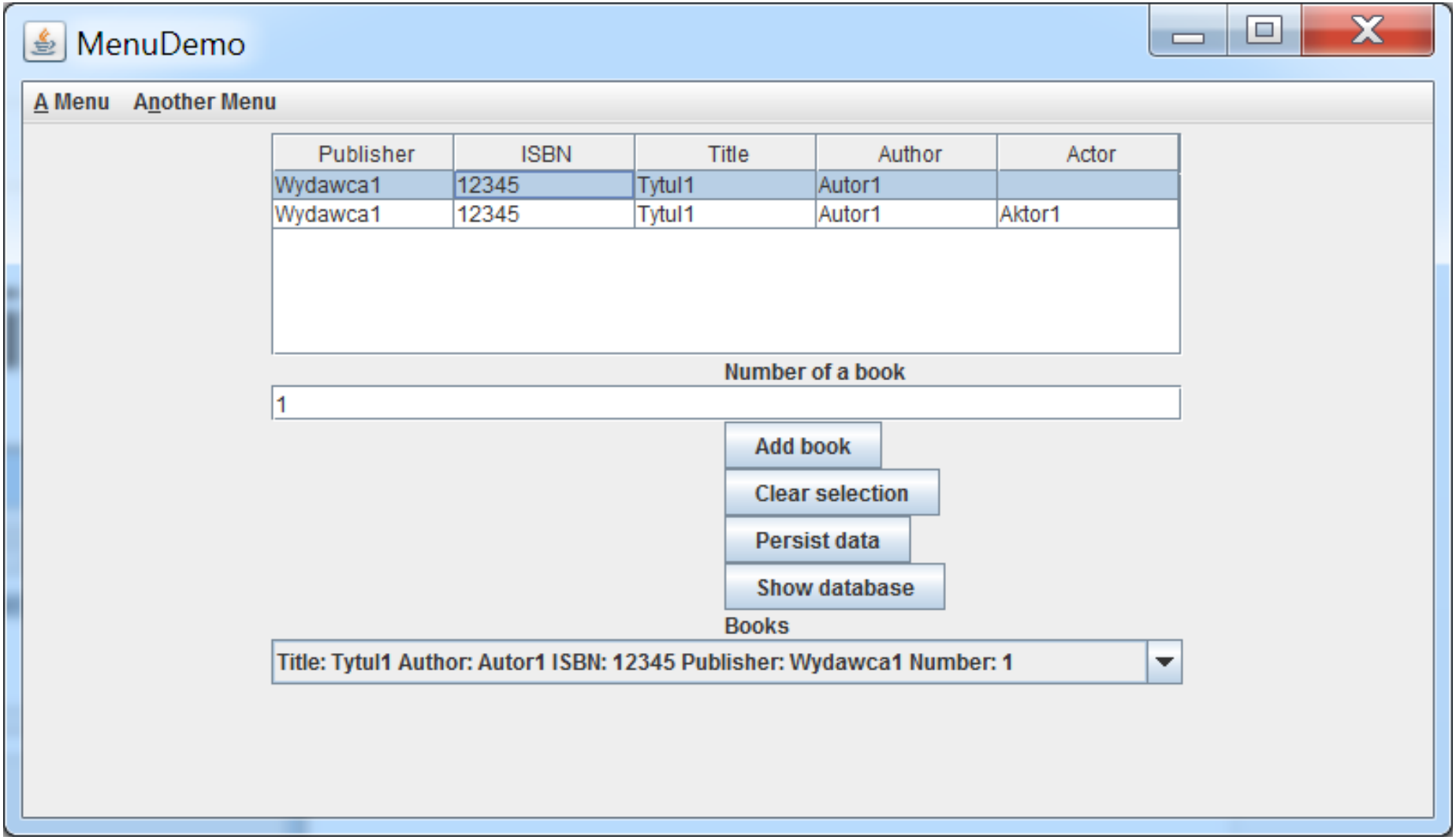




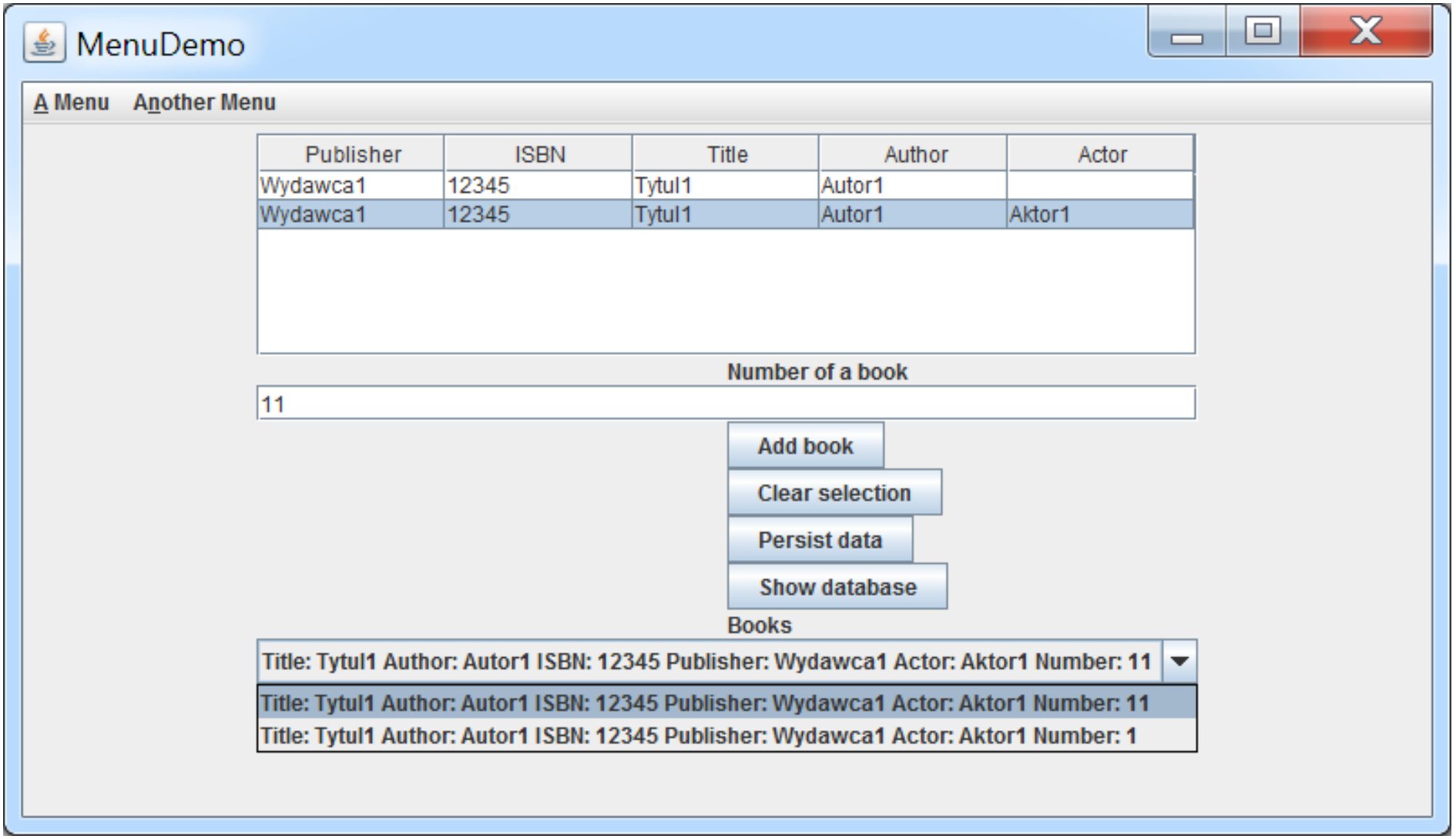
21. Uzupełnienie formularza Book_form o uruchamianie operacji wyświetlanie tych pobranych z bazy danych i utrwalania danych tytułów i książek w bazie danych.

Jest to tymczasowy sposób prezentacji z GUI opracji na bazie danych

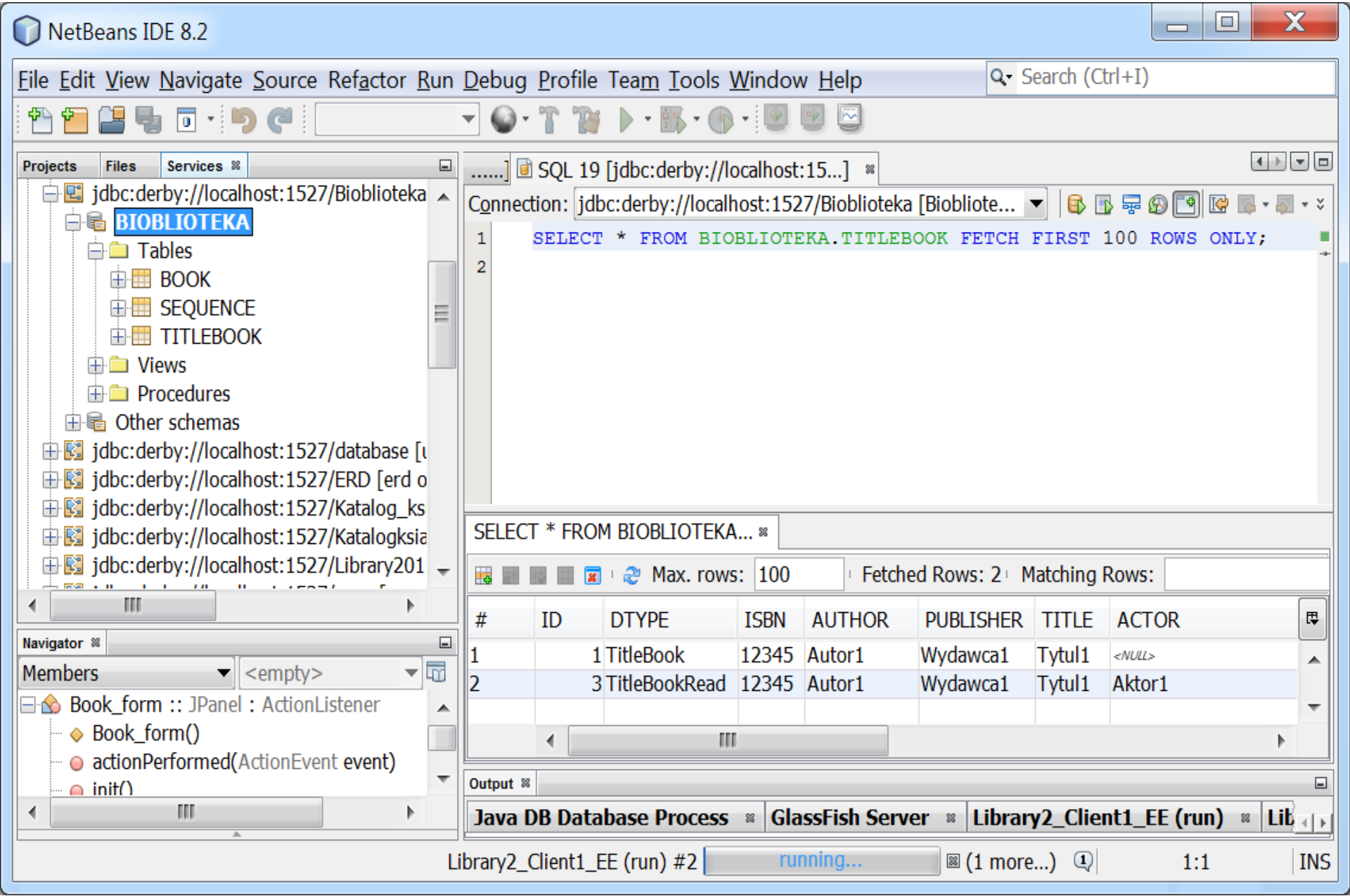
22. Jedna z aplikacji klienckich (po uruchomieniu za pomocą operacji Run) po wprowadzeniu danych tytułów i książek – dane 1-go tytułu i jego książek przechowywane w pamięci programu pokazane w formularzu Book_form.



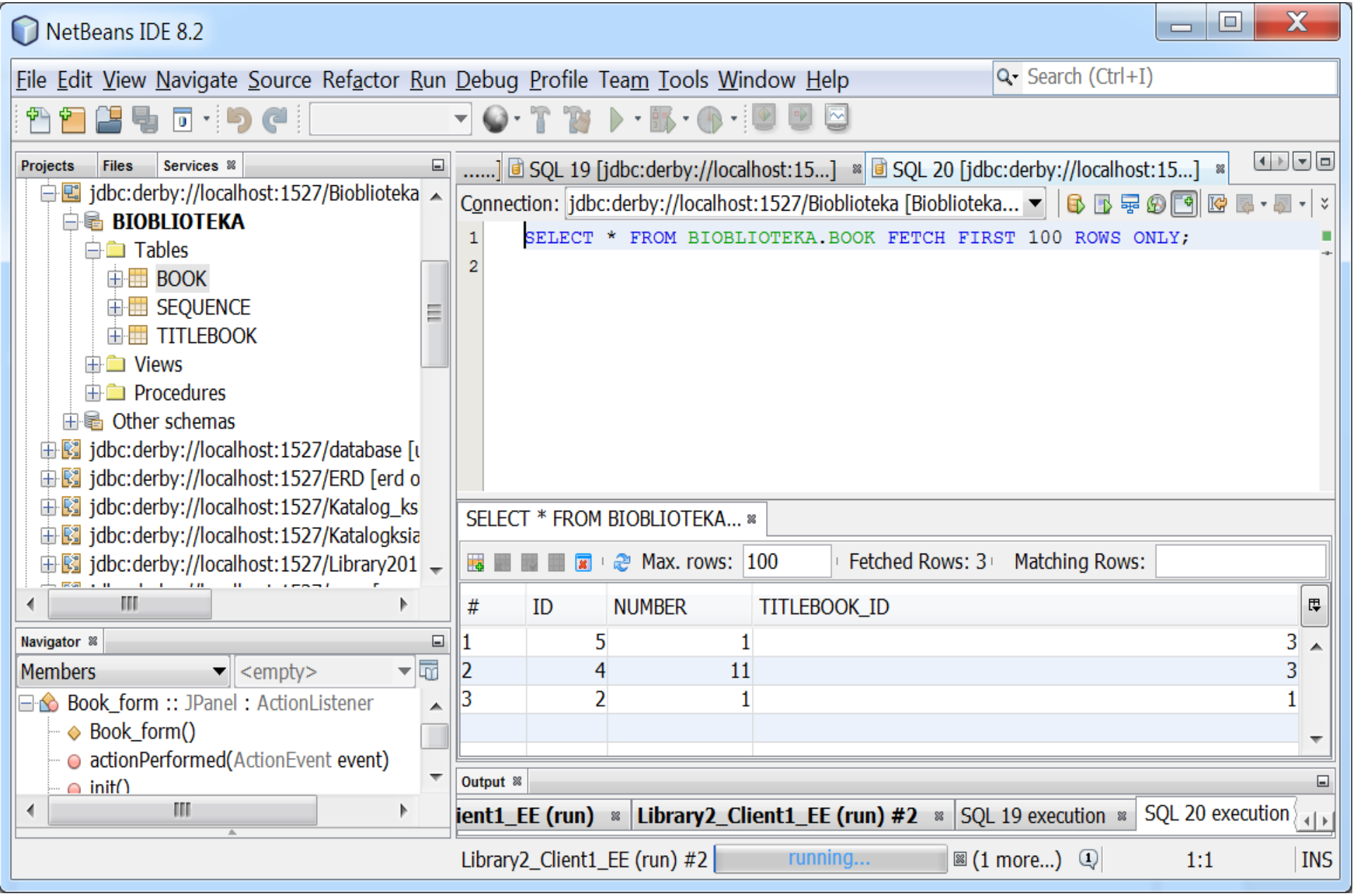
23. Druga z aplikacji klienckich (po uruchomieniu za pomocą operacji Run) po wprowadzeniu danych tytułów i książek – dane 2-go tytułu i jego książek przechowywane w pamięci programu pokazane w formularzu Book_form



24. Dane tytułów zapisane w bazie danych po naciśnięciu klawisza Persist data na formularzu Book_form, zaprezentowane po kliknięciu na nazwę tabeli TitleBook i wybór pozycji View Data...



25. Dane książek zapisane w bazie danych podczas zapisania tytułów (@OneToMany(mappedBy = "titleBook", cascade=ALL)) zaprezentowane po kliknięciu na nazwę tabeli Book i wybór pozycji View Data...



26. Dane tytułów i książek pobrane z bazy danych i wyświetlone w liście rozwijanej (przycisk Show database).