



Wrocław University of Technology

Master programmes in English
at Wrocław University of Technology



Projekt aplikacji wielowarstowej typu Java EE 6.0

Autor: Zofia Kruczkiewicz



HUMAN CAPITAL
HUMAN – BEST INVESTMENT!



Wrocław University of Technology

EUROPEAN
SOCIAL FUND





Projekt aplikacji wielowarstowej typu Java EE 6.0

- Warstwa klienta: klient internetowy oraz aplikacji
- Warstwa prezentacji oparta na technologii JSF 2.1
- Warstwa biznesowa oparta na komponentach EJB
- Warstwa integracji oparta na technologii ORM (JPA 2.0)
- Warstwa zasobów oparta bazie danych JavaDerby





Design patterns used to build the Integration nad Resources Tiers

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

Outline of creating the Library Catalogue Java Application

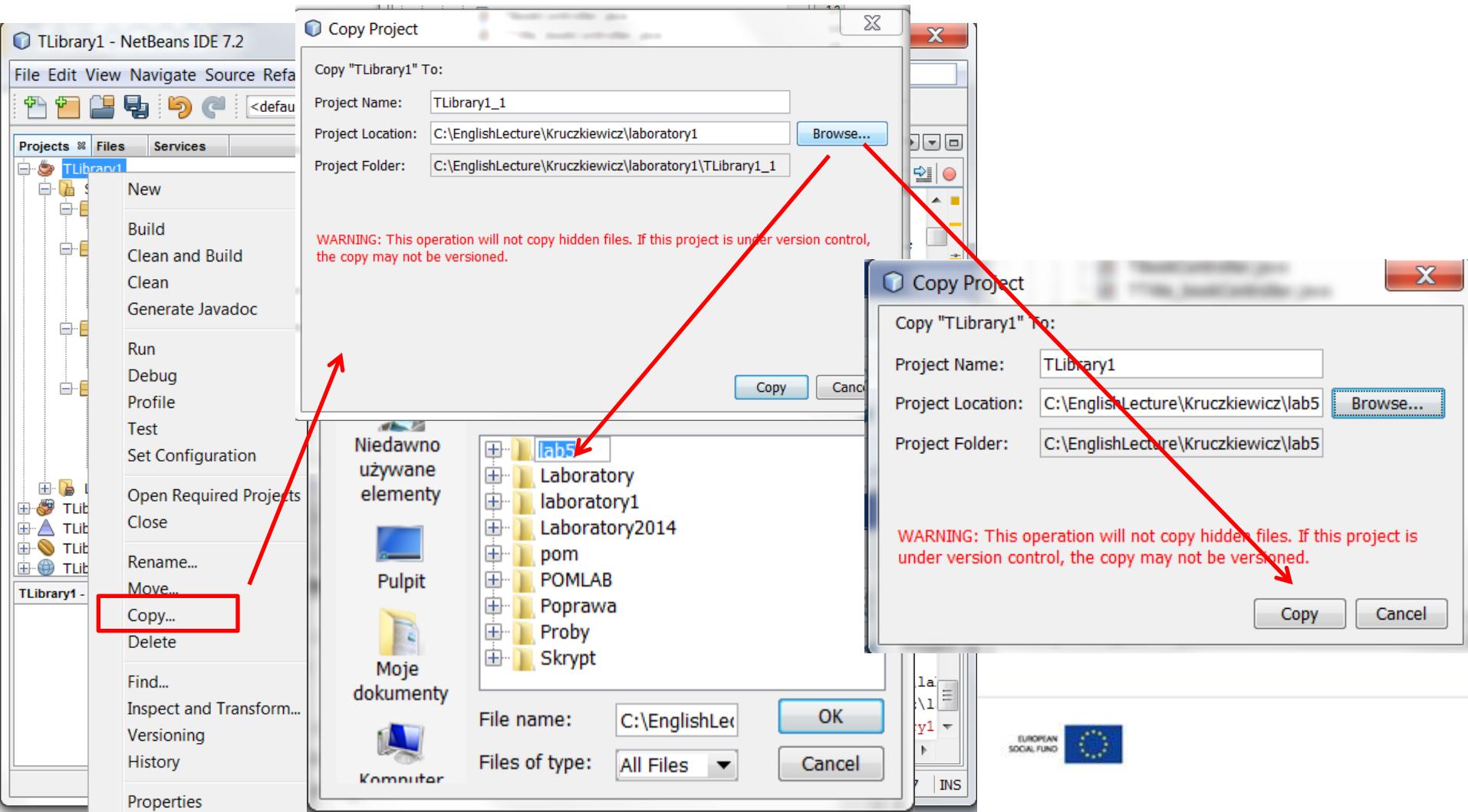
During previous step you must do:

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)





1.0. Create the new folder – lab5 and make the copy of the TLibrary1 (any name) project as the result of your works on lab4





2.0. Now, to this copy of TLibrary1 (the Java Class Library type of project) you must add the **new class named TBase**, as the TFacade design pattern, with methods **for database handling**. Below – it is code of this class.

```
package integration_tier;

import java.util.ArrayList;
import java.util.List;
import sub_business_tier.TFacade;
import sub_business_tier.entities.TBook;
import sub_business_tier.entities.TTitle_book;

public class TBase {

    private TTitle_bookController titleJpaController;
    private TBookController bookJpaController;
    private TFacade facade;
    private TTitle_book titles[];
    private TBook books[];
```

JPA controller for database handling as the ORM process for objects of the TTitle_book family

JPA controller for database handling as the ORM process for objects of the TBook family

TFacade type of object as provider of the application data from data object model consisting of objects of Ttitle_book and TBook families

The cache of TTitle_book objects

The cache of TBook objects





2.1. The continuation of the TBase class code

```
public TBase(TFacade facade_) {  
    facade = facade_;  
    titleJpaController = new TTITLE_bookController();  
    bookJpaController = new TBookController();  
    try {  
        update_data();  
    } catch (Exception e) {}  
}
```

```
public void update_data() throws Exception {  
    update_titles();  
    update_books();  
    facade.update_data(titles, books);  
}
```

You must add the **update_data** method to the TFaçade class (TLibrary1 project)

The constructor creating the JPA controllers and updating the application data based on data from database using ORM technology

```
public synchronized void update_data(TTITLE_book titles[],  
    TBook books[]) {  
    mTitle_books.clear();  
    for (TTITLE_book t: titles) {  
        mTitle_books.add(t);  
    }  
    for (TTITLE_book title: mTitle_books) {  
        for (TBook book: books) {  
            TTITLE_book title1 = book.getmTitle_book();  
            if (title1 != null) {  
                if (title1.equals(title)) {  
                    title.getmBooks().add(book);  
                }  
            }  
        }  
    }  
}
```





2.2. The continuation of the **TBase class code**

```
public void update_titles() throws Exception {  
    titles = (TTitle_book[]) titleJpaController.getTTitles_books_();  
}
```

The methods reading the data from database by using the JPA controllers

```
public void update_books() throws Exception {  
    books = (TBook[]) bookJpaController.getTBooks_();  
}
```

```
public void add_titles() throws Exception {  
    try {  
        titleJpaController.addTTitles_books(facade.getmTitle_books());  
    } catch (Exception e) {  
    }  
}
```

The methods storing the application data in database by using the JPA controllers

```
public void add_books() throws Exception {  
    try {  
        bookJpaController.addTBooks(facade.getmTitle_books());  
    } catch (Exception e) {  
    }  
}
```





2.3. The continuation of the **TBase class code**

```
public ArrayList<ArrayList<String>> titles() throws Exception {  
    List<TTitle_book> help1 = titleJpaController.getTTitle_books();  
    ArrayList<ArrayList<String>> help2 = new ArrayList();  
    for (TTITLE_book t : help1) {  
        ArrayList<String> help3 = new ArrayList();  
        help3.add(t.getPublisher());  
        help3.add(t.getISBN());  
        help3.add(t.getTitle());  
        help3.add(t.getAuthor());  
        help3.add(t.getActor());  
        help2.add(help3);  
    }  
    return help2;  
}
```

The method reading the data of the TTitle_book family from database by using the JPA controller (titleJpaController) and preparing the data model for the JSF h:dataTable view component





3.0. Creation the **Java EE project** with EE module and Web module for the Web Client Tier, and the Enterprise Application Client Tier – **Chosse Project**

New Project X

Steps

1. Choose Project
2. ...

Choose Project

Categories:

- Java
- JavaFX
- Java Web
- Java EE
- Java Card
- Java ME
- Maven
- PHP
- Groovy
- C/C++
- NetBeans Modules
- Samples

Projects:

- Enterprise Application
- Enterprise Application with Existing Sources
- EJB Module
- EJB Module with Existing Sources
- Enterprise Application Client
- Enterprise Application Client with Existing Sources

Description:

Creates a new enterprise application in a standard project. You can also create an EJB module project and Web application project in the enterprise application. A standard project uses an **IDE-generated Ant build script** to build and run your projects.

[< Back](#) [Next >](#) [Finish](#) [Cancel](#) [Help](#)



3.1. Creation the **Java EE project (TLibrary2_EE)** with EE module and Web module for the Web Client Tier, and the Enterprise Application Client Tier – **Name and Location (lab5 – with the Library project TLibrary1)**

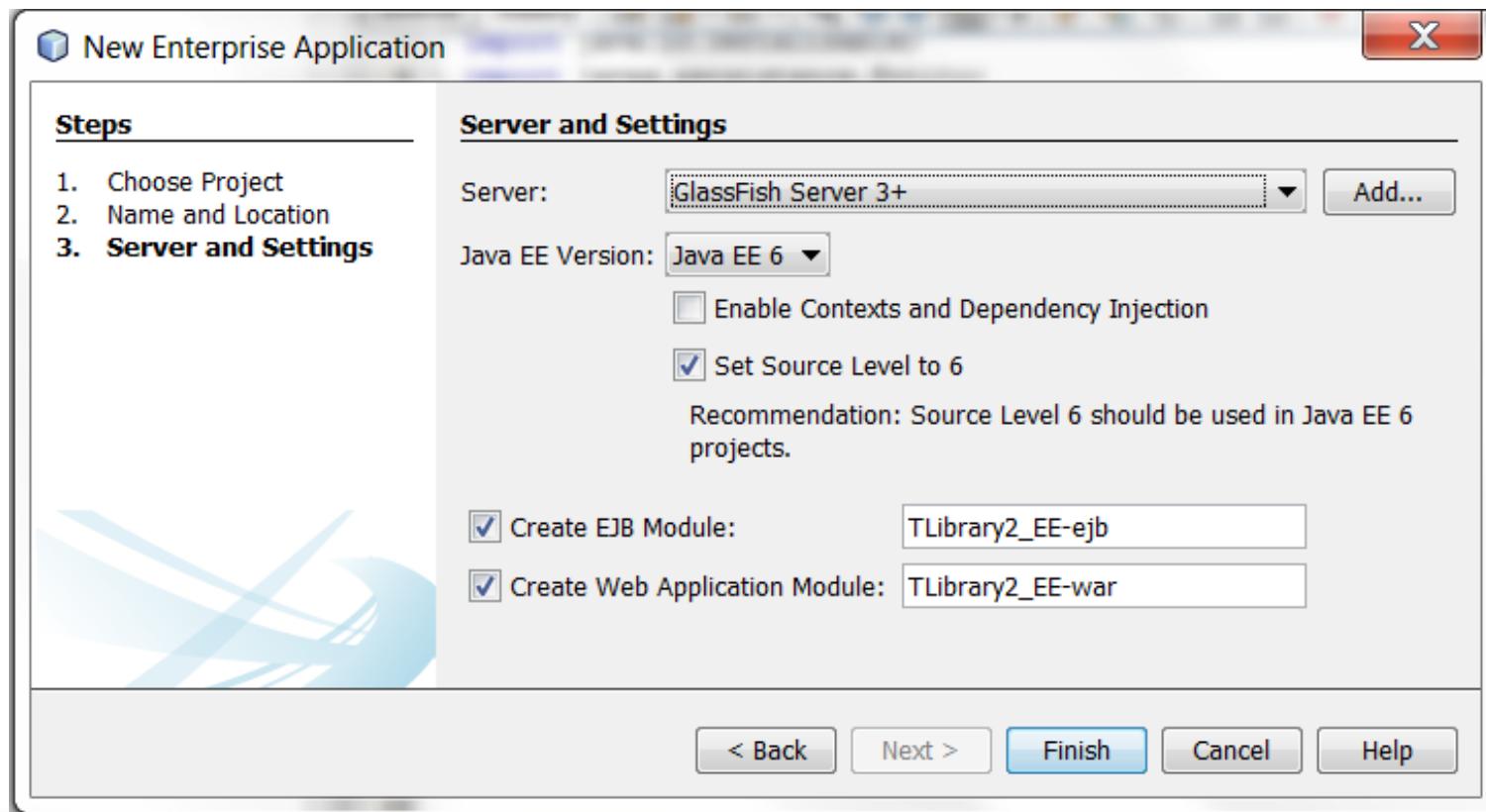
New Enterprise Application X

Steps	Name and Location
1. Choose Project	Project Name: <input type="text" value="TLibrary2_EE"/>
2. Name and Location	Project Location: <input type="text" value="C:\EnglishLecture\Kruczkiewicz\lab5"/> <input type="button" value="Browse..."/>
3. Server and Settings	Project Folder: <input type="text" value="C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_EE"/>
	<input type="checkbox"/> Use Dedicated Folder for Storing Libraries Libraries Folder: <input type="text"/> <input type="button" value="Browse..."/>
	Different users and projects can share the same compilation libraries (see Help for details).
	<input type="button" value="< Back"/> <input type="button" value="Next >"/> <input type="button" value="Finish"/> <input type="button" value="Cancel"/> <input type="button" value="Help"/>





3.2. Creation the **Java EE project (TLibrary2_EE)** with **EE module (TLibrary2_EE-ejb)** and **Web module (TLibrary2_EE-war)** for the Web Client Tier, and the Enterprise Application Client Tier – Server and Settings (GlassFish Server 4.1 too)





3.3. The results of creation of projects: Java EE project (TLibrary2_EE) with EE module (TLibrary2_EE-ejb) and Web module (TLibrary2_EE-war)

NetBeans IDE 7.2

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

Search (Ctrl+I)

Projects Files Services

TLibrary1

- Source Packages
 - META-INF
 - persistence.xml
 - integration_tier
 - TBase.java
 - TBookController.java
 - TTtitle_bookController.java
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTtitle_book.java
 - TTtitle_book_on_tape.java
- Libraries
 - TLibrary2_EE
- Java EE Modules
 - TLibrary2_EE-war.war
 - TLibrary2_EE-ejb.jar
- Configuration Files
- Server Resources

TTtitle_book_on_tape.java

```
import java.io.Serializable;
import javax.persistence.Entity;

/**
 *
 * @author kruczakiewicz
 */
@Entity
public class TTtitle_book_on_tape extends TTtitle_book
    implements Serializable {

    private static final long serialVersionUID = 1L;
    private String actor;

    @Override
    public String getActor() {
        return actor;
    }

    @Override
    public void setActor(String val) {
        actor = val;
    }

    @Override
    public String toString() {
        String help = super.toString();
        help += " Actor: " + getActor();
        return help;
    }
}
```

Output

Java DB Database Process GlassFish Server 3+ SQL Command 3 execution

TLibrary1_client_ejb (run) running...

HUMAN



4.0. Addition of EJB component: Creation the **Java Class Library project (TLibrary2_interface_ejb)** as the interface for the Enterprise Session Bean component, which will be stored in **the EE module (TLibrary2_EE-ejb)**

The screenshot shows the NetBeans IDE 7.2 interface with the title bar "TLibrary2_client_ejb - NetBeans IDE 7.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profiler, Tear Off, Window, Help. A search bar "Search (Ctrl+I)" is at the top right. The main window displays the "New Project" wizard. The left sidebar lists recent actions: New Project..., New File..., Open Project..., Open Recent Project, Close Project (TLibrary2_client_ejb), Open File..., Open Recent File, Project Group, Project Properties (TLibrary2_client_ejb), Import Project, Export Project, Save (Ctrl+S), Save As..., Save All (Ctrl+Shift+S), Page Setup..., Print... (Ctrl+Alt+Shift+P), Print to HTML..., and Exit. The central "New Project" window has two tabs: "Steps" and "Choose Project". The "Choose Project" tab shows "Categories: Java" with sub-options Java, JavaFX, Java Web, Java EE, Java Card, Java ME. The "Projects:" section lists Java Application, Java Class Library (which is selected and highlighted in blue), Java Project with Existing Sources, and Java Free-Form Project. A "Description:" box states: "Creates a new Java SE library in a standard IDE project. A Java SE library does not contain a main class. Standard projects use an IDE-generated Ant build script to build, run, and debug your..." At the bottom are buttons for < Back, Next >, Finish, Cancel, and Help.



New Java Class Library X

Steps

1. Choose Project
2. Name and Location

Name and Location

Project Name:

Project Location:

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder:

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





TLibrary2_EE-ejb - NetBeans IDE 7.2

File Edit View Naviga Sourc Refact Rui Debu Profi Tool Windo Help Search (Ctrl+I)

Projects Files Services

Source Packages
META-INF
integration_tier
sub_business_tier
sub_business_tier.entities
TBook.java
TBook_period.java
Title_book.java

New

- Java Class...
- Session Bean...
- Persistence Unit...
- Java Package...
- Session Beans For Entity Classes...
- GlassFish Descriptor...
- Entity Classes from Database...
- Session Beans For Entity Classes...
- Message-Driven Bean...
- Entity Class...
- Web Service...
- Web Service Client...

Build

Clean and Build

Clean

Verify

Generate Javadoc

Run

Deploy

Debug

Profile

Test Alt+F6

Open Required Projects

Close

Rename...

EUROPEAN SOCIAL FUND

4.1. Addition of EJB component: Addition of the **Facade class as the Enterprise Session Bean type to the **the EE module (TLibrary2_EE-ejb)** - Choose File Type**



New File X

Steps

1. Choose File Type
2. ...

Choose File Type

Project: **TLibrary2_EE-ejb**

Categories:

- Bean Validation
- Enterprise JavaBeans
- Contexts and Dependenc
- Java
- JavaBeans Objects
- Unit Tests
- Persistence
- Groovy
- Web Services
- XML
- ClassFiles

File Types:

- Session Bean**
- Timer Session Bean
- Message-Driven Bean
- Service Locator
- Caching Service Locator
- Session Beans For Entity Classes
- Standard Deployment Descriptor

Description:

Creates an empty Session Enterprise JavaBean (EJB) component. A session bean is typically used to encapsulate business logic or enterprise resources. This template creates the Java classes for a single session bean and registers the bean in the EJB module's deployment descriptor, if required.

< Back **Next >** Finish Cancel Help





4.2. Addition of EJB component: Addition of the **Facade class as the Enterprise Session Bean** type to the **the EE module (TLibrary2_EE-ejb)** - Name and Location

New Session Bean

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

EJB Name:

Project:

Location: ▾

Package: ▾

Session Type:

Stateless

Stateful

Singleton

Create Interface:

Local

Remote in project: ▾

< Back Cancel Help



4.3. Addition of EJB component:
Content of the interface of the Facade EJB component – as the **FacadeRemote** interface in the **TLibrary2_interface_ejb** project. It is the sum of public declaration of methods of the **TFacade** and **TBase** classes from the **TLibrary1** project

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Search (Ctrl+I)
Projects Files Services <default config> FacadeRemote.java
Source History ...
1 package business_tier;
2
3 import java.util.ArrayList;
4 import javax.ejb.Remote;
5
6 @Remote
7 public interface FacadeRemote {
8
9     // methods for application data service
10    public Object[][] gettitle_books();
11
12    public ArrayList<String> add_title_book(String data[]);
13
14    public ArrayList<String> add_book(String data1[], String data2[]);
15
16    public ArrayList<String> Search_title_book(String data[]);
17
18    // methods for database service
19    public void update_titles() throws Exception;
20
21    public void update_books() throws Exception;
22
23    public void update_data() throws Exception;
24
25    public void add_titles() throws Exception;
26
27    public void add_books() throws Exception;
28
29    public ArrayList<ArrayList<String>> titles() throws Exception;
30
31 }
32
business_tier.FacadeRemote >
Output
Java DB Database Process > GlassFish Server 4.1 >
at com.sun.enterprise.v3.admin.CommandRunnerImpl.doCommand...
nd(CommandRunnerImpl.java:1464)
at com.sun.enterprise.v3.admin.CommandRunnerImpl.access$...
1300(CommandRunnerImpl.java:109)
at com.sun.enterprise.v3.admin.CommandRunnerImpl$Executi...
onContext.execute(CommandRunnerImpl.java:1846)
at com.sun.enterprise.v3.admin.CommandRunnerImpl$Executi...
INS
```



4.4. The code of FacadeRemote interface

```
package business_tier;

import java.util.ArrayList;
import javax.ejb.Remote;

@Remote
public interface FacadeRemote {
    // methods for application data services – from TFacade class

    public Object[][] gettitle_books();

    public ArrayList<String> add_title_book(String data[]);

    public ArrayList<String> add_book(String data1[], String data2[]);

    public ArrayList<String> Search_title_book(String data[]);
}
```



4.6. The code of FacadeRemote interface - continuation

// methods for database service – from TBase class

```
public void update_titles() throws Exception;  
  
public void update_books() throws Exception;  
  
public void update_data() throws Exception;  
  
public void add_titles() throws Exception;  
  
public void add_books() throws Exception;  
  
public ArrayList<ArrayList<String>> titles() throws Exception;  
}
```





4.7. You must add project TLibrary1 from lab5 folder to the Librray folder of the TLibrary2_EE-ejb module

TLibrary2_EE-ejb - NetBeans IDE 7.2

File Ed Vie Navig Sour Refac Ru Deb Prof Tea Toc Wind Help Search (Ctrl+I)

Projects Files Services

- TableDialogEditDemo
- TableSelectionDemo
- TLibrary1
- TLibrary2_EE
 - Java EE Modules
 - TLibrary2_EE-war.war
 - TLibrary2_EE-ejb.jar
 - Configuration Files
 - Server Resources
- TLibrary2_EE-ejb
 - Source Packages
 - business tier
 - Facade.java
- Lib
 - Add Project...
 - Add Library...
 - Add JAR/Folder...
 - Properties

FacadeRemote.java...

```
public void Print_
    facade.Print_t
}
// definition of method
public void update
    base.update_ti
}

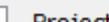
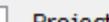
public void update
```

Add Project

Niedawno
używane
elementy

Look in: lab5

- + TLibrary1
- + TLibrary2_EE
- + TLibrary2_interface_ejb



Project Name:

TLibrary1

Project JAR Files:

dist/TLibrary1.jar

File name: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary1

Add Project JAR Files

Files of type: Project Folder

Cancel



4.8. It is code of the **Facade** of the Session Bean type class.

```
package business_tier;

import integration_tier.TBase;
import java.util.ArrayList;
import javax.ejb.Stateless;
import sub_business_tier.TFacade;

@Stateless
public class Facade implements FacadeRemote {

    TFacade facade = new TFacade();
    TBase base = new TBase(facade);
```



The code of Facade class - continuation

// definition of methods for application data service

```
public Object[][] gettitle_books() {  
    return facade.gettitle_books();  
}
```

```
public ArrayList<String> add_title_book(String data[]) {  
    return facade.add_title_book(data);  
}
```

```
public ArrayList<String> add_book(String data1[], String data2[]) {  
    return facade.add_book(data1, data2);  
}
```

```
public ArrayList<String> Search_title_book(String data[]) {  
    return facade.Search_title_book(data);  
}
```



The code of Facade class - continuation

// definition of methods for database service

```
public void update_titles() throws Exception {  
    base.update_titles();  
}  
  
public void update_books() throws Exception {  
    base.update_books();  
}  
  
public void update_data() throws Exception {  
    base.update_data();  
}  
  
public void add_titles() throws Exception {  
    base.add_titles();  
}  
  
public void add_books() throws Exception {  
    base.add_books();  
}  
  
public ArrayList<ArrayList<String>> titles() throws Exception {  
    return base.titles();  
}
```



5. Creation of the EE client tier

The screenshot shows the NetBeans IDE 8.0.2 interface. The title bar reads "TLibrary2_EE-ejb - NetBeans IDE 8.0.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The search bar at the top right says "Search (Ctrl+I)". The left sidebar shows the "File" menu with options like New Project..., Open Project..., Import Project, Save, and Exit. Below the menu is a tree view of the project structure under "TLibrary2_EE-war": Libraries, Enterprise Beans, Configuration Files, Server Resources, TLibrary2_EE-war, and TLibrary2_interface_ejb. Under TLibrary2_interface_ejb, there are Source Packages (business_tier) containing FacadeRemote.java and Libraries. The main editor window displays the code for FacadeRemote.java:

```
// Add business logic below. (Right-click in editor and choose
// "Insert Code > Add Business Method")
// Add business logic below. (Right-click in editor and choose
// "Insert Code > Add Business Method")
TFacade facade = new TFacade();
TBase base = new TBase(facade);

// definition of methods for application data service
public Object[][] gettitle_books() {
    return facade.gettitle_books();
}

public ArrayList<String> add_title_book(String data[]) {
    return facade.add_title_book(data);
}

public ArrayList<String> add_book(String data1[], String data2[]) {
    return facade.add_book(data1, data2);
}

public ArrayList<String> Search_title_book(String data[]) {
    return facade.Search_title_book(data);
}

// definition of methods for database service
public void update_titles() throws Exception {
    base.update_titles();
}
```

The Output window at the bottom shows logs from Java DB Database Process and GlassFish Server 4.1. The GlassFish log includes entries for HttpHandler.runService, HttpHandler.doHandle, HttpServerFilter.handleRead, ExecutorResolver\$9.execute, and ExecutorResolver\$9.





5.1. Creation the Enterprise Application Client Tier - **Choose Project**

The screenshot shows the 'New Project' dialog box with the title 'Choose Project'. On the left, under 'Steps', it says '1. Choose Project' and '2. ...'. The main area is titled 'Choose Project' and contains two sections: 'Categories:' and 'Projects:'. In 'Categories:', there are icons for Java, JavaFX, Java Web, Java EE, Java Card, Java ME, Maven, and PHP. In 'Projects:', there are icons for Enterprise Application, Enterprise Application with Existing Sources, EJB Module, EJB Module with Existing Sources, Enterprise Application Client (which is selected and highlighted in blue), and Enterprise Application Client with Existing Sources. Below these sections is a 'Description:' box containing the text: 'Creates a new Enterprise application client project in a standard IDE project. Standard projects use an IDE-generated Ant build script to build, run, and debug the project.' At the bottom of the dialog are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.





5.2. Creation the Enterprise Application Client Tier – Name and Location

New Enterprise Application Client X

Steps

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

Name and Location

Project Name:

Project Location: Browse...

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

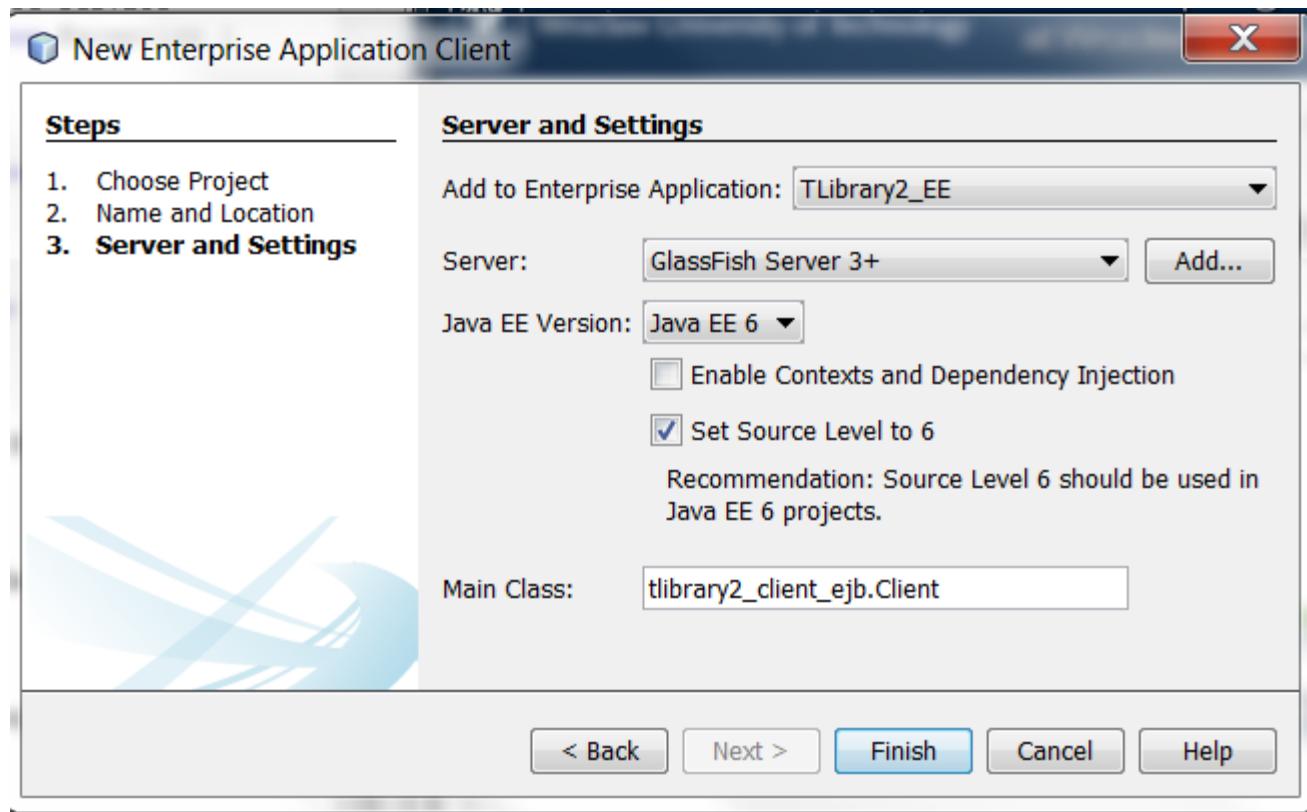
Different users and projects can share the same compilation libraries (see Help for details).

< Back Next > Finish Cancel Help





5.3. Creation the Enterprise Application Client Tier – **Server and Settings**.





5.4. Code of the EE client based on code of SE client – only change of TFacade class from POJO SE TFacade class into the EJB Facade class

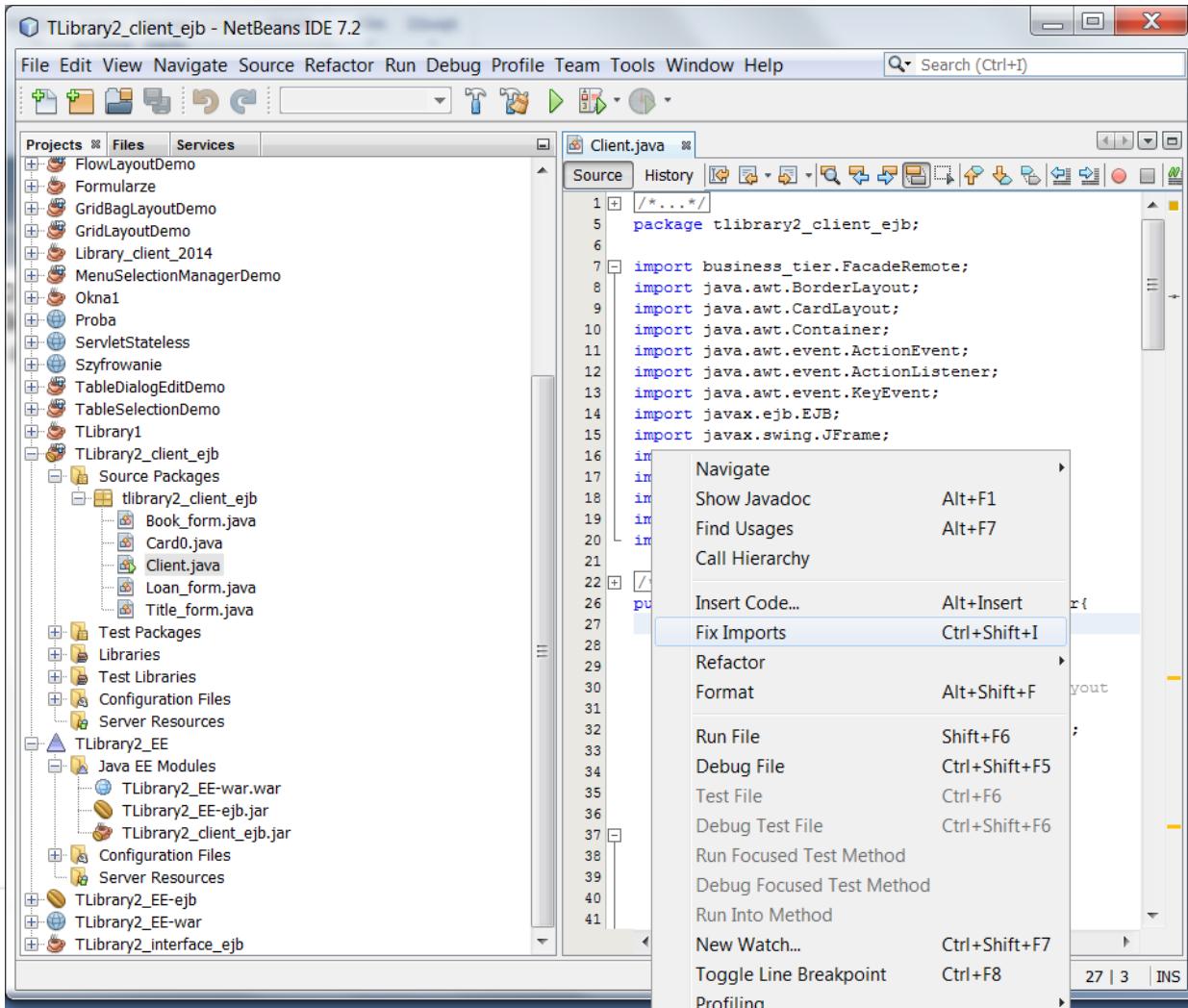
The screenshot shows the NetBeans IDE 7.2 interface with the title bar "TLibrary2_client_ejb - NetBeans IDE 7.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. A search bar "Search (Ctrl+I)" is also present. The left pane displays the project structure under "Projects": FlowLayoutDemo, Formularze, GridBagLayoutDemo, GridLayoutDemo, Library_client_2014, MenuSelectionManagerDemo, Okna1, Proba, ServletStateless, Szyfrowanie, TableDialogEditDemo, TableSelectionDemo, TLibrary1, and TLibrary2_client_ejb. The TLibrary2_client_ejb node has "Source Packages" expanded, showing sub-packages like tlibrary2_client_ejb containing Book_form.java, Card0.java, Client.java (the current file being edited), Loan_form.java, and Title_form.java. Below "Source Packages" are "Test Packages", "Libraries", "Test Libraries", "Configuration Files", and "Server Resources". The "TLibrary2_EE" node contains "Java EE Modules" with TLibrary2_EE-war.war, TLibrary2_EE-ejb.jar, and TLibrary2_client_ejb.jar. It also has "Configuration Files" and "Server Resources". The "TLibrary2_EE-ejb" and "TLibrary2_EE-war" nodes are shown below. The "TLibrary2_interface_ejb" node is at the bottom. The right pane shows the code editor for "Client.java". The code implements the ActionListener interface and creates a JMenuBar. It defines several static strings: NOTHING1, TITLE, BOOK, and LOAN. It creates a menu bar with a single menu item for "TITLE". The code uses Java Swing components like JPanel, JMenuBar, JMenu, and JMenuItem.

```
22  /** ... */
26  public class Client implements ActionListener{
27
28
29
30      JPanel cards; //a panel that uses CardLayout
31      final static String NOTHING1 = "Empty1";
32      final static String TITLE = "Title form";
33      final static String BOOK = "Book form";
34      final static String LOAN = "Loan book";
35      //static TFacade facade = new TFacade();
36
37      public JMenuBar createMenuBar() {
38          JMenuBar menuBar;
39          JMenu menu, submenu;
40          JMenuItem menuItem;
41
42          //Create the menu bar.
43          menuBar = new JMenuBar();
44
45          menu = new JMenu("A Menu");
46          menu.setMnemonic(KeyEvent.VK_A);
47          menuBar.add(menu);
48
49          menuItem = new JMenuItem(TITLE, KeyEvent.VK_T);
50          menuItem.setMnemonic(KeyEvent.VK_T); //used
51          menuItem.setAccelerator(KeyboardStroke.getKeyStroke(
52              KeyEvent.VK_1, ActionEvent.ALT_MASK));
53          menuItem.addActionListener(this);
54          menu.add(menuItem);
55
56          menuItem = new JMenuItem(BOOK);
57          menuItem.setMnemonic(KeyEvent.VK_B);
58          menuItem.addActionListener(this);
59      }
60  }
```

Project co-financed from the EU European Social Fund

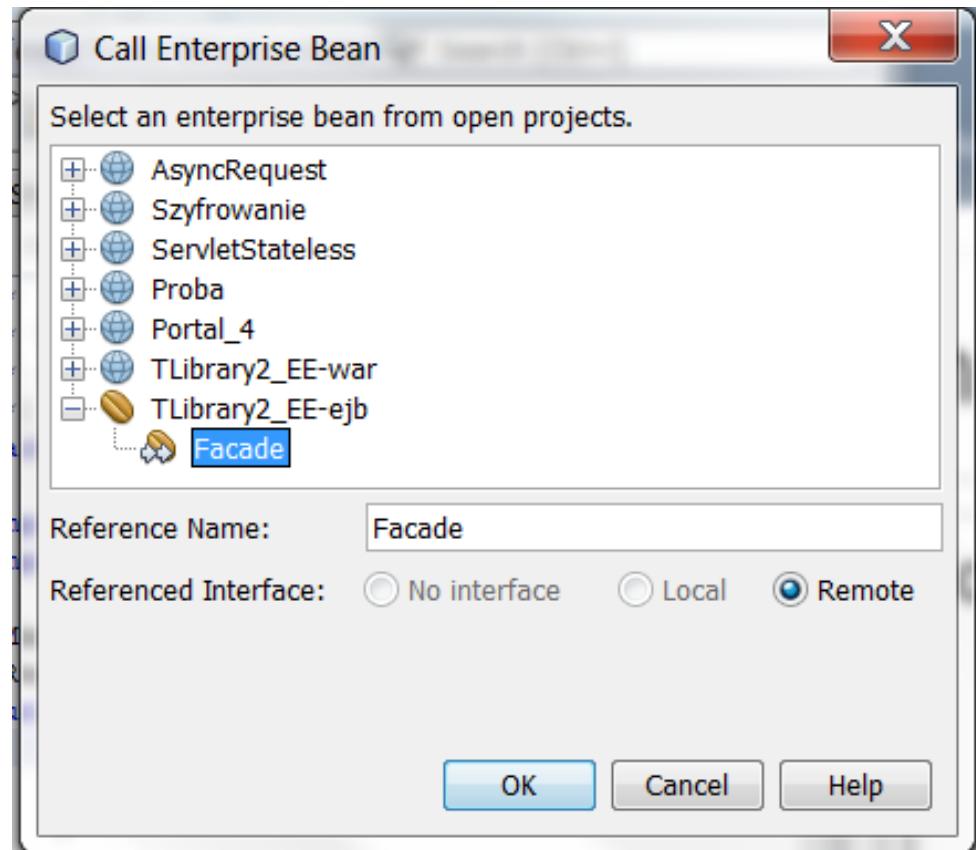
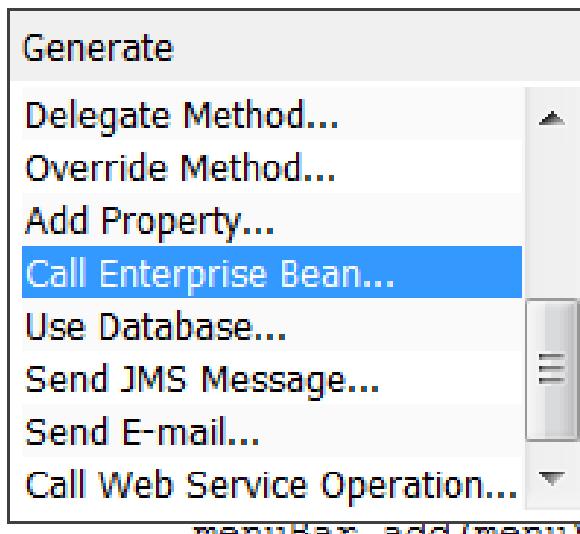


5.5. Creation of connection from the Facade Session Bean of Library_EJB1-ejb module to the Client class of the **TLibrary2_client_ejb** project - right click the edytor window with Client class and choose the Insert code item.





5.6. Select the Call Enterprise Bean item and then select the Facade EJB from the Library_EJB1-ejb module





5.7. The result

The screenshot shows the NetBeans IDE 7.2 interface with the project **TLibrary2_client_ejb** open. The left pane displays the project structure under the **Projects** tab, showing various Java files and packages. A blue arrow points from the text "The result" towards the code editor. The right pane shows the **Client.java** file with the following code:

```
1  /* */
2  package tlibrary2_client_ejb;
3
4  import business_tier.FacadeRemote;
5  import java.awt.BorderLayout;
6  import java.awt.CardLayout;
7  import java.awt.Container;
8  import java.awt.event.ActionEvent;
9  import java.awt.event.ActionListener;
10 import java.awt.event.KeyEvent;
11 import javax.ejb.EJB;
12 import javax.swing.JFrame;
13 import javax.swing.JMenu;
14 import javax.swing.JMenuBar;
15 import javax.swing.JPanel;
16 import javax.swing.JPanel;
17 import javax.swing.KeyStroke;
18
19 /**
20  */
21 public class Client implements ActionListener{
22     @EJB
23     private static FacadeRemote facade;
24
25     JPanel cards; //a panel that uses CardLayout
26     final static String NOTHING1 = "Empty1";
27     final static String TITLE = "Title form";
28     final static String BOOK = "Book form";
29     final static String LOAN = "Loan book";
30
31     // TFacade facade = new TFacade();
32
33     public JMenuBar createMenuBar() {
34         JMenuBar menuBar;
35         JMenu menu, submenu;
36         JMenuItem menuItem;
37
38         menuBar = new JMenuBar();
39         menu = new JMenu("File");
40         submenu = new JMenu("Edit");
41
42         menu.add(new JMenuItem("New"));
43         menu.add(new JMenuItem("Open"));
44         menu.add(new JMenuItem("Save"));
45
46         submenu.add(new JMenuItem("Cut"));
47         submenu.add(new JMenuItem("Copy"));
48         submenu.add(new JMenuItem("Paste"));
49
50         menuBar.add(menu);
51         menuBar.add(submenu);
52
53         return menuBar;
54     }
55
56     public void actionPerformed(ActionEvent e) {
57         if (e.getSource() instanceof JMenuItem) {
58             JMenuItem item = (JMenuItem) e.getSource();
59             if (item.getText().equals("New")) {
60                 new Book_form();
61             } else if (item.getText().equals("Open")) {
62                 new Card0();
63             } else if (item.getText().equals("Save")) {
64                 new Loan_form();
65             }
66         }
67     }
68 }
```



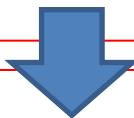


5.8. The change of getter and setter methods of the Facade class as the Session Bean type attribute

```
static TFacade facade = new TFacade();

public static TFacade getFacade() {
    return facade;
}

public static void setFacade(Facade façade) {
    this.facade = facade;
}
```



```
@EJB
private static FacadeRemote facade;

public static FacadeRemote getFacade() {
    return facade;
}

public static void setFacade(FacadeRemote façade) {
    Client.facade = facade;
}
```



5.9. The same code – different type of return value from client.getFacade()

```
public void actionPerformed(ActionEvent  
evt) {  
    String[] data = form_title();  
    if (data == null) {  
        return;  
    }  
  
    Client.getFacade().add_title_book(data);  
}
```

Object POJO in SE
Client tier

```
public void actionPerformed(ActionEvent  
evt) {  
    String[] data = form_title();  
    if (data == null) {  
        return;  
    }  
  
    Client.getFacade().add_title_book(data);  
}
```

EJB in EE Client
tier





6. 0. Definition of the Web Client Tier (**TLibrary2_EE-war**, prepared during creation of **TLibrary2_EE**) project) for inserting application data into the database and displaying the data from database – change the Web Framework to the JavaServer Faces type

The screenshot shows the NetBeans IDE interface. The title bar says "TLibrary2_EE-war - NetBeans IDE 7.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for New, Build, Clean, Verify, Generate Javadoc, Run, Deploy, Debug, Profile, Test RESTful Web Services, and Test. The Projects tab shows the project structure: TLibrary2_EE-war (Java EE Modules: TLibrary2_EE-war.war, TLibrary2_EE-ejb.jar, TLibrary2_client_ejb.jar), Configuration Files, Server Resources, TLibrary2_EE-ejb (Source Packages: business tier (Facade.java)), Libraries (TLibrary2_interface_ejb - dist/TLibrary2_interface_ejb.jar, TLibrary1 - dist/TLibrary1.jar), JDK 1.7 (Default), GlassFish Server 3+, Enterprise Beans, Configuration Files, and Server Resources. The Files tab shows files like Client.java, Title_form.java, Book_form.java, Card0.java, and Loan_form.java. The Services tab is empty. The right panel shows the code for Client.java:

```
public Container createContentPane() {
    //Create the content-pane-to-be.

    Card0 card0 = new Card0();
    Title_form card1 = new Title_form(this);
    Book_form card2 = new Book_form(this);
    Loan_form card3 = new Loan_form(this);

    //Create the panel that contains the "cards".
    cards = new JPanel(new CardLayout());
    cards.add(card0, NOTHING1);
    cards.add(card1, TITLE);
    cards.add(card2, BOOK);
    cards.add(card3, LOAN);

    JPanel p1 = new JPanel();

    p1.add(cards, BorderLayout.CENTER);
    return p1;
}

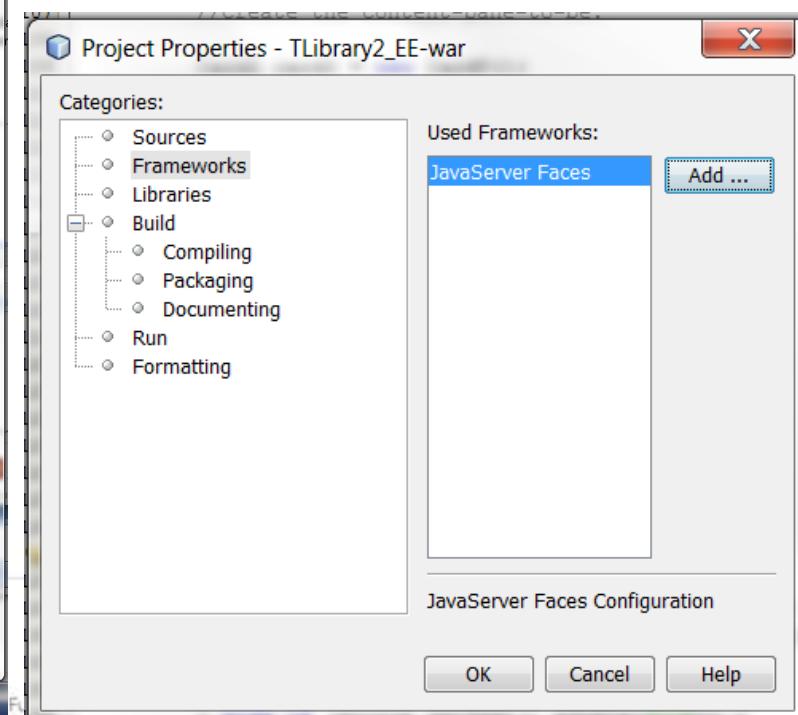
public void actionPerformed(ActionEvent e) {

    JMenuItem source = (JMenuItem) (e.getSource());
    CardLayout cl = (CardLayout) (cards.getLayout());
    if (source.getText().equals(TITLE)) {
        cl.show(cards, TITLE);
    } else if (source.getText().equals(BOOK)) {
        cl.show(cards, BOOK);
    } else if (source.getText().equals(NOTHING1)) {
        cl.show(cards, NOTHING1);
    } else if (source.getText().equals(LOAN)) {
        cl.show(cards, LOAN);
    }
}

/*
 * Create the GUI and show it. For thread safety, this method should be
 * invoked from the event-dispatching thread.
 */
}
```

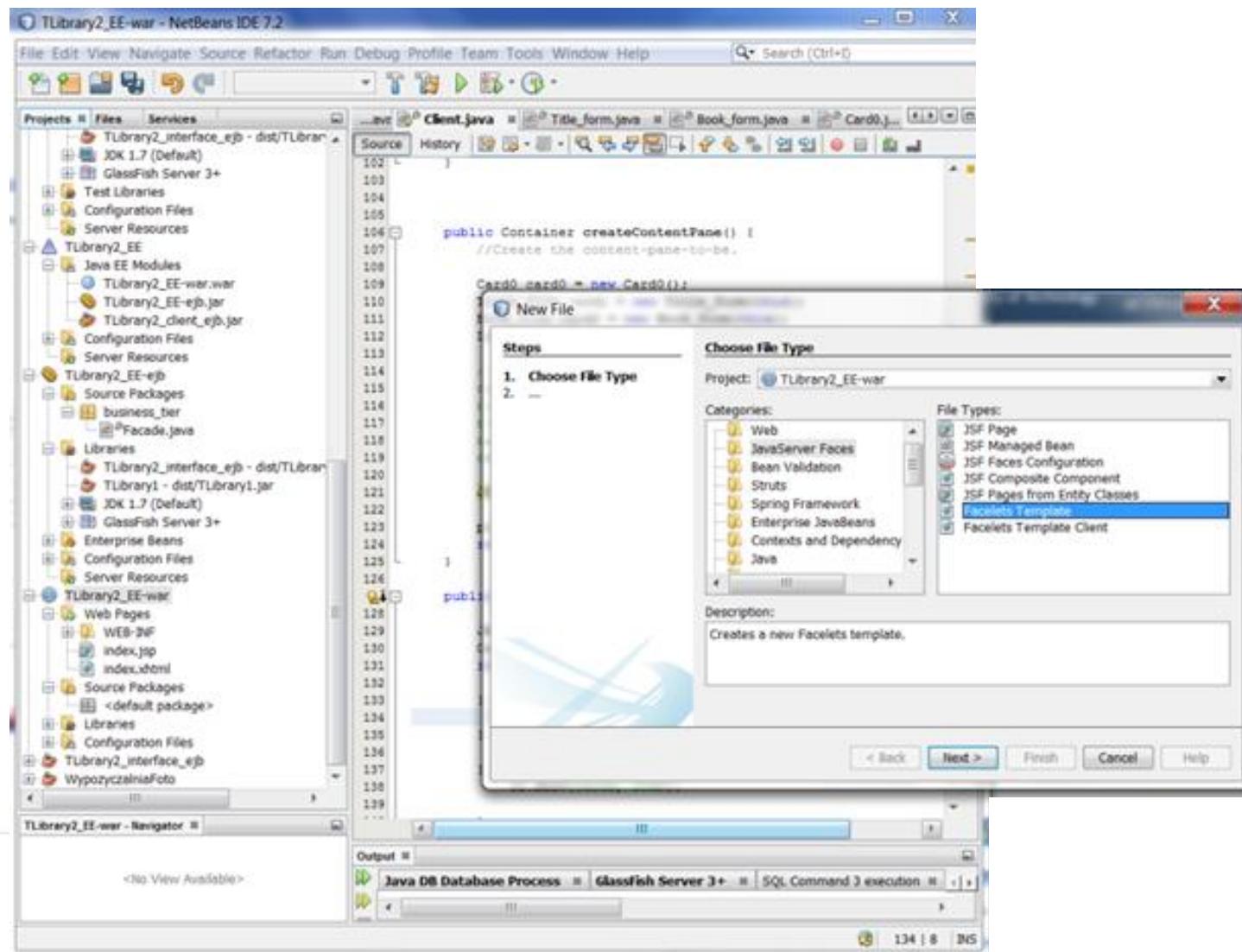
At left: Right click the name of project in the projects Tab, and choose the Properties item

Below: Choose the Framework item, click the Add button, and select the JavaServer Faces framework



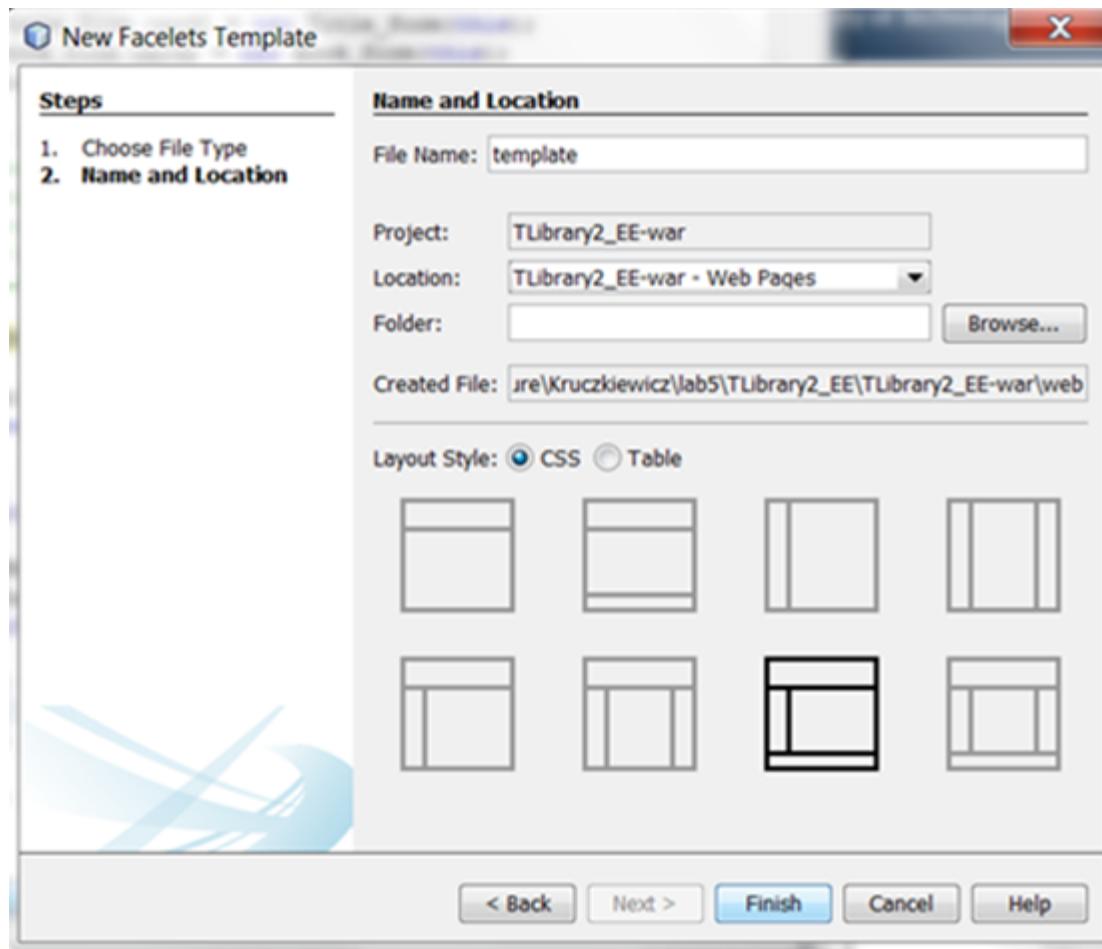


6.1. Addition the template of web pages: Right click the name of project in the Projects tab, choose the New item, then select JavaServer Faces File type, and the the Facelets Template file.





Below: choose the template and set the name of template file





TLibrary2_EE-war - NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Projects Files Services

TLibrary2_interface_ejb - dist/TLibrary2_EE-war.war TLibrary2_EE-ejb.jar TLibrary2_client_ejb.jar

JDK 1.7 (Default) GlassFish Server 3+ Test Libraries Configuration Files Server Resources

TLibrary2_EE Java EE Modules TLibrary2_EE-war.war TLibrary2_EE-ejb.jar TLibrary2_client_ejb.jar Configuration Files Server Resources

TLibrary2_EE-ejb Source Packages business_tier Facade.java

Libraries TLibrary2_interface_ejb - dist/TLibrary2_EE-war.war TLibrary1 - dist/TLibrary1.jar JDK 1.7 (Default) GlassFish Server 3+ Enterprise Beans Configuration Files Server Resources

TLibrary2_EE-war Web Pages WEB-INF resources index.jsp index.xhtml template.xhtml

template.xhtml - Navigator CSS HTML XHTML

Filters:

Output Java DB Database Process GlassFish Server 3+ SQL Command 3 execution SQL Command 4 execution

```
<h:outputStylesheet name="css/default.css" />
<h:outputStylesheet name="css/cssLayout.css"/>
<title>Library</title>
```

<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
 xmlns:ui="http://java.sun.com/jsf/facelets"
 xmlns:h="http://java.sun.com/jsf/html">

 <h:head>
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
 <link href=".//resources/css/default.css" rel="stylesheet" type="text/css" />
 <link href=".//resources/css/cssLayout.css" rel="stylesheet" type="text/css" />
 <title>Facelets Template</title>
 </h:head>

 <h:body>

 <div id="top">
 <ui:insert name="top">Top</ui:insert>
 </div>
 <div>
 <div id="left">
 <ui:insert name="left">Left</ui:insert>
 </div>
 <div id="content" class="left_content">
 <ui:insert name="content">Content</ui:insert>
 </div>
 <div id="bottom">
 <ui:insert name="bottom">Bottom</ui:insert>
 </div>
 </div>
 </h:body>

</html>

**6.2. Addition
the template
of web pages:**
Change the
the content of
attributes of
the **h:head** tag

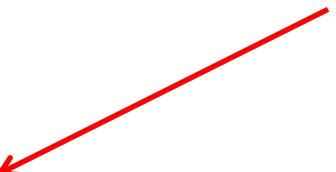


```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html">

  <h:head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <h:outputStylesheet name="css/default.css" />
    <h:outputStylesheet name="css/cssLayout.css"/>
    <title>Library</title>
  </h:head>

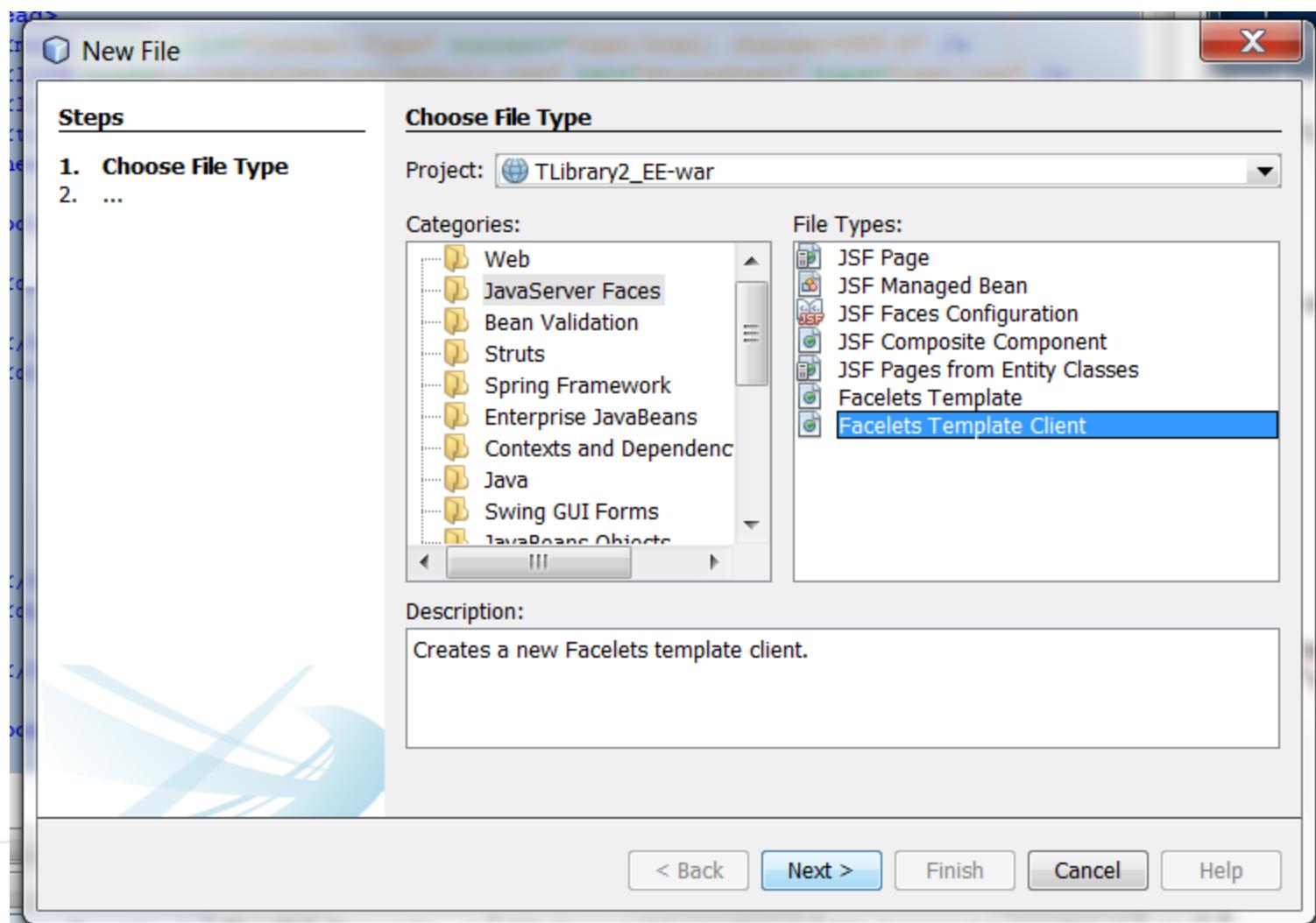
  <h:body>
    <div id="top">
      <ui:insert name="top">Top</ui:insert>
    </div>
    <div>
      <div id="left">
        <h:link outcome="/faces/presentation_tier_view/Store_data" value="Store data"/><br/>
        <h:link outcome="/faces/presentation_tier_view>Show_data" value="Show data"/>
      </div>
      <div id="content" class="left_content">
        <ui:insert name="content">Content</ui:insert>
      </div>
    </div>
    <div id="bottom">
      <ui:insert name="bottom">Bottom</ui:insert>
    </div>
  </h:body>
</html>
```

6.3. Addition the template of web pages: Insert the new code into the div tag with id equals left – this will be a menu of all web pages based on this template





6.4. Creation the main JSF page (index2), based on the previous defined template





New Facelets Template Client

Steps

1. Choose File Type
2. Name and Location

Name and Location

File Name: index2

Project: TLibrary2_EE-war

Folder:

Created File: ire\Kruczkiej\lab5\TLibrary2_EE\TLibrary2_EE-war\web\index2.xhtml

Template:

Generated Root Tag: <html>
 <ui:composition>

Select a template for which the client will be generated.

< Back Next > Finish Cancel Help

6.5. Creation the main JSF page (index2), based on the previous defined template – continuation

Below: choose the proper template

Browse Files

Folders:

- Web Pages
 - WEB-INF
 - presentation_tier_view
 - resources
 - index.xhtml
 - index.jsp
 - template.xhtml

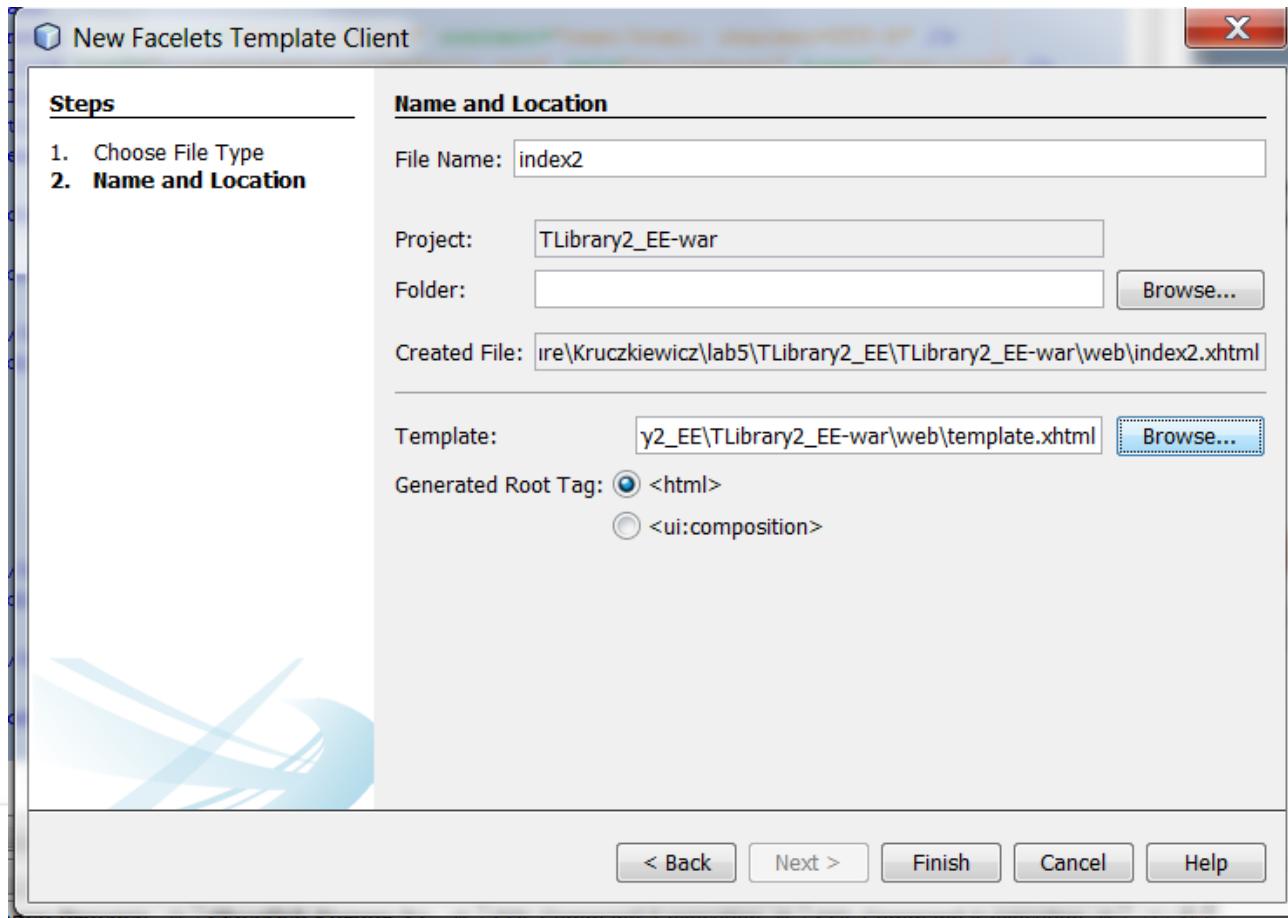
Select File Cancel





6.6. Creation the main JSF page (index2), based on the previous defined template – continuation

Below: after choosing the proper template





6.7. Creation the main JSF page (index2), based on the previous defined template – the result

At left: the view of the unused JSF and JSP main pages (index.jsp and index.xhtml);
At right: after deleting unused pages - the index2.xhtml is the main JSF page

NetBeans IDE 7.2

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

Search (Ctrl+I)

Projects Files Services

Loan_form.java template.xhtml index2.xhtml

Source History

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/1999/xhtml"
  xmlns:ui="http://java.sun.com/jsf/facelets">

<body>

  <ui:composition template="./template.xhtml">

    <ui:define name="top">
      top
    </ui:define>

    <ui:define name="content">
      content
    </ui:define>

    <ui:define name="bottom">
      bottom
    </ui:define>

  </ui:composition>

</body>
</html>
```

Output

Java DB Database Process GlassFish Server 3+ SQL Command 3 exec

1 | 1 INS

presentation_tier_view - Navigator

<No View Available>

NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run

Search (Ctrl+I)

Projects Files Services

TLibrary2_EE-ejb

Source Packages business tier Facade.java

Libraries TLibrary2_interface_ejb - dist/TLibrary1.jar TLibrary1 - dist/TLibrary1.jar JDK 1.7 (Default) GlassFish Server 3+ Enterprise Beans Configuration Files Server Resources

TLibrary2_EE-war

Web Pages WEB-INF resources index.jsp index.xhtml index2.xhtml template.xhtml

Source Packages <default package>

Libraries Configuration Files TLibrary2_interface_ejb WypożyczalniaFoto

index.jsp - Navigator

<No View Available>



6.8. The set up the main web page as the index2.xhtml – in the web.xml file (the descriptor of the web module).

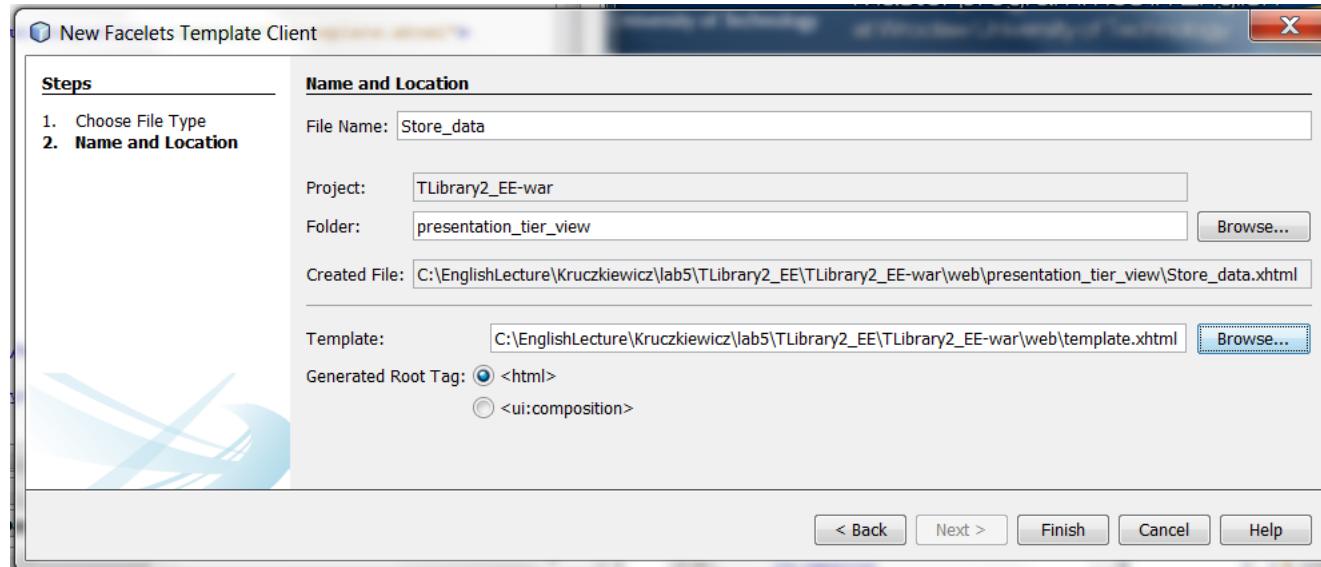
The screenshot shows the NetBeans IDE 7.2 interface with the project **TLibrary2_EE-war** open. The left pane displays the project structure under **Java EE Modules**, showing **TLibrary2_EE-war.war**, **TLibrary2_EE-ejb.jar**, and **TLibrary2_client_ejb.jar**. The **Web Pages** node contains **WEB-INF/web.xml**, **presentation_tier_view>Show_data.xhtml**, **Store_data.xhtml**, **resources/index2.xhtml**, and **template.xhtml**. The **Source Packages** node contains **Managed_Bean1.java**. The right pane shows the **web.xml** file content:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web">
    <context-param>
        <param-name>javax.faces.PROJECT_STAGE</param-name>
        <param-value>Development</param-value>
    </context-param>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>/faces/*</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>30</session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>faces/index2.xhtml</welcome-file>
    </welcome-file-list>
</web-app>
```

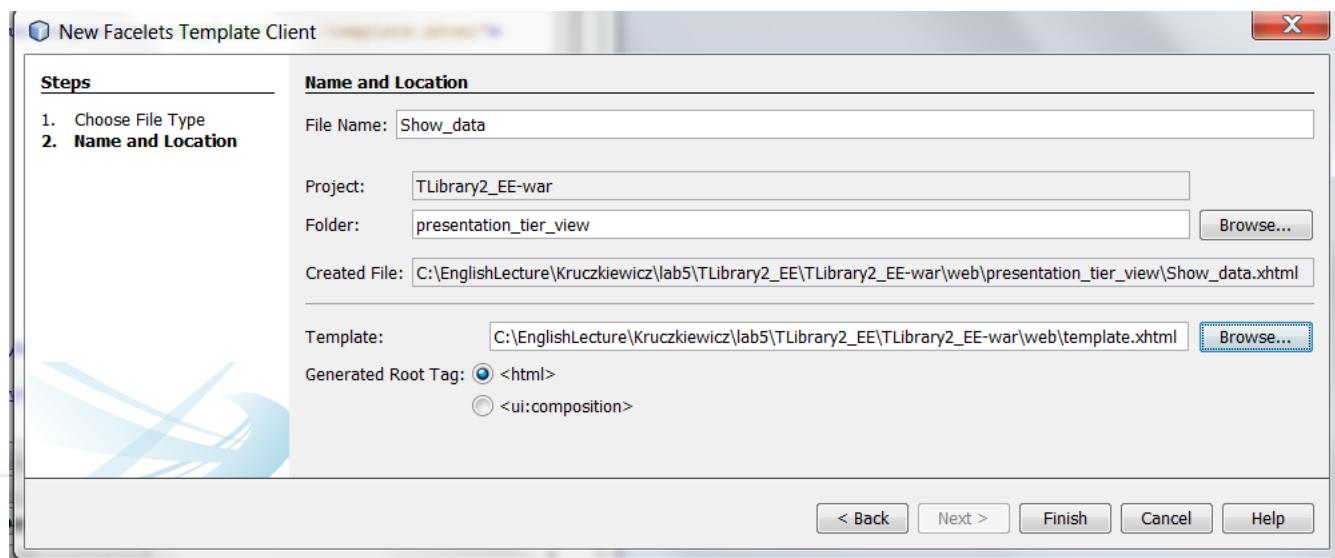
The **Output** pane at the bottom shows deployment logs for **GlassFish Server 3+**:

```
Java DB Database Process
GlassFish Server 3+
TLibrary2_client_ejb (run)

pre-run-deploy:
Distributing C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_client_ejb\dist\TLibrary2_client_ejb.jar
Initializing...
post-run-deploy:
```



6.9. Addition
the two JSF
pages, based
on the
previous
defined
template for
inserting
data in the
database and
displaying
the data
from
databases





6.10. The new JSF page for inserting data in the database – Store_data.xhtml

The screenshot shows the NetBeans IDE interface with the project **TLibrary2_EE-war** open. The **Store_data.xhtml** file is selected in the Source editor. The code for the page is as follows:

```
html body
1  <?xml version='1.0' encoding='UTF-8' ?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "
3  <html xmlns="http://www.w3.org/1999/xhtml"
4      xmlns:ui="http://java.sun.com/jsf/facelets"
5      xmlns:h="http://java.sun.com/jsf/html">
6      <body>
7
8          <ui:composition template="../../template.xhtml">
9
10         <ui:define name="content">
11             <h:form>
12                 <h:commandButton action="#{managed_Bean1.store_data}"
13                     value="Store data"/><br/>
14             </h:form>
15         </ui:define>
16     </ui:composition>
17
18     </body>
19 </html>
```

A red arrow points to the `<h:commandButton action="#{managed_Bean1.store_data}"` line, indicating the target of the button.

Tag for calling the store_data method from managed_Bean1 object (the Managed Bean type) – its definition is presented further part of this instruction



6.11. The definition of code of the Store_data.xhtml JSF page.

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:h="http://java.sun.com/jsf/html">

<body>

<ui:composition template=".../template.xhtml">

<ui:define name="content">
<h:form>
  <h:commandButton action="#{managed_Bean1.store_data}"
    value="Store data"/><br/>
</h:form>
</ui:define>
</ui:composition>
</body>
</html>
```



6.12. The code of the second JSF page for displaying data from the database , using the h: dataTable JSF component – the Show_data.xhtml page

TLibrary2_EE-war - NetBeans IDE 7.2

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

Search (Ctrl+I)

Projects Files Services

Source History

...ava template.xhtml index2.xhtml Store_data.xhtml Show_data.xhtml Managed_Bean1.java web.xml build-imp...

html body ui:composition ui:define h:form

1 <?xml version="1.0" encoding="UTF-8" ?>

2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"

3 <html xmlns="http://www.w3.org/1999/xhtml"

4 xmlns:ui="http://java.sun.com/jsf/facelets"

5 xmlns:h="http://java.sun.com/jsf/html"

6 xmlns:fn="http://java.sun.com/jsf/core">

7

8 <body>

9 <ui:composition template="../../template.xhtml">

10

11 <ui:define name="content">

12 <h:form styleClass="jsfcrud_list_form">

13 <h:panelGroup id="messagePanel" layout="block">

14 <h:messages errorStyle="color: red" infoStyle="color: green" layout="table"/>

15 </h:panelGroup>

16 <h:outputText escape="false" value="Lista produktów pusta" rendered="#{managed_Bean1.items.rowCount == 0}">

17 <h:panelGroup rendered="#{managed_Bean1.items.rowCount > 0}">

18 <h:dataTable value="#{managed_Bean1.items}" var="item" border="0"

19 cellpadding="2" cellspacing="0" rowClasses="jsfcrud_odd_row,jsfcrud_even_row"

20 rules="all" style="border:solid 1px">

21 <h:column>

22 <f:facet name="header">

23 <h:outputText value="Publisher"/>

24 </f:facet>

25 <h:outputText value="#{item.get(0)}"/>

26 </h:column>

27 <h:column>

28 <f:facet name="header">

29 <h:outputText value="ISBN"/>

30 </f:facet>

31 <h:outputText value="#{item.get(1)}"/>

32 </h:column>

Output

Java DB Database Process GlassFish Server 3+ TLibrary2_client_ejb (run)

PROJECT FINANCED FROM THE EU/EUROPEAN SOCIAL FUND

16 | 6 INS



6.13. The code of the second JSF page for displaying data from the database - continuation

TLibrary2_EE-war - NetBeans IDE 7.2

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

Search (Ctrl+I)

Projects Files Services

TLibrary2_EE-war

- Web Pages
 - WEB-INF
 - web.xml
 - presentation_tier_view
 - Show_data.xhtml
 - Store_data.xhtml
 - resources
 - index2.xhtml
 - template.xhtml
 - Source Packages
 - <default package>
 - Libraries
 - JDK 1.7 (Default)
 - GlassFish Server 3+
 - TLibrary2_interface_ejb

Store_data.xhtml - Navigator

html

Filters:

Source History

...tm index2.xhtml Store_data.xhtml Show_data.xhtml

```
html body ui:composition ui:define h:form h:panelGroup h:dataTable h:column
33         </h:column>
34     <h:column>
35         <f:facet name="header">
36             <h:outputText value="Title"/>
37         </f:facet>
38         <h:outputText value="#{item.get(2)}"/>
39     </h:column>
40     <h:column>
41         <f:facet name="header">
42             <h:outputText value="Author"/>
43         </f:facet>
44         <h:outputText value="#{item.get(3)}"/>
45     </h:column>
46     <h:column>
47         <f:facet name="header">
48             <h:outputText value="Actor"/>
49         </f:facet>
50         <h:outputText value="#{item.get(4)}"/>
51     </h:column>
52 </h:dataTable>
53 </h:panelGroup>
54 </h:form>
55 </ui:define>
56 </ui:composition>
57
58 </body>
59 </html>
```

Output

Java DB Database Process GlassFish Server 3+ SQL Command 3 execution SQL C

50 | 12 | INS



6.14. The code of the second JSF page for displaying data from the database - continuation

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core">

<body>
<ui:composition template=".../template.xhtml">

<ui:define name="content">
<h:form styleClass="jsfcrud_list_form">
<h:panelGroup id="messagePanel" layout="block">
<h:messages errorStyle="color: red" infoStyle="color: green" layout="table"/>
</h:panelGroup>
<h:outputText escape="false" value="Lista_produktow_pusta" rendered="#{managed_Bean1.items.rowCount == 0}"/>
<h:panelGroup rendered="#{managed_Bean1.items.rowCount > 0}">
<h:dataTable value="#{managed_Bean1.items}" var="item" border="0"
            cellpadding="2" cellspacing="0" rowClasses="jsfcrud_odd_row,jsfcrud_even_row"
            rules="all" style="border:solid 1px">
```



6.15. The code of the second JSF page for displaying data from the database - continuation

```
<h:column>
    <f:facet name="header">
        <h:outputText value="Publisher"/>
    </f:facet>
    <h:outputText value="#{item.get(0)}"/>
</h:column>
<h:column>
    <f:facet name="header">
        <h:outputText value="ISBN"/>
    </f:facet>
    <h:outputText value="#{item.get(1)}"/>
</h:column>
<h:column>
    <f:facet name="header">
        <h:outputText value="Title"/>
    </f:facet>
    <h:outputText value="#{item.get(2)}"/>
</h:column>
```





6.16. The code of the second JSF page for displaying data from the database - continuation

```
<h:column>
    <f:facet name="header">
        <h:outputText value="Author"/>
    </f:facet>
    <h:outputText value="#{item.get(3)}"/>
</h:column>
<h:column>
    <f:facet name="header">
        <h:outputText value="Actor"/>
    </f:facet>
    <h:outputText value="#{item.get(4)}"/>
</h:column>
</h:dataTable>
</h:panelGroup>
</h:form>
</ui:define>
</ui:composition>

</body>
</html>
```





TLibrary2_EE-war - NetBeans IDE 7.2

File Edi View Navig Sourc Refact Rur Debu Profil Tear Tool Windc Help Search (Ctrl+I)

Projects Files Services

TBook.java
TBook_period.java
TTtitle_book.java
TTtitle_book_on_tape.java

Generated Sources (ap-source-output)

Libraries

TLibrary TLibrary TLibrary

Source Libraries Enterprise Configuration Servers

TLibrary Web Sources Libraries Enterprise Configuration Servers

Test RESTful Web Services Test Alt+F6

Open Required Projects Close

Rename...

New

Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services Test Alt+F6

JSF Managed Bean...
Facelets Template Client...
Facelets Template...
Java Class...
Session Bean...
Persistence Unit...
JSF Faces Configuration...
JSF Page...
JSF Pages from Entity Classes...
Java Package...
JDialog Form...
Session Beans For Entity Classes...
GlassFish Descriptor...
Entity Classes from Database...
Session Beans For Entity Classes...
Other...

6.17. Creation of the Managed Bean type of object as the Presentation Tier object based on JSF technology – Choose File Type

e

TLibrary2_EE-war

File Types:

JSF Page
JSF Managed Bean
JSF Faces Configuration
JSF Composite Component
JSF Pages from Entity Classes
Facelets Template
Facelets Template Client

managed bean class.

Back Next > Finish Cancel Help



New File

Steps

1. Choose File Type
2. ...

Choose File Type

Project: TLibrary2_EE-war

Categories:

- Web
- JavaServer Faces
- Bean Validation
- Struts
- Spring Framework
- Enterprise JavaBeans
- Contexts and Depend
- Java

File Types:

- JSF Page
- JSF Managed Bean
- JSF Faces Configuration
- JSF Composite Component
- JSF Pages from Entity Classes
- Facelets Template
- Facelets Template Client

New JSF Managed Bean

Steps

1. Choose File Type
2. Name and Location

Name and Location

Class Name: Managed_Bean1

Project: TLibrary2_EE-war

Location: Source Packages

Package: presentation_tier

Created File: 2_EE\TLibrary2_EE-war\src\java\presentation_tier\Managed_Bean1.java

Add data to configuration file

Configuration File:

Name: managed_Bean1

Scope: request

Bean Description:

< Back Next > Finish Cancel Help

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!

6.18. Creation of the Managed Bean type of object as the Presentation Tier object based on JSF technology - continuation



6.19. Creation of the Managed Bean type of object as the Presentation Tier object based on JSF technology - the generated code of this class

TLibrary2_EE-war - NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Search (Ctrl+I)

Projects Files Services

JDK 1.7 (Default)
GlassFish Server 3+
Enterprise Beans
Configuration Files
Server Resources
TLibrary2_EE-war
Web Pages
WEB-INF
presentation tier view
Show_data.xhtml
Store_data.xhtml
resources
index2.xhtml
template.xhtml
Source Packages
presentation tier
Managed_Bean1.java
Libraries
TLibrary2_EE-ejb - dist/TLibrary2_EE-
JDK 1.7 (Default)
GlassFish Server 3+
Configuration Files

Managed_Bean1.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package presentation_tier;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;

@ManagedBean
@RequestScoped
public class Managed_Bean1 {

    public Managed_Bean1() {
    }
}
```

Managed_Bean1 - Navigator

Members View

Managed_Bean1
Managed_Bean1()

Output

Java DB Database Process GlassFish Server 3+ SQL

9 | 1 INS



6.20. Creation the connection with the Facade EJB from the TLibrary2_EE-ejb module by using the mechanism of Insert code (right click the code in the editor window, select the the **Insert code** item, and then select the **Call Enterprise Bean**, and at last select the Facade component from the list).

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

- NetBeans IDE 7.2**: The main application window.
- Projects**: A tree view showing the project structure. The **TLibrary2_EE-war** module is expanded, showing **Web Pages**, **WEB-INF**, and **presentation_tier**. **Managed_Bean1.java** is selected in the presentation_tier folder.
- Managed_Bean1.java - Editor**: The code editor window showing Java code for a Managed Bean. A context menu is open over the code, with the **Insert Code...** option highlighted.
- Generate**: A context menu for the code editor, listing options like Constructor..., Logger..., and Call Enterprise Bean... (which is also highlighted).
- Call Enterprise Bean**: A dialog box titled "Call Enterprise Bean" with the sub-instruction "Select an enterprise bean from open projects." It lists several projects and modules:
 - AsyncRequest
 - Szyfrowanie
 - ServletStateless
 - Proba
 - Portal_4
 - TLibrary2_EE-war
 - TLibrary2_EE-ejbThe **Facade** module under **TLibrary2_EE-ejb** is selected.
- Output**: A panel showing build logs and database connections.
- Members View**: A panel showing the members of the current class.



6.21. The code of the Managed_Bean1 class with added reference of the Facade EJB by using the annotation and injecting mechanism.

The screenshot shows the NetBeans IDE 7.2 interface with the following details:

- Title Bar:** TLibrary2_EE-war - NetBeans IDE 7.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Tools Window Help
- Toolbar:** Standard NetBeans toolbar with icons for file operations, search, and project navigation.
- Projects Tab:** Shows the project structure:
 - Server Resources
 - TLibrary2_EE-war
 - Web Pages
 - WEB-INF
 - web.xml
 - presentation_tier_view
 - Show_data.xhtml
 - Store_data.xhtml
 - resources
 - index2.xhtml
 - template.xhtml
 - Source Packages
 - presentation_tier
 - Managed_Bean1.java
 - Libraries
 - TLibrary2_EE-ejb - dist/TLibrary2_EE-ejb.jar
 - TLibrary2_interface_ejb - dist/TLibrary2_interface_ejb.jar
 - JDK 1.7 (Default)
 - GlassFish Server 3+
 - Enterprise Beans
 - Facade
 - Configuration Files
 - MANIFEST.MF
 - web.xml
- Managed_Bean1.java Editor:** The code is displayed in the main editor window. It includes annotations for Managed Bean and RequestScoped, and an injection point for a FacadeRemote interface.
- Navigator:** Shows the members of the Managed_Bean1 class, including the facade field and its constructor.
- Output:** Shows Java DB Database Process and GlassFish Server 3+ logs.
- Status Bar:** Displays the page number (15 | 5) and an INS indicator.



6.22. The code of the Managed_Bean1 class

```
package presentation_tier;

import business_tier.FacadeRemote;
import javax.ejb.EJB;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.faces.model.DataModel;
import javax.faces.model.ListDataModel;

@ManagedBean
@RequestScoped
public class Managed_Bean1 {
    @EJB
    private FacadeRemote facade;
    private DataModel items;

    public Managed_Bean1() {
    }

    public FacadeRemote getFacade() {
        return facade;
    }

    public void setFacade(FacadeRemote facade) {
        this.facade = facade;
    }
}
```

The model of the dataTable component used on the Show_data.xhtml JSF page for displaying the data from the database



6.23. The code of the Managed_Bean1 class - continuation

```
public String store_data() {  
    try {  
        facade.add_titles();  
        facade.add_books();  
    } catch (Exception e) {  
    }  
    return "/faces/index2";  
}
```

The **store_data** method for handling event of the h:commandButton component, used by the Store_data.xhtml page. This method calls two methods from the Facade EJB, which inserting data into the database by using ORM (JPA 2.0 controllers from TLibrary1 project). As the response, it returns to the main index2.xhtml JSF page (p. 4-7, 24)

```
public DataModel create_DataModel() {  
    try{  
        return new ListDataModel( facade.titles());  
    }  
    catch(Exception e)  
    {  
        System.out.println("Blad");  
        return null;  
    }  
}
```

The **create_DataModel** method for creation the new data model for h:DataTable component – this based on data returned from the titles method of the Facade EJB and they are read from database by using ORM mechanism (p. 24, 4-7)



6.24. The code of the Managed_Bean1 class - continuation

```
public DataModel getItems() {  
    if (items == null) {  
        System.out.println("Model");  
        items = create_DataModel();  
    }  
    return items;  
}
```

```
public void setItems(DataModel items) {  
    this.items = items;  
}
```



The getItems method for creation the new data model for h:DataTable component by using binding mechanism – this based on data returned from the create_DataModel (at the previous slide).





6.25. The code of the Managed_Bean1 class – the result

The screenshot shows the NetBeans IDE 7.2 interface with the project **TLibrary2_EE-war** open. The left pane displays the project structure under the **Projects** tab, showing modules like **TLibrary2_EE-ejb**, **TLibrary2_EE-war**, and their respective source packages and libraries. The right pane shows the **Managed_Bean1.java** file in the **Source** editor. The code implements a managed bean with methods for showing data and creating a data model.

```
39     }
40
41     public String show_data() {
42         create_DataModel();
43         return "/faces/presentation_tier_view>Show_da
44     }
45
46     public DataModel create_DataModel() {
47         try{
48
49             return new ListDataModel( facade.titles());
50
51         catch(Exception e)
52         {
53             System.out.println("Blad");
54             return null;
55         }
56     }
57
58     public DataModel getItems() {
59         if (items == null) {
60             System.out.println("Model");
61             items = create_DataModel();
62         }
63         return items;
64     }
65
66     public void setItems(DataModel items) {
67         this.items = items;
68     }
69 }
```

The **Output** window at the bottom is empty. The status bar at the bottom right shows the page number **54 | 26 | INS**.



7.0. Running the program

1. After this development, your program will be executed properly, if you **clean and build** the following programs:
 1. TLibrary1
 2. TLibrary2_interface_ejb
 3. TLibrary2_EE-ejb
 4. TLibrary2_client_ejb
 5. TLibrary2_EE-war
2. Then you must **deploy** the TLibrary2_EE program.
3. Finally, you may **run** a few instances of TLibrary2_client_ejb programs.
4. At last, insert the following url address: http://localhost:8080/TLibrary2_EE-war/ in any browser and run a few web clients.
5. These programs (p.3 i 4) share the common data as titles and books.
6. In the **Service Tab** you may see, if your EE program deploy properly (Server item). The other useful information you may get from the **Glasfish output window tab**.



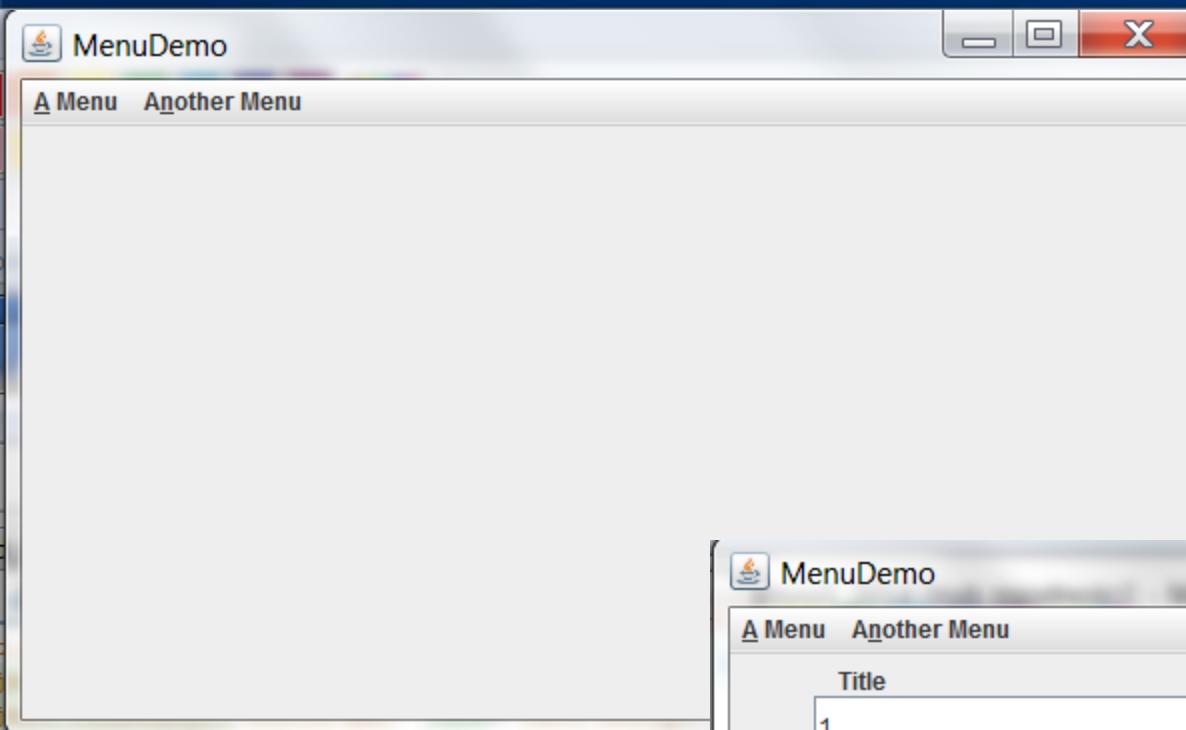


7.1. The view of the main web page (rendered by using the index2.xhtml JSF page) of the **TLibrary2_EE-war project**

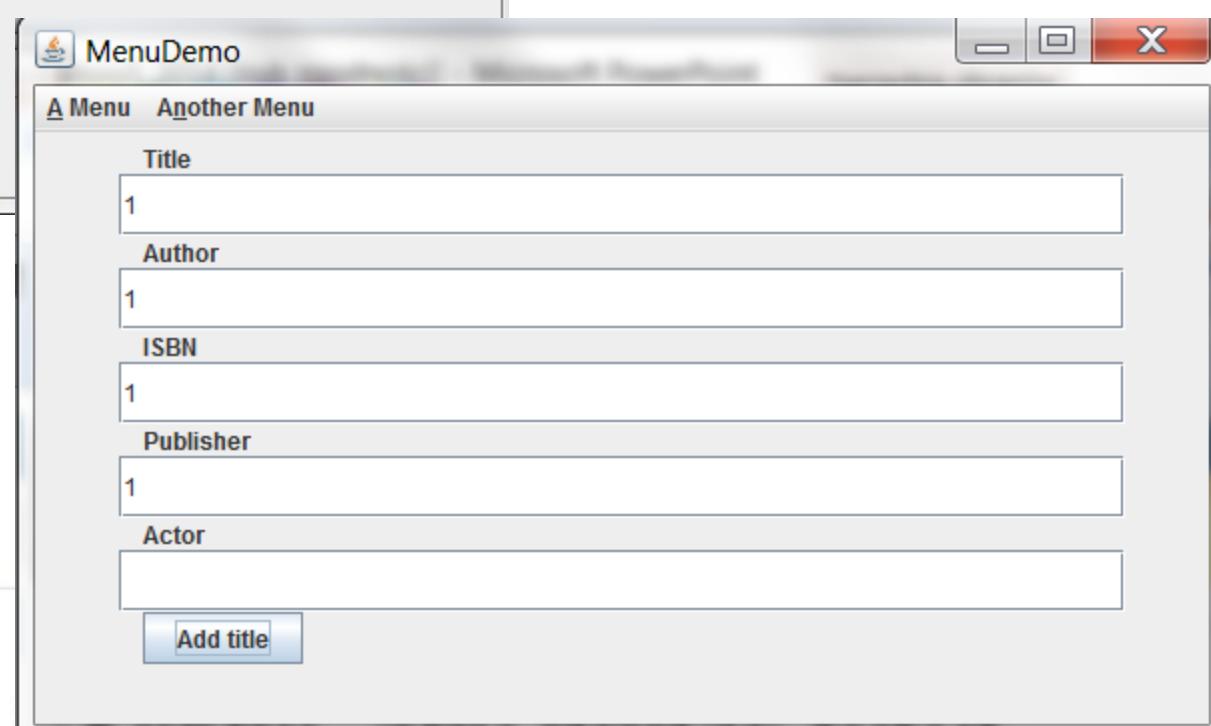
The screenshot shows a Mozilla Firefox browser window with the title "Library - Mozilla Firefox". The address bar displays "localhost:8080/TLibrary2_EE-war/". The page content is a JSF rendered view with the following structure:

- A blue header bar with the word "top".
- A yellow sidebar on the left containing "Store data" and "Show data" buttons.
- A grey content area labeled "content".
- A light blue footer area labeled "bottom".

The browser interface includes standard navigation buttons (back, forward, search, etc.) and a toolbar with icons for file operations.



7.2. The view of the Enterprise Application Client (**TLibrary2_client_ejb**) for processing of application data - with the same responsibilities as of the version of third laboratory.





7.3. The view after choose the **Store data** link (the left part of page) of the
TLibrary2_EE-war client

The screenshot shows a Mozilla Firefox browser window. The title bar reads "Library - Mozilla Firefox". The menu bar includes "Plik", "Edycja", "Widok", "Historia", "Zakładki", "Narzędzia", and "Pomoc". Below the menu is a toolbar with a "Library" icon, a close button ("x"), and an add button ("+"). The address bar shows the URL "localhost:8080/TLibrary2_EE-war/faces/presentation_tier_view/Store_data.xhtml". The browser's navigation bar includes back, forward, search, and other standard icons. The main content area displays a web page titled "Top". On the left, there are two buttons: "Store data" (highlighted in yellow) and "Show data". A "Bottom" section follows. The overall interface is a Java-based application within a browser.





7.4. The view after choosing the **Store data** button (the right part of page) – i.e. after inserting the application data into the database (p.7.2) by the **TLibrary2_EE-war** client

The screenshot shows a Mozilla Firefox window with the title "Library - Mozilla Firefox". The address bar displays "localhost:8080/TLibrary2_EE-war/faces/presentation_tier_view/Store_data.xhtml". The page content is organized into three horizontal sections: "top", "content", and "bottom". The "content" section contains two buttons: "Store data" (highlighted in yellow) and "Show data". The "bottom" section is currently empty.





7.5. The view after choosing the Show data button (the left part of page) – i.e. after getting data from the database by the **TLibrary2_EE-war** client

Library - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Library x +

Wprowadź adres lub szukaj Google

Często odwiedzane Pierwsze kroki Galeria obiektów Web ... Sugerowane witryny

Site F

Top

Store data Show data

Publisher	ISBN	Title	Author	Actor
1	1	1	1	

Bottom





7.6. The generated tables by using the ORM mechanism, accordingly to the annotation placed in the Entity classes (the instruction of the fourth laboratory)

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

- Projects Tab:** Shows a Java DB project named "LIBRARY1". A red box highlights the "Tables" section under "LIBRARY1".
- Code Editor:** Displays Java code for managing entities. The code uses EntityManager and Transaction to interact with the database.
- Output Window:** Shows the build process for "TLibrary2_client_ejb (run)". It includes messages about creating directories, compiling source files, and building a jar file.
- Toolbar:** Standard NetBeans toolbar with icons for file operations, search, and run.
- MenuBar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Search Bar:** Search (Ctrl+I) at the top right.

```
EntityManager em = getEntityManager();
try {
    em.getTransaction().begin();
    TTITLE_book Title_bookx =
        em.find(TTITLE_book.class, title_book.getId());
    em.remove(Title_bookx);
    em.getTransaction().commit();
} finally {
    em.close();
    return false;
}

public boolean updateTTITLE_book(TTITLE_book title_book) {
    EntityManager em = getEntityManager();
    try {
        em.getTransaction().begin();
        TTITLE_book Title_bookx =
            em.find(TTITLE_book.class, title_book.getId());
        Title_bookx.setTitle(title_book.getTitle());
        Title_bookx.setAuthor(title_book.getAuthor());
        Title_bookx.setISBN(title_book.getISBN());
        Title_bookx.setPublisher(title_book.getPublisher());
        em.getTransaction().commit();
    }
}
```

Output window logs:

- Created dir: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_interface_ejb\build\classes
- Created dir: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_interface_ejb\build\empty
- Created dir: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_interface_ejb\build\generated-
- Compiling 1 source file to C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_interface_ejb\bu
- TLibrary2_interface_ejb.compile:
- Building jar: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_interface_ejb\dist\TLibrary2_
- TLibrary2_interface_ejb.jar:
- deps-jar:



7.7. The data stored in the database by Store data web page- at previous time they have been inserted by the Enterprise Application Client program (**TLibrary2_client_ejb**) as the application data (p.7.2)

The screenshot shows the NetBeans IDE 7.2 interface. On the left, the Projects panel displays a Java DB connection named 'library1' containing tables like SEQUENCE, TBOOK, and TTITLE_BOOK. The Database browser panel shows the TTITLE_BOOK table with one row of data. The Output window at the bottom shows the execution results of a SQL query.

NetBeans IDE 7.2

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

Search (Ctrl+I)

Projects Files Services

Databases Java DB Drivers

jdbc:derby://localhost:1527/BibliotekaD []

jdbc:derby://localhost:1527/HotelBaza []

library1 [lib]

APP LIBRARY1

Tables

SEQUENCE

SEQ_NAME

SEQ_COUNT

Indexes

Foreign Keys

TBOOK

ID

DTYPE

NUMBER

MTITLE_BOOK_ID

PERIOD

Indexes

Foreign Keys

TTITLE_BOOK

ID

DTYPE

ISBN

AUTHOR

PUBLISHER

TITLE

ACTOR

Indexes

Foreign Keys

Views

Procedures

NULLID

Source History Connection: jdbc:derby://localhost:1527/library1... select * from LIBRARY1.TTITLE_BOOK

select * from LIBRARY1.TTITLE_BOOK

Total Rows: 1 Page: 1 of 1 Matching Rows: 1

#	ID	DTYPE	ISBN	AUTHOR	PUBLISHER	TITLE	ACTOR
1		1 TTtitle_book	1	1	1	1	<NULL>

Output

Java DB Database Process GlassFish Server 3+ TLibrary2_client_ejb (run) SQL Command 5 execution

Executed successfully in 0,01 s.
Line 1, column 1

Execution finished after 0,01 s, 0 error(s) occurred.

TLibrary2_client_ejb (run) running... 1 | 1 INS



7.8. The **TLibrary2_client_ejb** form to adding the new books of the selected title, as the application data.

MenuDemo

A Menu Another Menu

Publisher	ISBN	Title	Author	Actor
1	1	1	1	
1	1	1	1	1

Number of a book
11

Period of a book

Add book

Books

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 111 Period: Fri Mar 28 03:50:50 CET 2014

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 111 Period: Fri Mar 28 03:50:50 CET 2014

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 11





7.9. The view after choose the **Store data** link (the left part of page) of the
TLibrary2_EE-war client

The screenshot shows a Mozilla Firefox browser window. The title bar reads "Library - Mozilla Firefox". The menu bar includes "Plik", "Edycja", "Widok", "Historia", "Zakładki", "Narzędzia", and "Pomoc". The toolbar includes icons for back, forward, search, and other functions. The address bar shows the URL "localhost:8080/TLibrary2_EE-war/faces/presentation_tier_vie...". Below the address bar are links for "Często odwiedzane", "Pierwsze kroki", "Galeria obiektów Web ...", and "Sugerowane witryny". The main content area has a blue header bar with the word "Top". Below it, there are two buttons: "Store data" (highlighted in yellow) and "Show data". A light blue bar labeled "Bottom" is at the bottom of the content area.



7.10. The view after choosing the **Store data** button (the right part of page) – i.e. after inserting the application data into the database (p. 7.8) by the **TLibrary2_EE-war** client

The screenshot shows a Mozilla Firefox window with the title "Library - Mozilla Firefox". The address bar displays "localhost:8080/TLibrary2_EE-war/faces/presentation_tier_view/Store_data.xhtml". The main content area is divided into three horizontal sections: "top", "content", and "bottom". The "content" section contains a yellow sidebar with two buttons: "Store data" and "Show data". The "Store data" button is highlighted with a yellow background. The "content" section itself is currently empty.





7.11. The view after choosing the Show data button (the left part of page) – i.e. after getting data from the database by the **TLibrary2_EE-war** client

Library - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Library

localhost:8080/TLibrary2_EE-war/faces/presentation_tier_view>Show_data.xhtml

Często odwiedzane Pierwsze kroki Galeria obiektów Web ... Sugerowane witryny

Top

Store data Show data

Publisher	ISBN	Title	Author	Actor
1	1	1	1	
1	1	1	1	1

Bottom





7.12. The data stored in the database by the Store data web page - at previous time they have been inserted by the Enterprise Application Client program (**TLibrary2_client_ejb**) as the application data (p.7.8) - the view of auxiliary sequence table (to support the AUTO mechanism of generating the keys of persisted data during ORM mechanism)

The screenshot shows the NetBeans IDE 7.2 interface. On the left, the Database browser displays a tree structure of databases, tables, and their columns. The 'LIBRARY1' database is selected, showing tables like SEQUENCE, TBOOK, and TTITLE_BOOK. The SEQUENCE table has columns SEQ_NAME and SEQ_COUNT, with a value of 50. The TBOOK table has columns ID, DTYPY, NUMBER, MTITLE_BOOK_ID, PERIOD, and others. The TTITLE_BOOK table has columns ID, DTYPY, ISBN, AUTHOR, PUBLISHER, TITLE, ACTOR, and others. The bottom pane shows the output of a SQL query: 'select * from LIBRARY1.SEQUENCE'. The results grid shows one row: SEQ_NAME is SEQ_GEN and SEQ_COUNT is 50. The output window below shows the execution details: 'Executed successfully in 0,01 s.' and 'Line 1, column 1'. The status bar at the bottom indicates 'TLibrary2_client_ejb (run)' and 'running...'.

#	SEQ_NAME	SEQ_COUNT
1	SEQ_GEN	50

Java DB Database Process ■ GlassFish Server 3+ ■ TLibrary2_client_ejb (run) ■ SQL Command 5 execution ■ SQL
Executed successfully in 0,01 s.
Line 1, column 1
Execution finished after 0,01 s, 0 error(s) occurred.

TLibrary2_client_ejb (run) | running... | 1 | 1 | INS



7.13. The data stored in the database (titles) by Store data web page- at previous time they have been inserted by the Enterprise Application Client program (**TLibrary2_client_ejb**) as the application data (p. 7.8)

The screenshot shows the NetBeans IDE 7.2 interface. On the left, the Database browser displays a tree structure of databases, drivers, and tables. The 'LIBRARY1' database is selected, showing its tables: SEQUENCE, TBOOK, TTITLE_BOOK, and others. The TTITLE_BOOK table is expanded, showing columns: ID, DTYPY, NUMBER, MTITLE_BOOK_ID, PERIOD, ISBN, AUTHOR, PUBLISHER, TITLE, and ACTOR. In the center, a SQL editor window contains the query: 'select * from LIBRARY1.TTITLE_BOOK'. Below it, a results grid shows two rows of data:

#	DTYPY	ISBN	AUTHOR	PUBLISHER	TITLE	ACTOR
1	1 TTTitle_book	1	1	1	1	<NULL>
2	2 TTTitle_book_on_tape	1	1	1	1	1

At the bottom, the Output window shows the execution results:

```
Execution finished after 0 s, 0 error(s) occurred.
```



7.14. The data stored in the database (books) by Store data web page- at previous time they have been inserted by the Enterprise Application Client program (**TLibrary2_client_ejb**) as the application data (p. 7.8)

The screenshot shows the NetBeans IDE 7.2 interface. On the left, the Database browser displays a tree structure of databases, drivers, and tables. Under the 'LIBRARY1' database, the 'TBOOK' table is selected, showing its columns: ID, DTYPY, NUMBER, MTITLE_BOOK_ID, and PERIOD. Below the table, the 'TTITLE_BOOK' table is also listed with its columns: ID, DTYPY, ISBN, AUTHOR, PUBLISHER, TITLE, ACTOR, and INDEXES. The central pane shows an SQL query: 'select * from LIBRARY1.TBOOK'. The results pane displays the following data:

#	ID	DTYPY	NUMBER	MTITLE_BOOK_ID	PERIOD
1		3 TBook		1	1 <NULL>
2		6 TBook		11	2 <NULL>
3		4 TBook_period		11	1 2014-03-28
4		5 TBook_period		111	2 2014-03-28

The bottom pane, 'Output', shows the execution results:

```
ejb (run) * | SQL Command 5 execution * | SQL Command 6 execution * | SQL Command 7 execution * | SQL Com
Executed successfully in 0 s.
Line 1, column 1

Execution finished after 0 s, 0 error(s) occurred.
```



7.15. The close of Enterprise application – after undeploy process

The screenshot shows the NetBeans IDE 7.2 interface. A red arrow points from the text "The close of Enterprise application – after undeploy process" to the "Undeploy" option in the context menu of the "TLibrary" application under the "GlassFish Server 3+" node in the Projects tab.

Projects Tab:

- Nodes: TITLE, ACTOR, Indexes, Foreign Keys, Views, Procedures, NULLID, SQL, SYS, SYSCAT, SYSCS_DIAG, SYCS_UTIL, SYSFUN, SYSIBM, SYSPROC, SYSSTAT.
- Connections: jdbc:derby://localhost:1527/Library_OR, jdbc:derby://localhost:1527/Proba7 [Pro], jdbc:derby://localhost:1527/sample [ap], jdbc:derby://localhost:1527/Serwis [Se], jdbc:derby://localhost:1527/Serwis1 [S], jdbc:derby://localhost:1527/Transport [T].
- Servers: Apache Tomcat 7.0.27.0, GlassFish Server 3+ (selected), Web Services.
- GlassFish Server 3+ Node:
 - Applications: TLibrary (selected).
 - Actions: Undeploy (highlighted with a red arrow), Enable, Disable.
- Maven Repository, Cloud, Hudson Builders.

Central Area:

- SQL Command 5: select * from LIBRARY1.TBOOK
- SQL Command 6: select * from LIBRARY1.TB...
- Output Tab:
 - (run) SQL Command 5 execution SQL Command 6 execution SQL Command 7 execution SQL Command 8 execution
 - Executed successfully in 0 s.
 - Line 1, column 1
 - Execution finished after 0 s, 0 error(s) occurred.



7.16. Restored data from database, after again opening EE application with two kinds of clients: **TLibrary2_client_ejb** as the Enterprise Application client (below) and **TLibrary2_EE-war** as the web client

MenuDemo

A Menu Another Menu

Publisher	ISBN	Title	Author	Actor
1	1	1	1	
1	1	1	1	1

Numeber of a book

Period of a book

Add book

Books

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 11

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 11

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 111 Period: Fri Mar 28 00:00:00 CET 2014



7.17. Restored data from database, after again opening EE application with two kinds of clients: **TLibrary2_client_ejb** as the Enterprise Application client and the web client (below) as the **TLibrary2_EE-war** project.

Library - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Library x +

localhost:8080/TLibrary2_EE-war/faces/presentation_tier_view>Show_data.xhtml

Często odwiedzane Pierwsze kroki Galeria obiektów Web ... Sugerowane witryny

Top

Store data Show data

Publisher	ISBN	Title	Author	Actor
1	1	1	1	
1	1	1	1	1

Bottom

