



Projekt warstwy integracji

Autor: Zofia Kruczkiewicz



Design patterns used to build the Integration nad ResourcesTiers

D.Alur, J.Crupi, D. Malks, Core J2EE. Desin Patterns

Outline of creating the Library Catalogue Java Application

1. Create a database in the Derby database system
2. Creating a Persistence Unit
3. Create Annotations in the object model of the Business Tier
4. Create the following classes of the Integration Tier: TTitle_bookController and the TBookController
5. You may add annotation to your own new classes and create the proper controllers – for higher assessment (5.0 or 5.5)



1. Create a database in the Derby database system

1. Check a directory where you should create an empty database of the Derby database system by using the NetBeans - the default directory is a directory of Windows users \. Netbeans-derby – (slide 1)
2. Select the Services tab and then expand the Databases node. Then right-click on Java DB item and select Create Database – (slide 2)
3. In the form for creating a database (slide 3), enter the data as follows: Database Name, User, Password and the Database Location (default or selected by the user). In the selected directory will be created a directory named database with an empty database (slide 4)
4. Check a directory where you have created from NetBeans an empty database in the system database Derby - - you can see the new catalogue Library1 (slide 4)
5. To connect to the database: Right-click the database connection node and choose the Connect item (slide 6) - **`jdbc:derby://localhost:1527/Library1:[Library1 on LIBRARY1] (jdbc:derby://localhost:1527/DataBase_Name:[User on DataBaseSchema])`** .
6. Now expand the database connection node to browse the database until disconnect from the database. Then expand the database node to see the empty Library Database (slide 7) p – the Library1 node is lacking.



1)

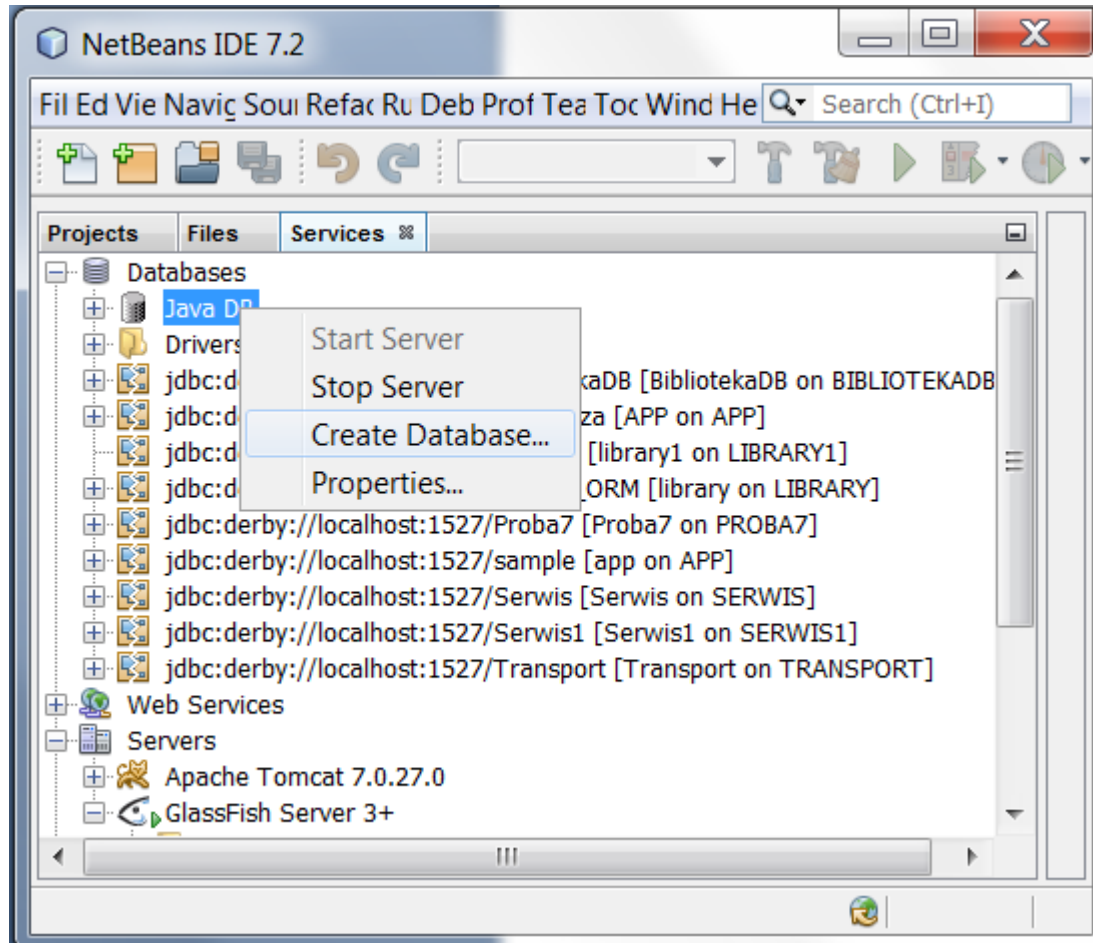
The screenshot shows a Windows Explorer window with the address bar set to: < C: > > Użytkownicy > kruczkiewicz > .netbeans-derby >. The search bar contains the text "Wyszukaj". The main pane displays a list of files and folders in a table format.

Nazwa	Data modyfikacji	Typ	Rozmiar
katalog1	2008-04-14 18:17	Folder plików	
sample	2008-04-14 13:18	Folder plików	
sun-appserv-samples	2008-04-14 00:44	Folder plików	
travel	2008-04-08 19:43	Folder plików	
vir	2008-04-13 03:53	Folder plików	
Zoo	2008-04-11 17:01	Folder plików	
derby	2008-04-16 11:34	Dokument tekstowy	1 KB
derby.properties	2008-03-25 14:13	Plik PROPERTIES	1 KB

At the bottom of the window, a status bar indicates "Elementów: 8".



2)





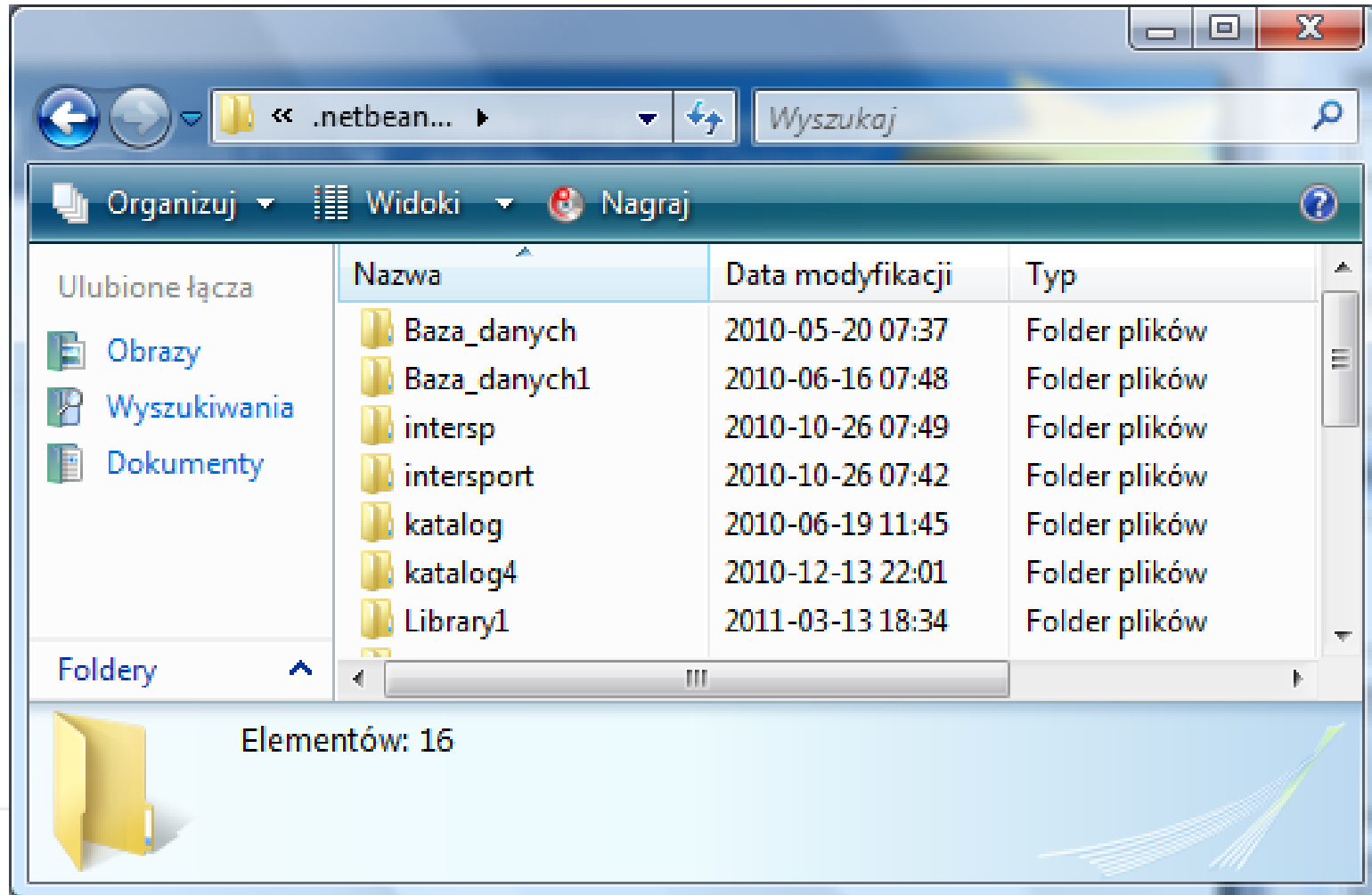
3)

The screenshot shows a dialog box titled "Create Java DB Database" with a close button (X) in the top right corner. The dialog contains the following fields and buttons:

- Database Name:** Library1
- User Name:** Library1
- Password:** Library1
- Database Location:** C:\Users\kruczkiewicz\.netbeans-derby
- Buttons:** Properties..., OK, and Cancel

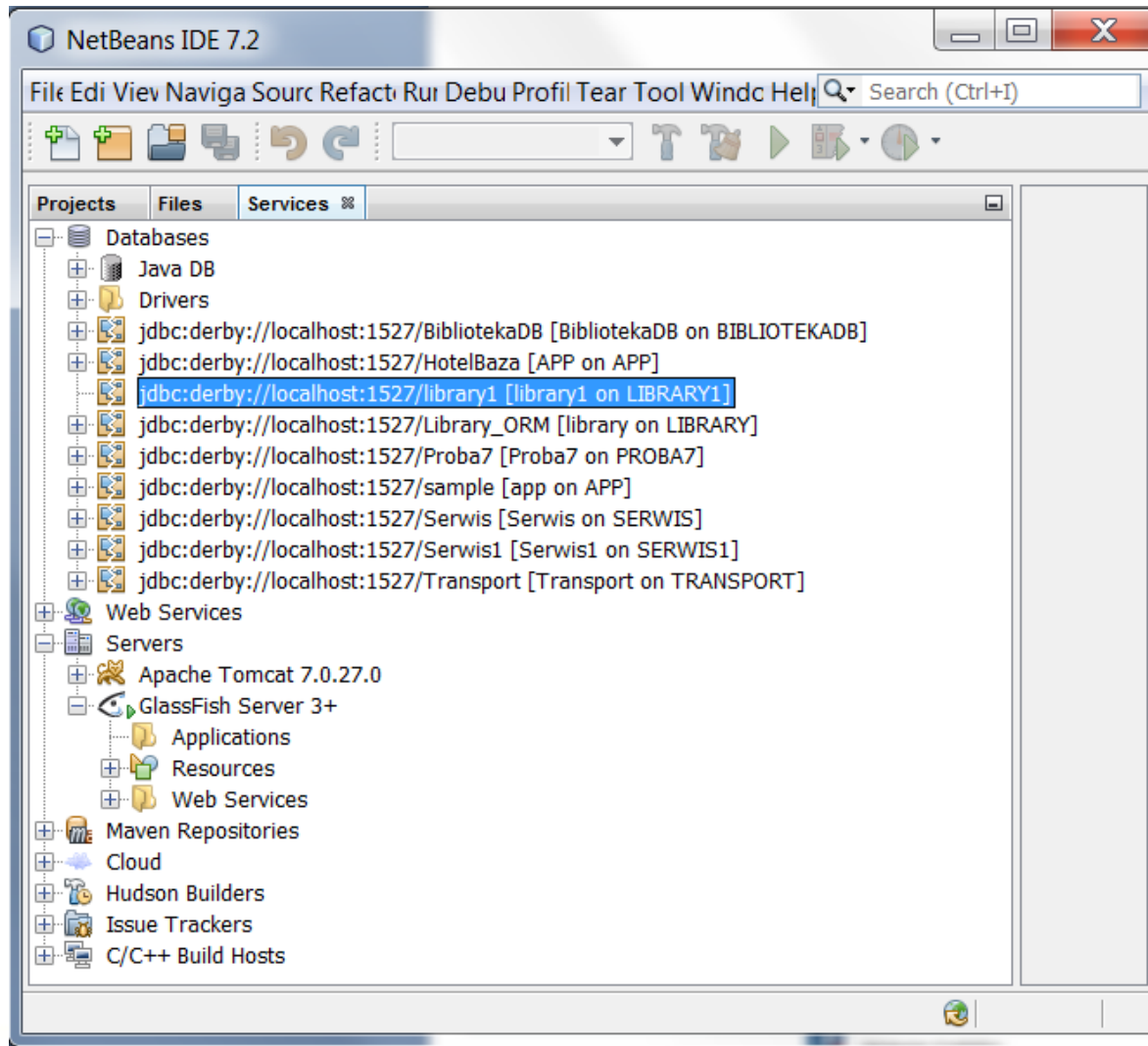


4)



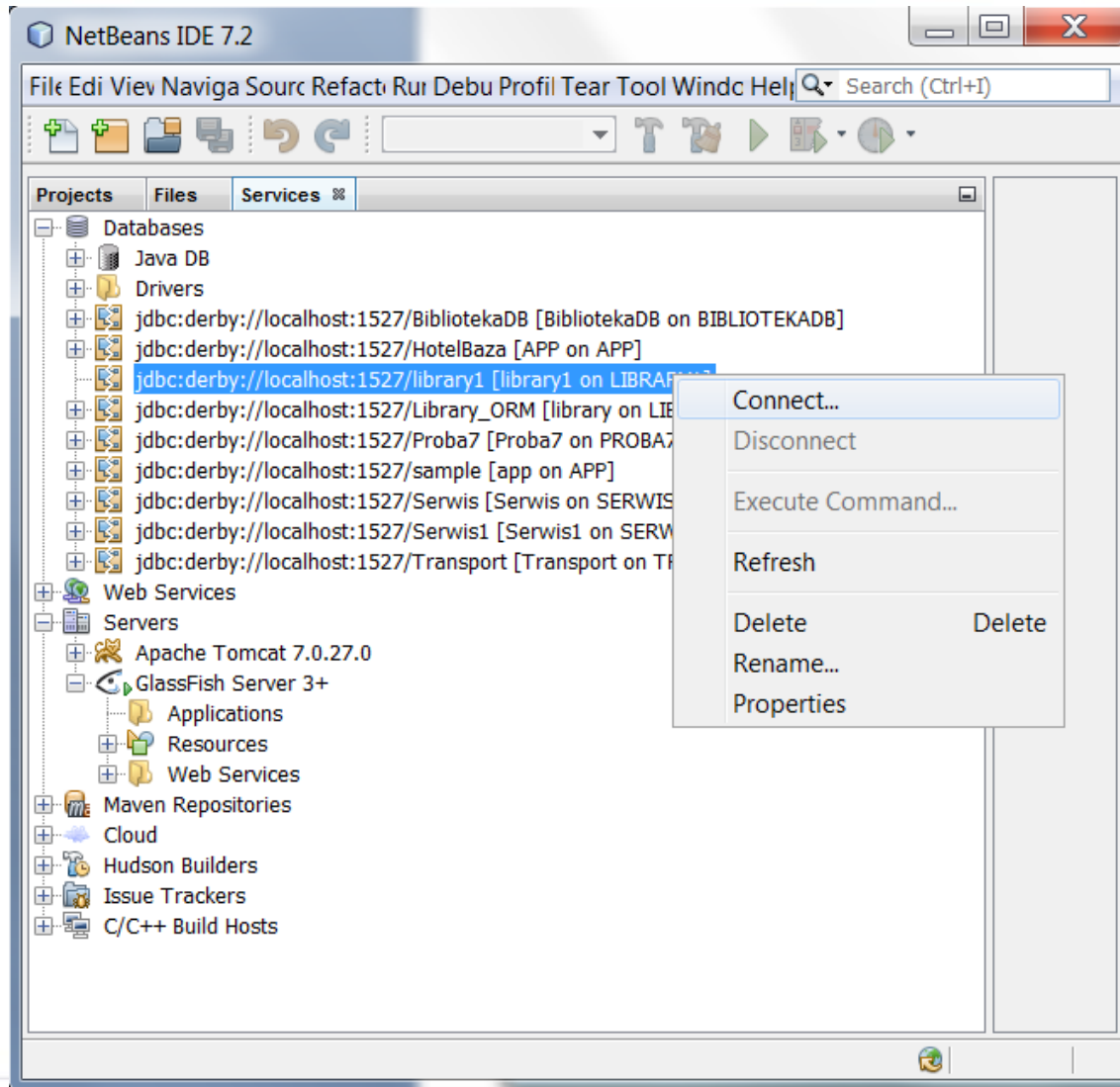


5)



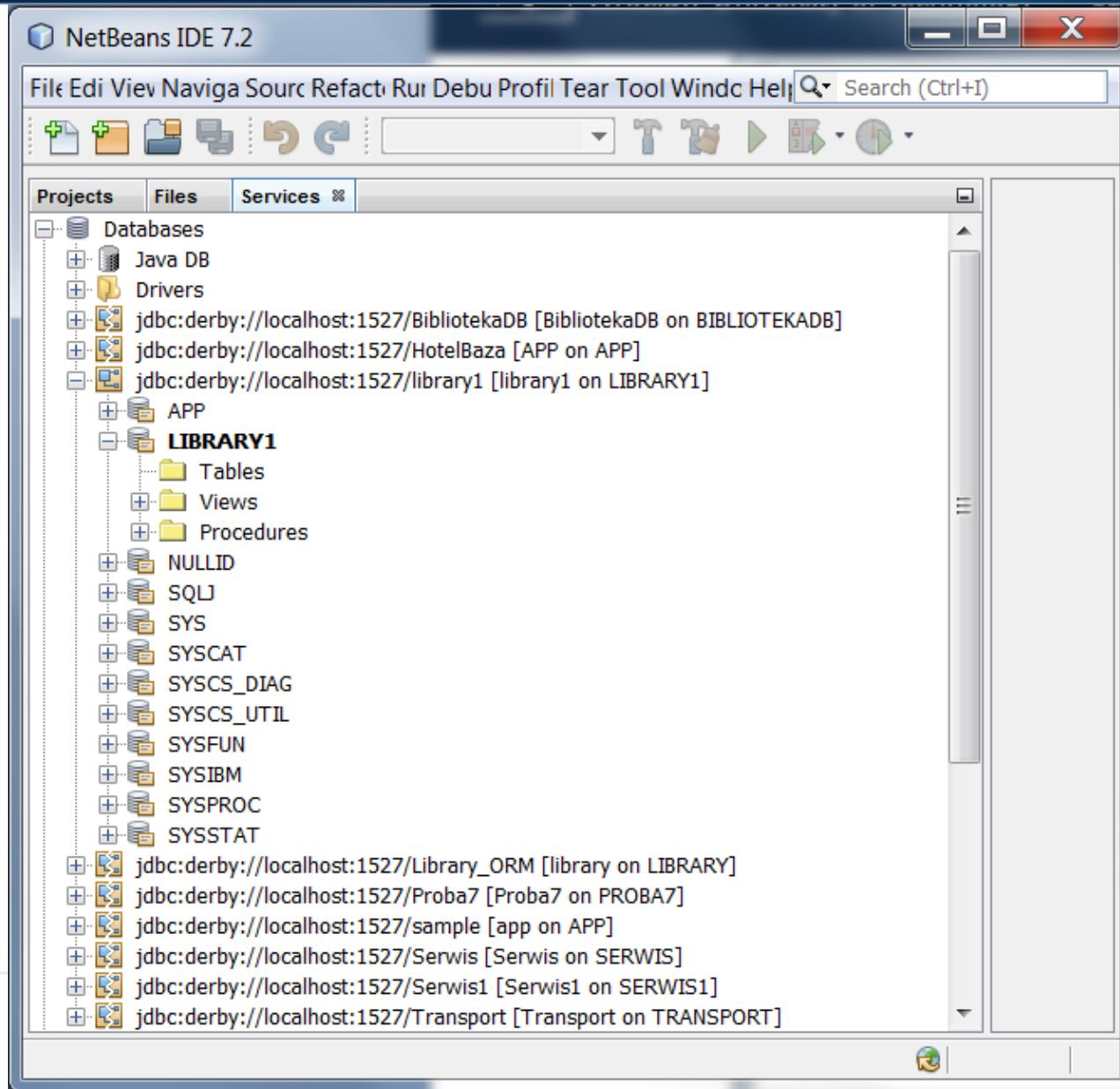


6)





7)





2. Creating a Persistence Unit

To create a persistence unit:

1. In the Projects window, right-click a Library1 project node and choose New > Other.
2. Select Persistence Unit in the Persistence category and click Next.
3. Specify a unique Persistence Unit Name. In most cases you can keep the default name suggested by the IDE.
4. Select a the EclipseLink library by choosing New Persistence Library.
5. Select a datasource from the drop-down menu. The data source can be the Library1 database connection. To appear in the list, the data source needs to be registered with the IDE.
6. Specify a table generation strategy as Create for your database.
7. Click Finish. When you click Finish, the file persistence.xml opens in the Source Editor.



1)

The screenshot shows the NetBeans IDE 7.2 interface. The 'Projects' window on the left displays a tree view with several projects, including 'TLibrary1'. A context menu is open over 'TLibrary1', listing various actions such as 'New', 'Build', 'Clean and Build', 'Run', 'Debug', 'Test', and 'Set Configuration'. The 'New' option is selected, opening a sub-menu with the following items:

- JSF Managed Bean...
- Facelets Template Client...
- Facelets Template...
- Java Class...
- Persistence Unit...
- JSF Faces Configuration...
- JSF Page...
- Java Package...
- JDialog Form...
- Entity Classes from Database...
- Java Interface...
- JPanel Form...
- JFrame Form...
- Entity Class...
- Web Service Client...
- Other...

At the bottom of the IDE, a small text string reads: 'Project co-financed from the EU, European Social Fund'.



2)

New File

Steps

1. Choose File Type
2. ...

Choose File Type

Project: Library1

Categories:

- Web
- JavaServer Faces
- Java
- Swing GUI Forms
- JavaBeans Objects
- AWT GUI Forms
- Unit Tests
- Persistence**
- Groovy
- Hibernate
- Web Services

File Types:

- Entity Class
- Entity Classes from Database
- JPA Controller Classes from Entity Classes
- Persistence Unit
- Database Schema

Description:

Creates an empty Java Persistence API entity class.

< Back Next > Finish Cancel Help

File Types:

- Entity Class
- Entity Classes from Database
- JPA Controller Classes from Entity Classes
- Persistence Unit**
- Database Schema

Description:

Creates an JSR220/JSR317 persistence unit. If persistence.xml doesn't exist, it is created.

< Back Next > Finish Cancel Help



New Persistence Unit

Steps

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

Provider and Database

Persistence Unit Name:

Specify the persistence provider and database for entity classes.

Persistence Library:

Database Connection:

Table Generation Strategy:

3)

4)

New Persistence Unit

Steps

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

Provider and Database

Persistence Unit Name:

Specify the persistence provider and database for entity classes.

Persistence Library:

Database Connection:

Table Generation Strategy:

5)



6)

New Persistence Unit

Steps

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

Provider and Database

Persistence Unit Name:

Specify the persistence provider and database for entity classes.

Persistence Library:

Database Connection:

Table Generation Strategy: Create Drop and Create None

< Back Next > **Finish** Cancel Help



Facade

Factory

Flyweight

Strategy



3. Create Annotations in the object model

Mapping Entity Classes (Help of the NetBeans)

- An entity class is used to represent a table in a database, and the fields in an entity class correspond to columns in that table. In an entity class, you can use annotations to specify how fields in an entity class are mapped to the corresponding database columns and tables.
- If the name of the mapped database column is the same as the field or property name because they are mapped by default.

The following annotations are commonly used when mapping entity classes:

- @Entity defines the Entity class
- @Id Specifies the primary key property or field of an entity.
- @GeneratedValue Allows you to specify the strategy that automatically generates the values of primary keys. Used with @Id.
- @Column Specifies a mapped column for a persistent property or field.
- @ManyToMany Defines a many-valued association with many-to-many multiplicity.
- @ManyToOne Defines a single-valued association to another entity class that has many-to-one multiplicity.
- @OneToMany Defines a many-valued association with one-to-many multiplicity. For more on using annotations and annotation elements to map entities in an enterprise application, see the Java EE 5 Tutorial

Insert the annotations shown in the next slides to the TTitle_book, TTitle_book_on_tape, TBook, TBook_period

Click the Add Class and select all classes as the Entity classes (slide 5)



TLibrary1 - NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

1)

Projects Files Services

- Library_client_2017
 - MenuSelectionManagerDemo
 - Okna1
 - Proba
 - ServletStateless
 - Szyfrowanie
 - TableDialogEditDemo
 - TableSelectionDemo
 - TLibrary1
 - Source Packages
 - META-INF
 - persistence.xml
 - integration_tier
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTitle_book.java**
 - TTitle_book_on_tape.java
 - Generated Sources (ap-source-output)
 - Libraries
 - EclipseLink (JPA 2.0) - eclipselink-2.3.2.j
 - EclipseLink (JPA 2.0) - javax.persistence
 - EclipseLink (JPA 2.0) - org.eclipse.persis
 - JDK 1.7 (Default)

```
TTitle_book.java
Source History
16
17 @Entity
18 public class TTitle_book implements Serializable {
19
20     private static final long serialVersionUID = 1L;
21     private String publisher;
22     private String ISBN;
23     private String title;
24     private String author;
25
26     @Id
27     @GeneratedValue(strategy = GenerationType.AUTO)
28     private Long id;
29     public Long getId() { return id; }
30     public void setId(Long val) { id = val; }
31
32     @OneToMany(mappedBy = "mTitle_book")
33     private Collection<TBook> books;
34     public Collection<TBook> getBooks() { return books; }
35     public void setBooks(Collection<TBook> val) { books = val; }
36
37     public TTitle_book() { id = null; }
38
39     @Transient
40     private ArrayList<TBook> mBooks = new java.util.ArrayList<TBook>();
41     public ArrayList<TBook> getmBooks() { return mBooks; }
42     public void setmBooks(ArrayList<TBook> books) { mBooks = books; }
```

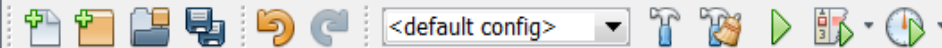


2)

TLibrary1 - NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)



Projects Files Services

- GridLayoutDemo
- Library_client_2014
- MenuSelectionManagerDemo
- Okna1
- Proba
- ServletStateless
- Szyfrowanie
- TableDialogEditDemo
- TableSelectionDemo
- TLibrary1
 - Source Packages
 - META-INF
 - persistence.xml
 - integration_tier
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTitle_book.java
 - TTitle_book_on_tape.java
 - Generated Sources (ap-source-output)
 - Libraries
 - EclipseLink (JPA 2.0) - eclipselink-2.3.2.jar
 - EclipseLink (JPA 2.0) - javax.persistence-2.0.3.jar
 - EclipseLink (JPA 2.0) - org.eclipse.persistence.jpa.jpql_1.0.1.jar
 - JDK 1.7 (Default)

TTitle_book.java TTitle_book_on_tape.java

Source History

```
5 package sub_business_tier.entities;
6
7 import java.io.Serializable;
8 import javax.persistence.Entity;
9
10 /** */
11
12 @Entity
13
14 public class TTitle_book_on_tape extends TTitle_book
15     implements Serializable
16
17
18     private static final long serialVersionUID = 1L;
19     private String actor;
20
21
22 @Override
23 public String getActor() { return actor; }
24
25 @Override
26 public void setActor(String val) {
27     actor = val;
28 }
29
30 @Override
31 public String toString() {
32     String help = super.toString();
33     help += " Actor: " + getActor();
34     return help;
35 }
```

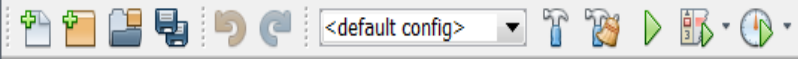


3)

TLibrary1 - NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)



Projects Files Services

- GridLayoutDemo
- Library_client_2014
- MenuSelectionManagerDemo
- Okna1
- Proba
- ServletStateless
- Szyfrowanie
- TableDialogEditDemo
- TableSelectionDemo
- TLibrary1
 - Source Packages
 - META-INF
 - persistence.xml
 - integration_tier
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - TBook.java**
 - TBook_period.java
 - TTitle_book.java
 - TTitle_book_on_tape.java
- Generated Sources (ap-source-output)
- Libraries
 - EclipseLink (JPA 2.0) - eclipselink-2.3.2.jar
 - EclipseLink (JPA 2.0) - javax.persistence-2.0.3.jar
 - EclipseLink (JPA 2.0) - org.eclipse.persistence.jpa.jpql_1.0.1.jar
 - JDK 1.7 (Default)

```
Source History
1 package sub_business_tier.entities;
2
3 import java.io.Serializable;
4 import java.util.Date;
5 import javax.persistence.Entity;
6 import javax.persistence.GeneratedValue;
7 import javax.persistence.GenerationType;
8 import javax.persistence.Id;
9 import javax.persistence.ManyToOne;
10
11 @Entity
12 public class TBook implements Serializable {
13     private static final long serialVersionUID = 1L;
14     private int number;
15
16     @Id
17     @GeneratedValue(strategy = GenerationType.AUTO)
18     private Long id;
19     public void setId(Long id) { this.id = id; }
20     public Long getId() { return id; }
21
22     @ManyToOne
23     private TTitle_book mTitle_book;
24     public TTitle_book getTitle_book() { return mTitle_book; }
25     public void setTitle_book(TTitle_book title_book) { mTitle_book = title_book; }
26
27     public TBook() { id = null; }
```



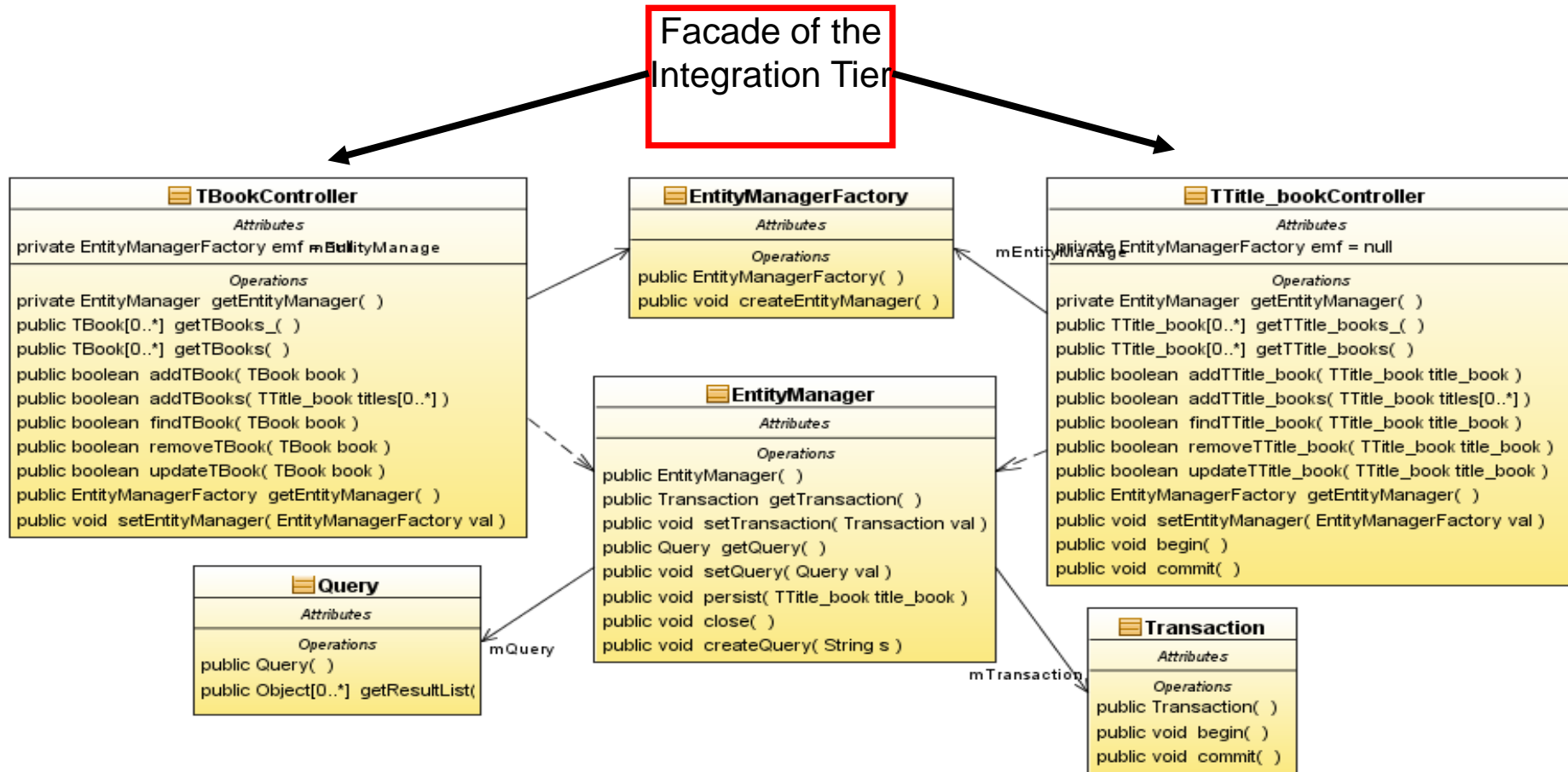

5)

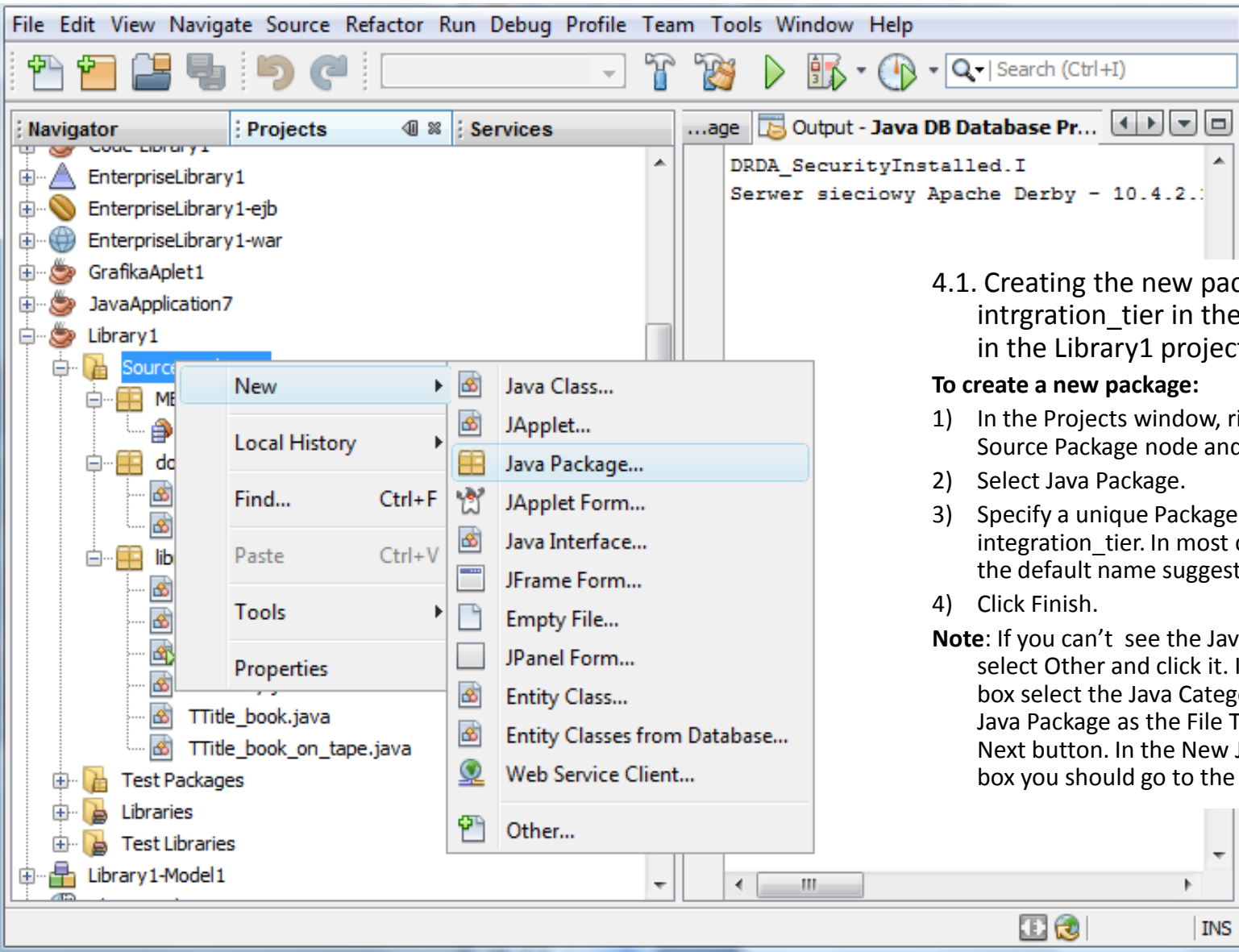
The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays a project named 'Library1' with a package structure including 'sub_business_tier.entities'. A blue box highlights this package and its contents. On the right, the Persistence Units view shows the configuration for 'Library1PU'. The 'Include Entity Classes' field contains the following classes: 'sub_business_tier.entities.TBook', 'sub_business_tier.entities.TBook_period', 'sub_business_tier.entities.TTitle_book', and 'sub_business_tier.entities.TTitle_book_on_tape'. A blue box highlights this list, and a blue arrow points from the 'sub_business_tier.entities' package in the Project Explorer to this list.

The names of packages on this slide are different than on other slides – you must select the entities from your own packages of your program. In real situation, the tool only allows you to select classes with Entity annotation.



4. Create the following classes of the Integration Tier: TTitle_bookController and the TBookController for persisting of objects.





4.1. Creating the new package intrgration_tier in the source packages in the Library1 project

To create a new package:

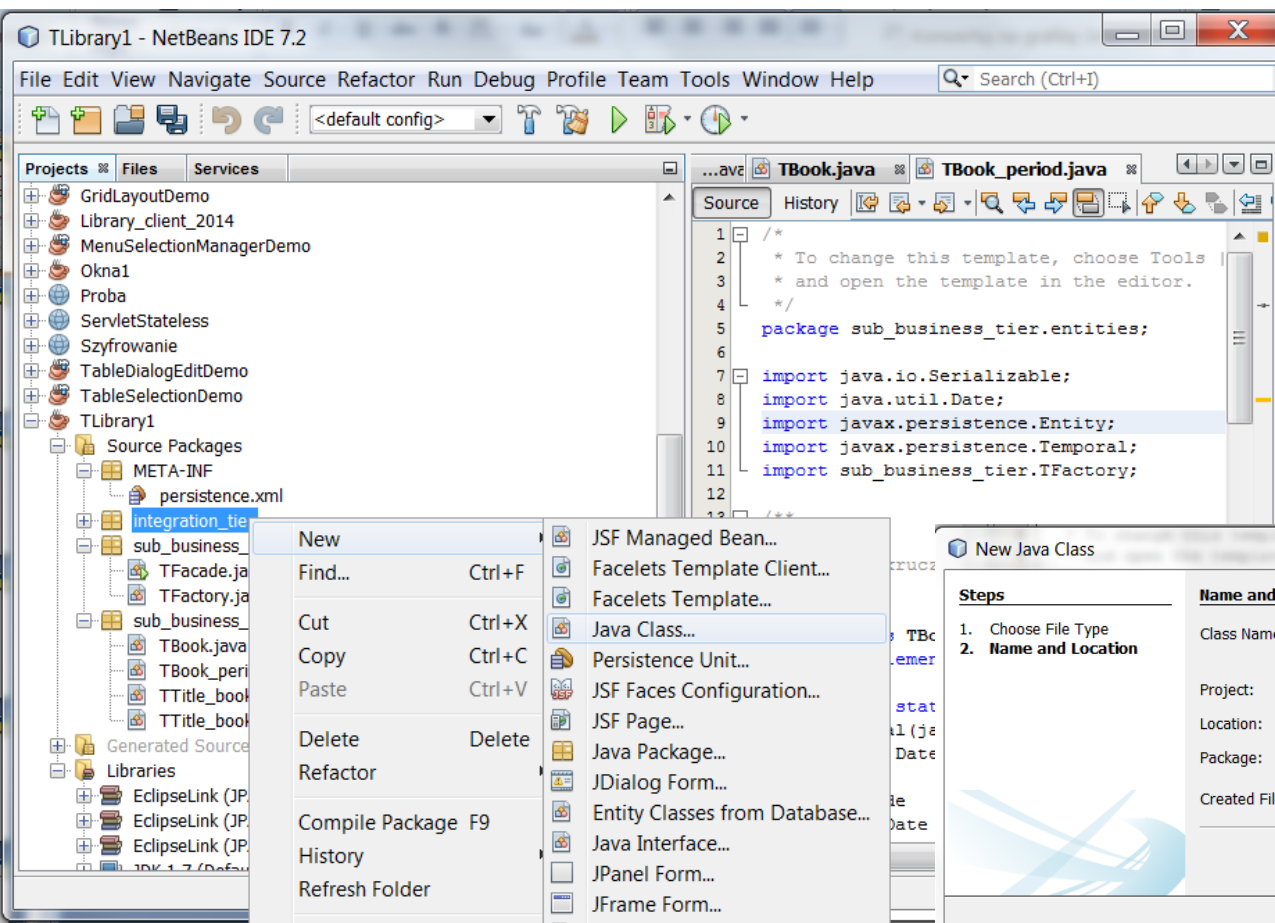
- 1) In the Projects window, right-click a Library1 Source Package node and choose New >.
- 2) Select Java Package.
- 3) Specify a unique Package Name – integration_tier. In most cases you can keep the default name suggested by the IDE.
- 4) Click Finish.

Note: If you can't see the Java Package item, you select Other and click it. In the New File dialog box select the Java Categories item and the Java Package as the File Types item. Click the Next button. In the New Java Package dialog box you should go to the p.3



To create a new Java Class file:

- 1) In the Projects window, right-click a Library1 Source Package node and choose New.
- 2) Select Java Class.
- 3) Specify a unique Class Name – TTile_bookController. In most cases you can keep the default name suggested by the IDE.
- 4) Click Finish.
- 5) Repeat these activities to create the TBookController Java Class (p.4.3).

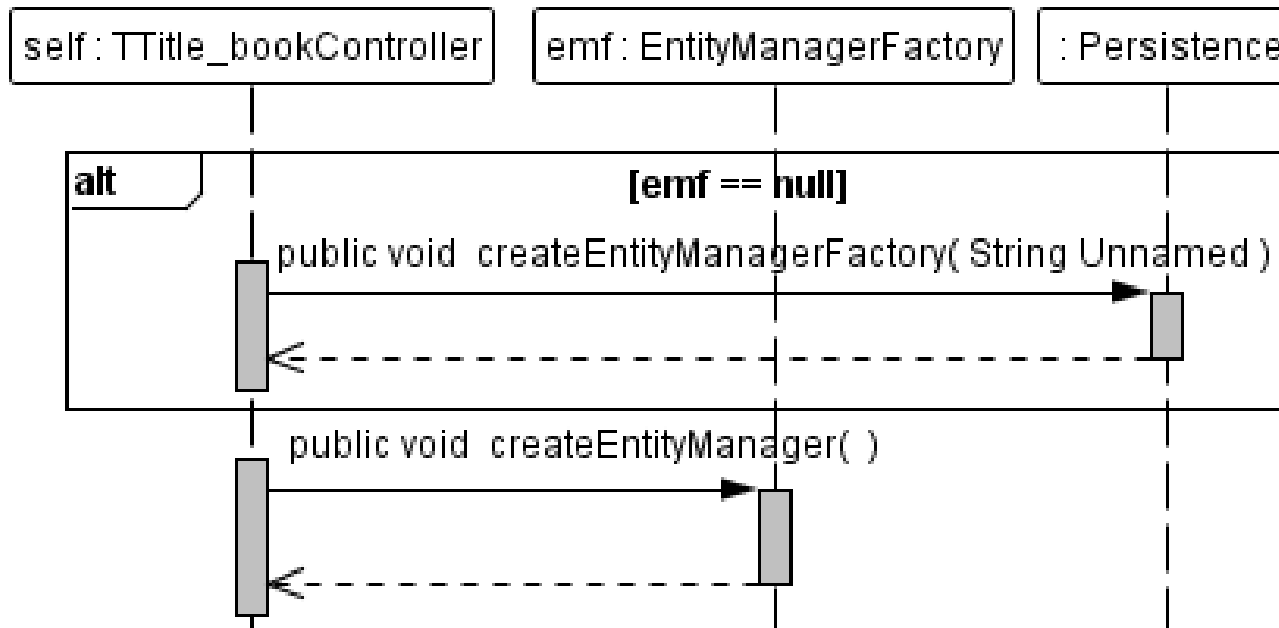


Note: If you can't see the Java Java Class item, you select Other and click it. In the New File dialog box select the Java Categories item and the Java Class as the File Types item. Click the Next button. In the New Java Class dialog box you should go to the p.3



4.2. Creating the the TBookController.java in the domainstore package
for persisting TTitle_book and TTitle_book_on_tape objects.
Applying of design patterns Domain Store i Transfer Object

(1) The TTitle_bookController class:
creating the EntityManager object
private EntityManager getEntityManager()



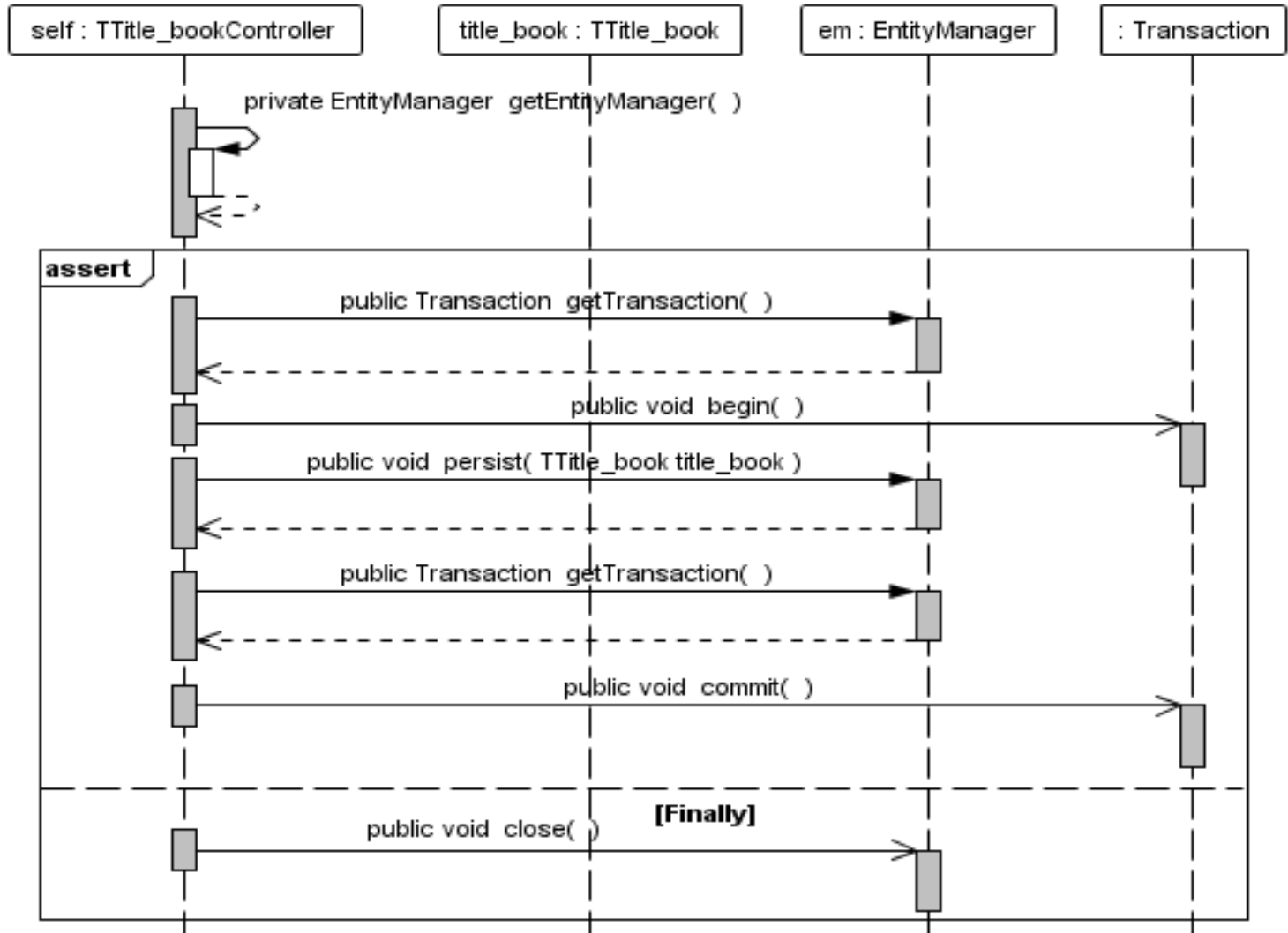


(2) The TTitle_bookController class: creating the EntityManager object
private EntityManager getEntityManager()

```
public class TTitle_bookController {  
  
    private EntityManagerFactory emf = null;  
  
    private EntityManager getEntityManager() {  
        if (emf == null) {  
            emf = Persistence.createEntityManagerFactory("Library1PU");  
        }  
        return emf.createEntityManager();  
    }  
}
```



(1) **The TTitle_bookController class:** Adding a title in the database
`public boolean addTTitle_book(TTitle_book title_book)`



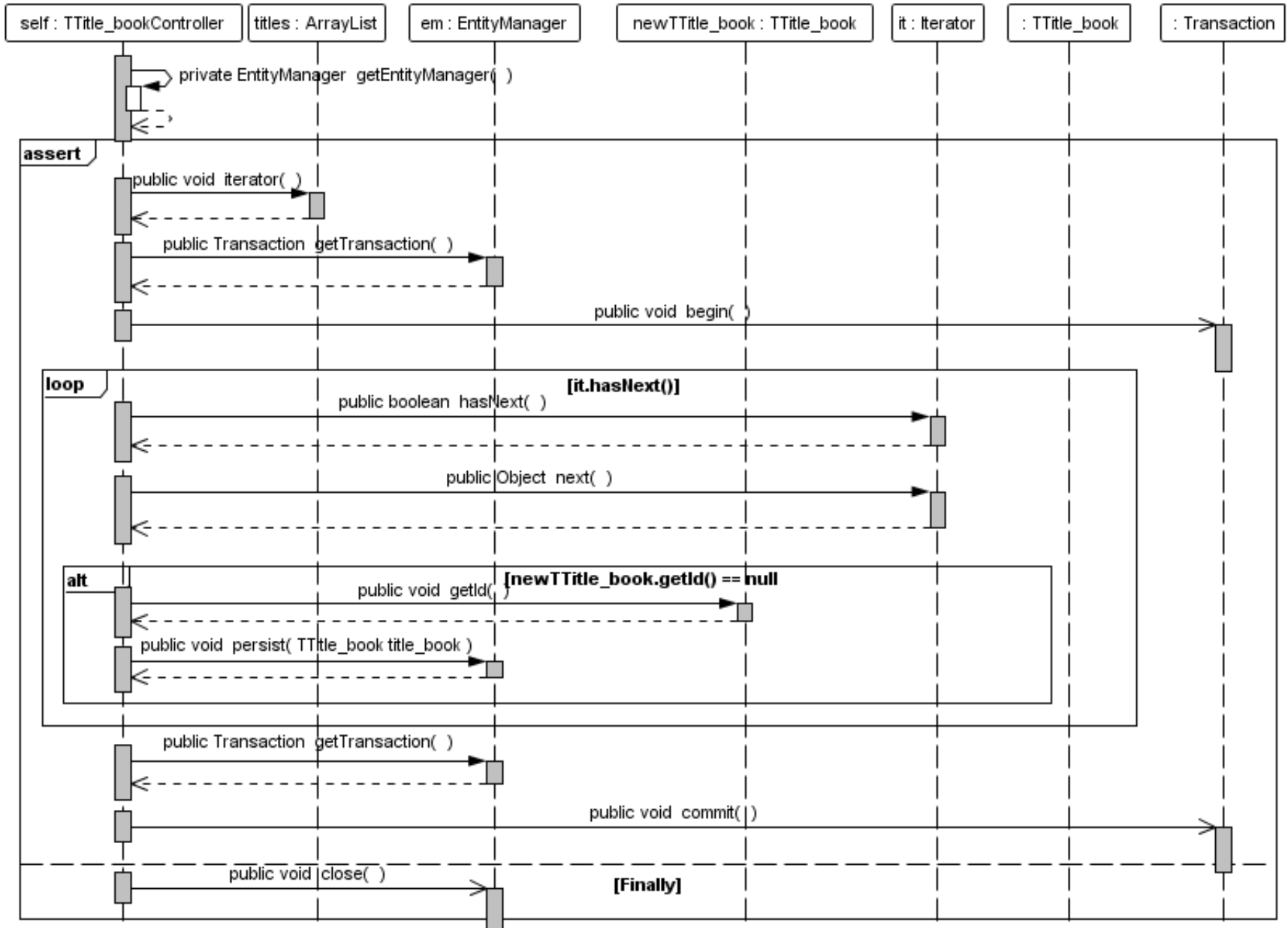


(2) The TTitle_bookController class: Adding a title in the database
public boolean addTTitle_book(TTitle_book title_book)

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

(1) The TTitle_bookController class: Adding titles in the database

public boolean addTTitle_books (ArrayList<TTitle_book> titles)



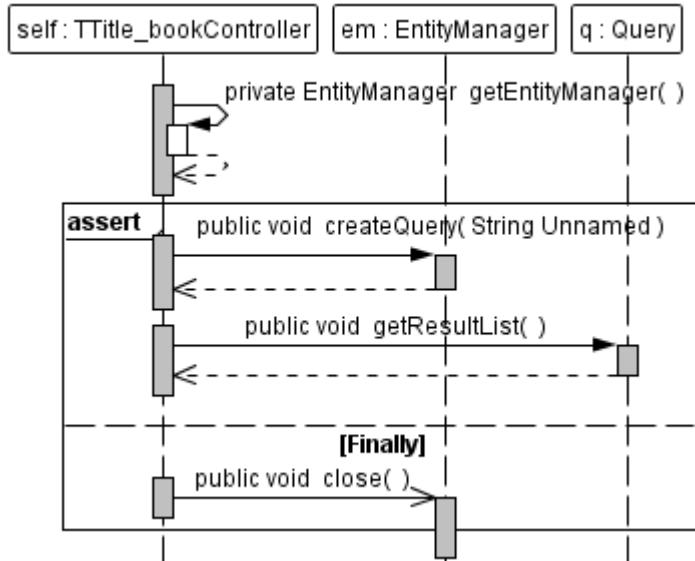


(2) The TTitle_bookController class: Adding titles in the database
public boolean addTTitle_books (ArrayList<TTitle_book> titles)

```
public boolean addTTitle_books(ArrayList<TTitle_book> titles) {  
    EntityManager em = getEntityManager();  
    TTitle_book newTTitle_book = null;  
    try {  
        Iterator it = titles.iterator();  
        em.getTransaction().begin();  
        while (it.hasNext()) {  
            newTTitle_book = (TTitle_book) it.next();  
            if (newTTitle_book.getId() == null) {  
                em.persist(newTTitle_book);  
            }  
        }  
        em.getTransaction().commit();  
    } finally {  
        em.close();  
        return false;  
    }  
}
```

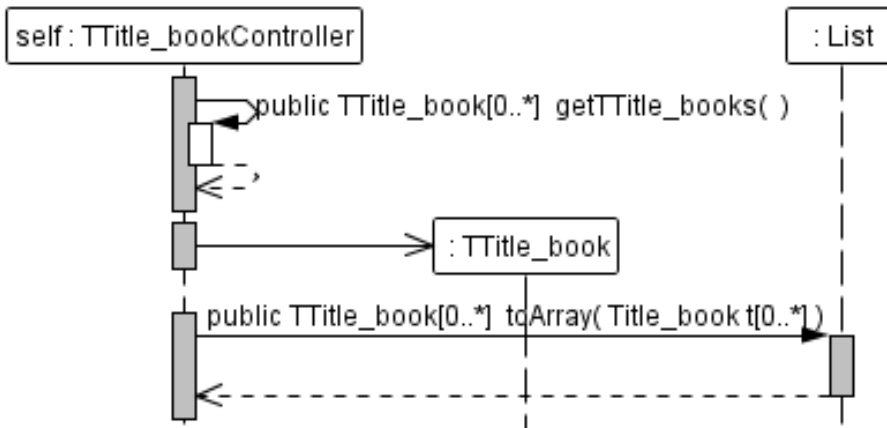


(1) Getting TTitle_book and TTitle_book_on_tape objects from the database (design pattern Transfer Object – TTitle_book[0..*])



```
public List<TTitle_book> getTTitle_books()
```

```
public TTitle_book[] getTTitle_books_()
```





(2) Getting TTitle_book and TTitle_book_on_tape objects from the database (design pattern Transfer Object – TTitle_book[0..*])

```
public TTitle_book[] getTTitle_books_() {  
    return (TTitle_book[]) getTTitle_books().toArray(new TTitle_book[0]);  
}  
  
public List<TTitle_book> getTTitle_books() {  
    EntityManager em = getEntityManager();  
    try {  
        javax.persistence.Query q =  
            em.createQuery("select c from TTitle_book as c");  
        return q.getResultList();  
    } finally {  
        em.close();  
    }  
}  
}  
} // end of TTitle_bookController
```



TLibrary1 - NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

- CustomLayoutDemo
- DialogDemo
- FlowLayoutDemo
- Formularze
- GridBagLayoutDemo
- GridLayoutDemo
- Library_client_2014
- MenuSelectionManagerDemo
- Okna1
- Proba
- ServletStateless
- Szyfrowanie
- TableDialogEditDemo
- TableSelectionDemo
- TLibrary1
 - Source Packages
 - META-INF
 - persistence.xml
 - integration_tier
 - TBase.java
 - TBookController.java
 - TTitle_bookController.java
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTitle_book.java
 - TTitle_book_on_tape.java
 - Generated Sources (ap-source-output)
 - Libraries
 - EclipseLink (JPA 2.0) - eclipselink-2.3.2.jar
 - EclipseLink (JPA 2.0) - javax.persistence-2.0.3.jar
 - EclipseLink (JPA 2.0) - org.eclipse.persistence.jpa.jpql_1.0.1.jar
 - JDK 1.7 (Default)

...ava TBook_period.java TTitle_bookController.java TBookController.java

Source History

```
5 package integration_tier;
6
7 import java.util.ArrayList;
8 import java.util.Iterator;
9 import java.util.List;
10 import javax.persistence.EntityManager;
11 import javax.persistence.EntityManagerFactory;
12 import javax.persistence.Persistence;
13 import sub_business_tier.entities.TTitle_book;
14 /**...*/
15
16 public class TTitle_bookController {
17
18     private EntityManagerFactory emf = null;
19
20     private EntityManager getEntityManager() {
21         if (emf == null) {
22             emf = Persistence.createEntityManagerFactory("Library1PU");
23         }
24         return emf.createEntityManager();
25     }
26
27     public TTitle_book[] getTTitle_books_() {
28         return (TTitle_book[]) getTTitle_books().toArray(new TTitle_book[0]);
29     }
30
31     public List<TTitle_book> getTTitle_books() {
32         EntityManager em = getEntityManager();
33         try {
34             javax.persistence.Query q =
35                 em.createQuery("select c from TTitle_book as c");
36             return q.getResultList();
37         } finally {
38             em.close();
39         }
40     }
41 }
42
```

13 | 47 | INS



TLibrary1 - NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

<default config>

Projects Files Services

- GridBagLayoutDemo
- GridLayoutDemo
- Library_client_2014
- MenuSelectionManagerDemo
- Okna1
- Proba
- ServletStateless
- Szyfrowanie
- TableDialogEditDemo
- TableSelectionDemo
- TLibrary1
 - Source Packages
 - META-INF
 - persistence.xml
 - integration_tier
 - TBase.java
 - TBookController.java
 - TTitle_bookController.java
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTitle_book.java
 - TTitle_book_on_tape.java
 - Generated Sources (ap-source-output)
 - Libraries
 - EclipseLink (JPA 2.0) - eclipselink-2.3.2.jar
 - EclipseLink (JPA 2.0) - javax.persistence-2.0.3.jar
 - EclipseLink (JPA 2.0) - org.eclipse.persistence.jpa.jpql_1.0.1.jar
 - JDK 1.7 (Default)

Source History

```
44 public boolean addTTitle_book(TTitle_book title_book) {
45     EntityManager em = getEntityManager();
46     try {
47         em.getTransaction().begin();
48         em.persist(title_book);
49         em.getTransaction().commit();
50     } finally {
51         em.close();
52         return false;
53     }
54 }
55
56 public boolean addTTitle_books(ArrayList<TTitle_book> titles) {
57     EntityManager em = getEntityManager();
58     TTitle_book newTTitle_book = null;
59     try {
60         Iterator it = titles.iterator();
61         em.getTransaction().begin();
62         while (it.hasNext()) {
63             newTTitle_book = (TTitle_book) it.next();
64             if (newTTitle_book.getId() == null) {
65                 em.persist(newTTitle_book);
66             }
67         }
68         em.getTransaction().commit();
69     } finally {
70         em.close();
71         return false;
72     }
73 }
74 }
```



4.3. Creating the TBookController.java in the integration_tier package

```
import javax.persistence.Persistence;
import library1.TBook;

public class TBookController {
    private EntityManagerFactory emf = null;

    private EntityManager getEntityManager() {
        if (emf == null) {
            emf = Persistence.createEntityManagerFactory("Library1PU");
        }
        return emf.createEntityManager();
    }
}
```



The TBookController class: Getting books from the database
public TBook[] getTBooks_(), public List<TBook> getTBooks()

```
public TBook[] getTBooks_() {  
    return (TBook[]) getTBooks().toArray(new TBook[0]); }  
}
```

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



The TBookController class: Adding a book in the database
public boolean addTTitle_books (ArrayList<TTitle_book> titles)

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



The TBookController class: Adding books in the database

public boolean addTBooks (ArrayList<TTitle_book> titles)

```
public boolean addTBooks(ArrayList<TTitle_book> titles) {
    EntityManager em = getEntityManager();
    TBook newBook = null;
    Iterator it = titles.iterator();
    em.getTransaction().begin();
    try {
        while (it.hasNext()) {
            TTitle_book newTitle_book = (TTitle_book) it.next();
            if (newTitle_book.getId() == null)
                continue;
            Iterator it_ = newTitle_book.getMBooks().iterator();
            while (it_.hasNext()) {
                newBook = (TBook) it_.next();
                if (newBook.getId() == null)
                    em.persist(newBook); }
            }
            em.getTransaction().commit();
        } finally {
            em.close();
            return false; }
    }
} // end of TBookController
```



TLibrary1 - NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

- ClientEditor
- CustomLayoutDemo
- DialogDemo
- FlowLayoutDemo
- Formularze
- GridBagLayoutDemo
- GridLayoutDemo
- Library_client_2014
- MenuSelectionManagerDemo
- Okna1
- Proba
- ServletStateless
- Szyfrowanie
- TableDialogEditDemo
- TableSelectionDemo
- TLibrary1
 - Source Packages
 - META-INF
 - persistence.xml
 - integration_tier
 - TBase.java
 - TBookController.java**
 - TTitle_bookController.java
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTitle_book.java
 - TTitle_book_on_tape.java
 - Generated Sources (ap-source-output)
 - Libraries
 - EclipseLink (JPA 2.0) - eclipselink-2.3.2.jar
 - EclipseLink (JPA 2.0) - javax.persistence-2.0.3.jar
 - EclipseLink (JPA 2.0) - org.eclipse.persistence.jpa.jpql_1.0.1.jar
 - JDK 1.7 (Default)

Source History

```
4  /*
5  package integration_tier;
6
7  import sub_business_tier.entities.TBook;
8  import java.util.ArrayList;
9  import java.util.Iterator;
10 import java.util.List;
11 import javax.persistence.EntityManager;
12 import javax.persistence.EntityManagerFactory;
13 import javax.persistence.Persistence;
14 import sub_business_tier.entities.TTitle_book;
15 /**...*/
16
17 // @RunAs("administrator")
18
19 public class TBookController {
20
21     private EntityManagerFactory emf = null;
22
23
24     private EntityManager getEntityManager() {
25         if (emf == null) {
26             emf = Persistence.createEntityManagerFactory("Library1PU");
27         }
28         return emf.createEntityManager();
29     }
30
31     public TBook[] getTBooks_() {
32         return (TBook[]) getTBooks().toArray(new TBook[0]);
33     }
34
35     public List<TBook> getTBooks() {
36         EntityManager em = getEntityManager();
37         try {
38             javax.persistence.Query q = em.createQuery("select c from TBook as c");
39             return q.getResultList();
40         } finally {
41             em.close();
42         }
43     }
44 }
```




TLibrary1 - NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

- CardLayoutDemo
- ClientEditor
- CustomLayoutDemo
- DialogDemo
- FlowLayoutDemo
- Formularze
- GridBagLayoutDemo
- GridLayoutDemo
- Library_client_2014
- MenuSelectionManagerDemo
- Okna1
- Proba
- ServletStateless
- Szyfrowanie
- TableDialogEditDemo
- TableSelectionDemo
- TLibrary1
 - Source Packages
 - META-INF
 - persistence.xml
 - integration_tier
 - TBase.java
 - TBookController.java**
 - TTitle_bookController.java
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTitle_book.java
 - TTitle_book_on_tape.java
 - Generated Sources (ap-source-output)
 - Libraries
 - EclipseLink (JPA 2.0) - eclipselink-2.3.2.jar
 - EclipseLink (JPA 2.0) - javax.persistence-2.0.3.jar
 - EclipseLink (JPA 2.0) - org.eclipse.persistence.jpa.jpql_1.0.1.jar
 - JDK 1.7 (Default)

Source History

```
45 public boolean addTBook(TBook book) {
46     EntityManager em = getEntityManager();
47     try {
48         em.getTransaction().begin();
49         em.persist(book);
50         em.getTransaction().commit();
51     } finally {
52         em.close();
53         return false;
54     }
55 }
56
57 public boolean addTBooks(ArrayList<TTitle_book> titles) {
58     EntityManager em = getEntityManager();
59     TBook newBook = null;
60     Iterator it = titles.iterator();
61     em.getTransaction().begin();
62     try {
63         while (it.hasNext()) {
64             TTitle_book newTitle_book = (TTitle_book) it.next();
65             if (newTitle_book.getId() == null) {
66                 continue;
67             }
68             Iterator it_ = newTitle_book.getMBooks().iterator();
69             while (it_.hasNext()) {
70                 newBook = (TBook) it_.next();
71                 if (newBook.getId() == null) {
72                     em.persist(newBook);
73                 }
74             }
75         }
76         em.getTransaction().commit();
77     } finally {
78         em.close();
79         return false;
80     }
81 }
```

14 | 47 | INS



5. You may add annotation to your own new classes and create the proper controllers – for higher assessment (5.0 or 5.5)