Projekt aplikacji wielowarstowej typu Java EE 6.0

Autor: Zofia Kruczkiewicz

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 ResourcesTiers

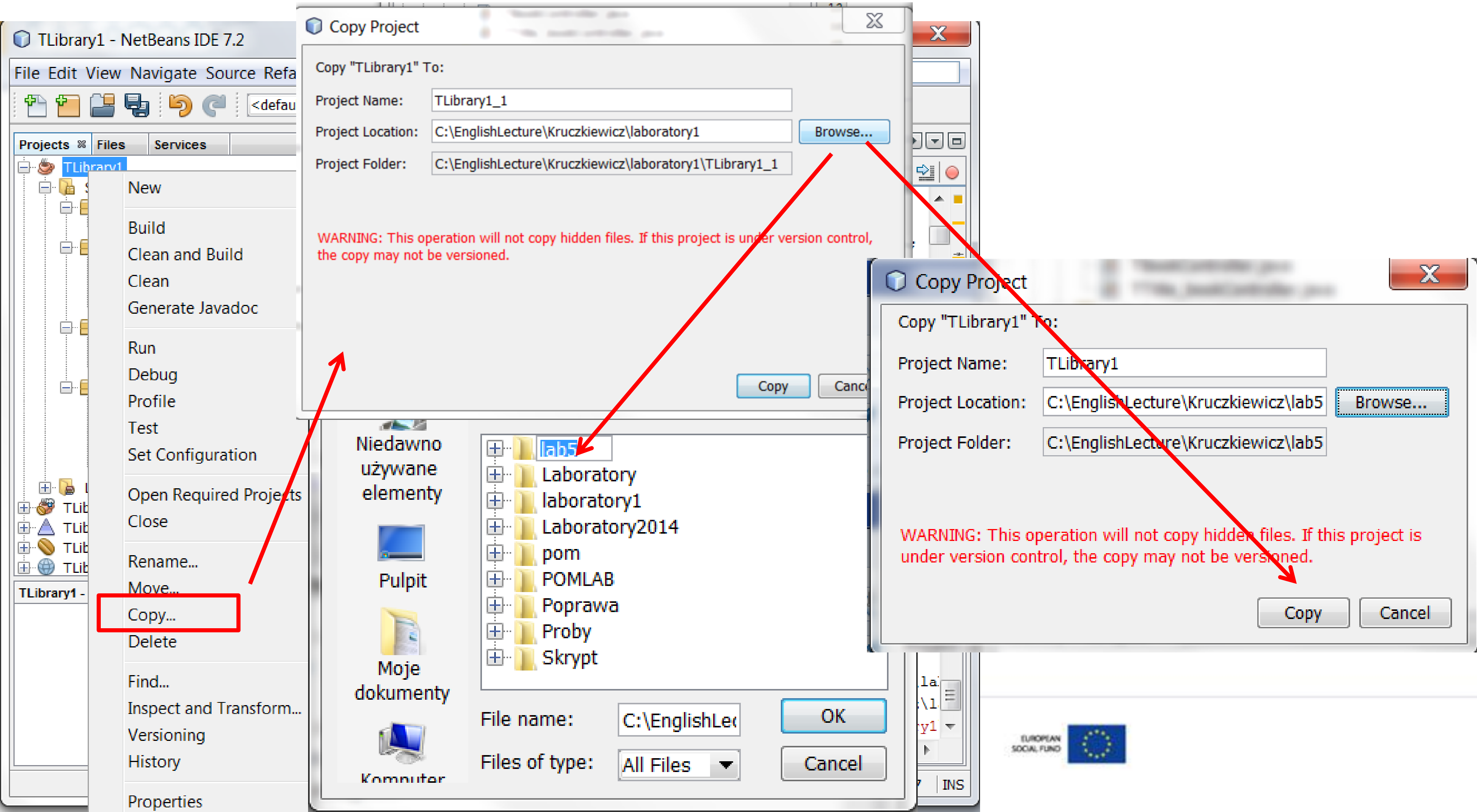
D.Alur, J.Crupi, D. Malks, Core J2EE. Desin Patterns

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



TLibrary1 - NetBeans IDE 7.2

Copy Project

Copy "TLibrary1" To:

Project Name: TLibrary1_1

Project Location: C:\EnglishLecture\Kruczkiewicz\laboratory1

Project Folder: C:\EnglishLecture\Kruczkiewicz\laboratory1\TLibrary1_1

WARNING: This operation will not copy hidden files. If this project is under version control, the copy may not be versioned.

Copy Project

Copy "TLibrary1" To:

Project Name: TLibrary1

Project Location: C:\EnglishLecture\Kruczkiewicz\lab5

Project Folder: C:\EnglishLecture\Kruczkiewicz\lab5

WARNING: This operation will not copy hidden files. If this project is under version control, the copy may not be versioned.

File name: C:\EnglishLec

Files of type: All Files

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) {
      }
  }
```

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

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

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

```
 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.getTTitle_books_();
}

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

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

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

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

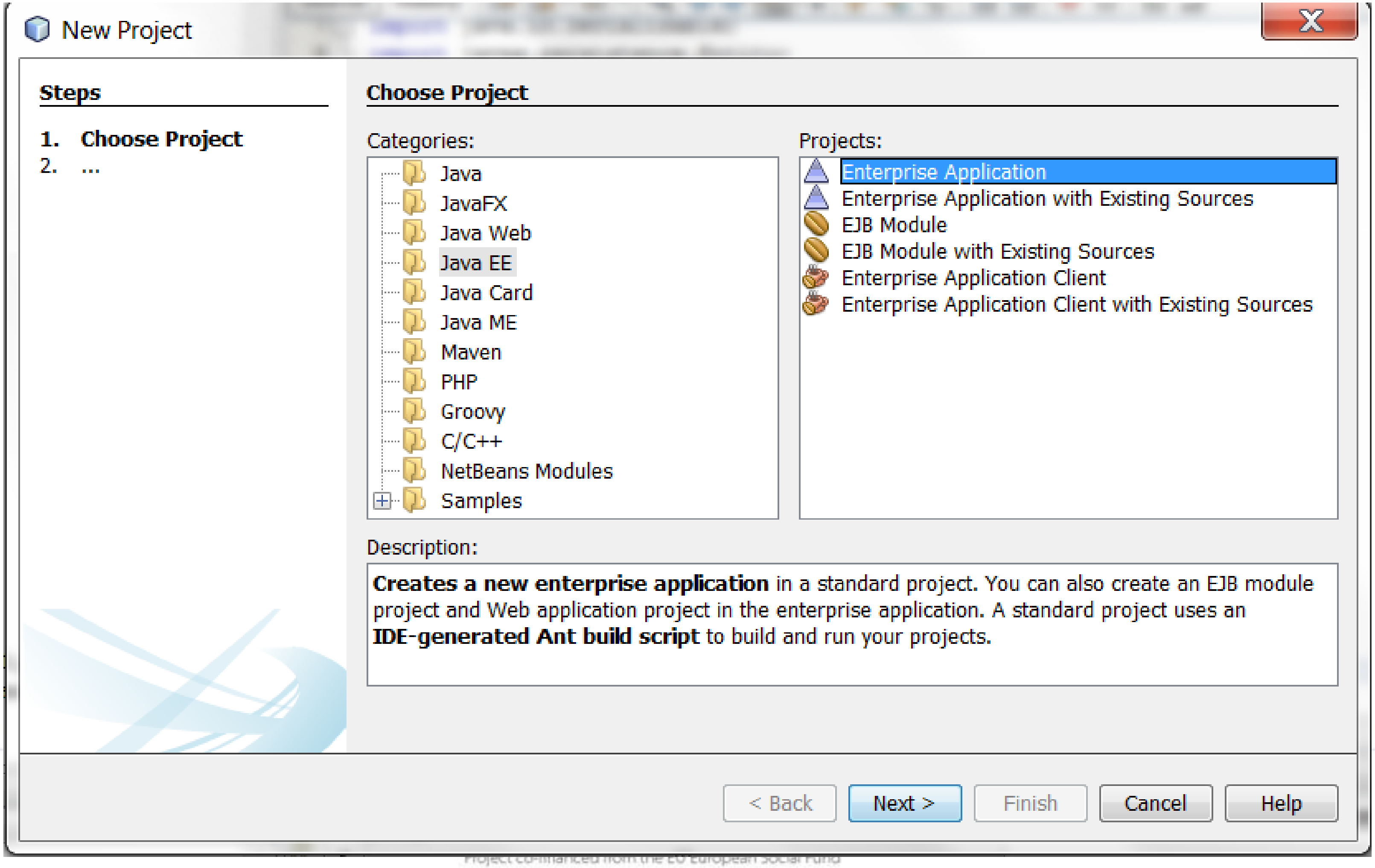
The methods storing the application data in database by using the JPA controllers

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;
  }
}
```

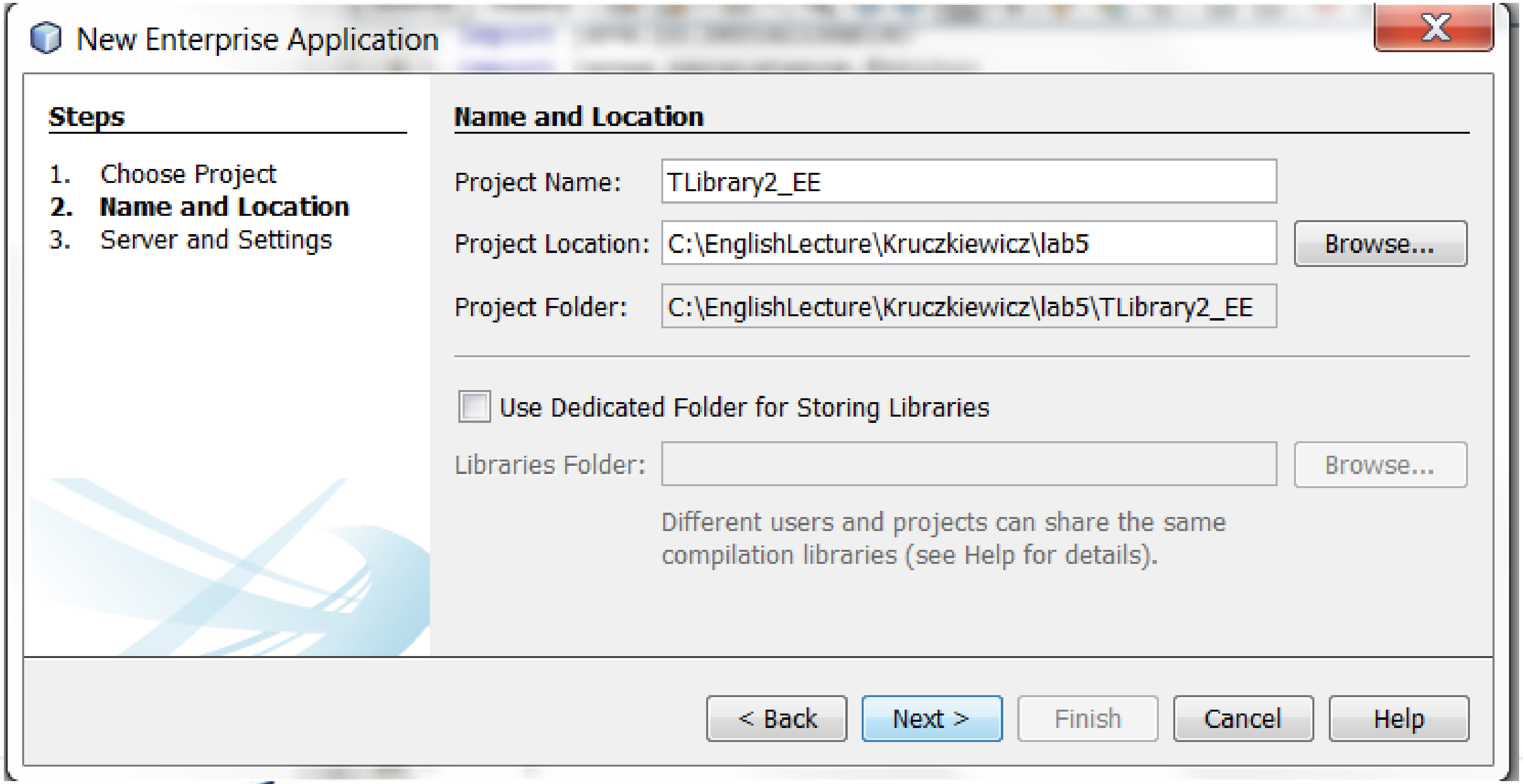
The method reading the data of theTTitle_book family from database by using the JPA controller (titleJpaController) and preparing the data model for the JSF h:dataTable view component

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**



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.

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

Steps

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

Name and Location

Project Name: TLibrary2_EE

Project Location: C:\EnglishLecture\Kruczkiewicz\lab5

Project Folder: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_EE

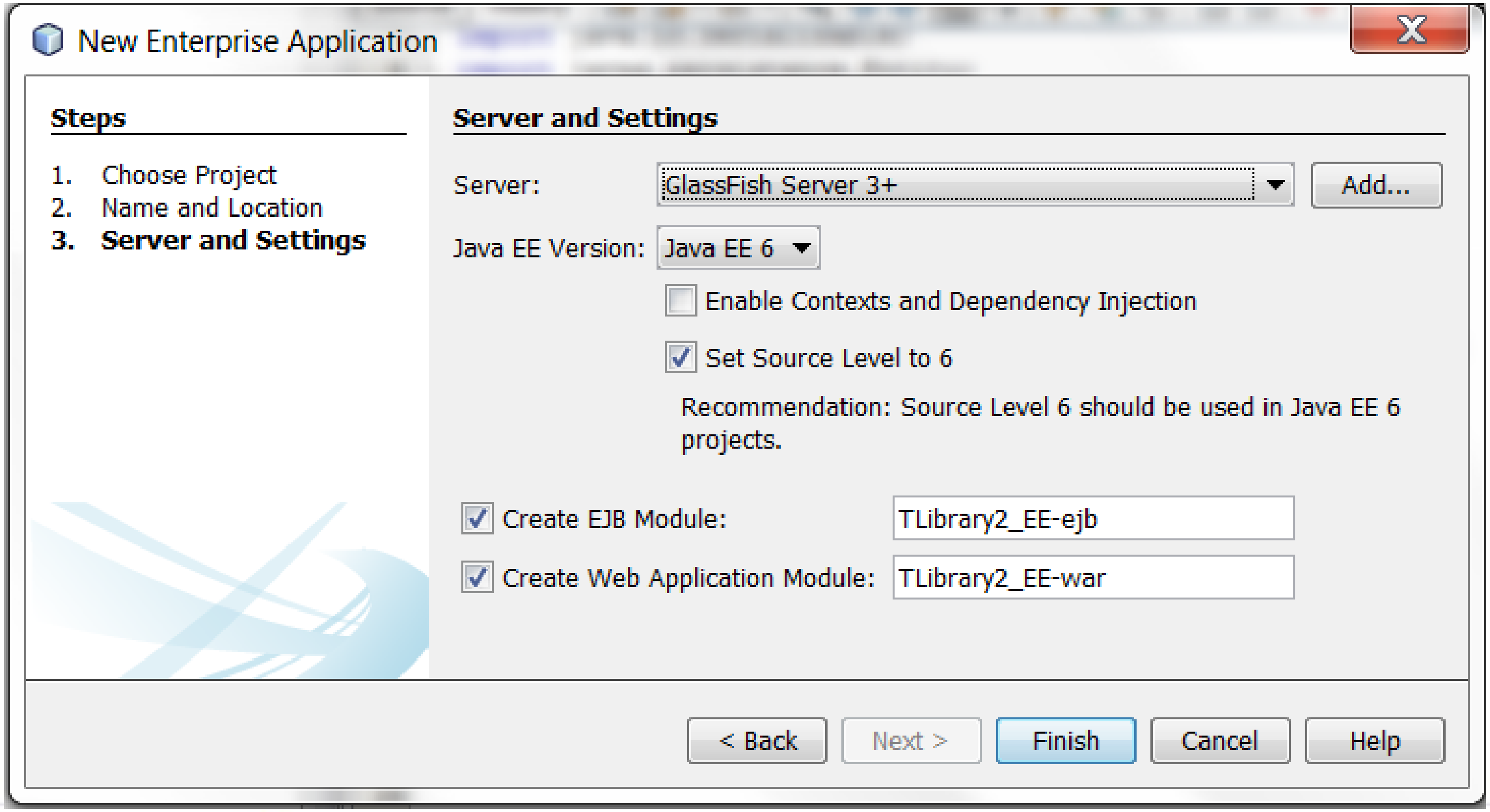
□ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

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)**



New Enterprise Application

Steps

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

Server and Settings

Server: GlassFish Server 3+

Java EE Version: Java EE 6

□ Enable Contexts and Dependency Injection

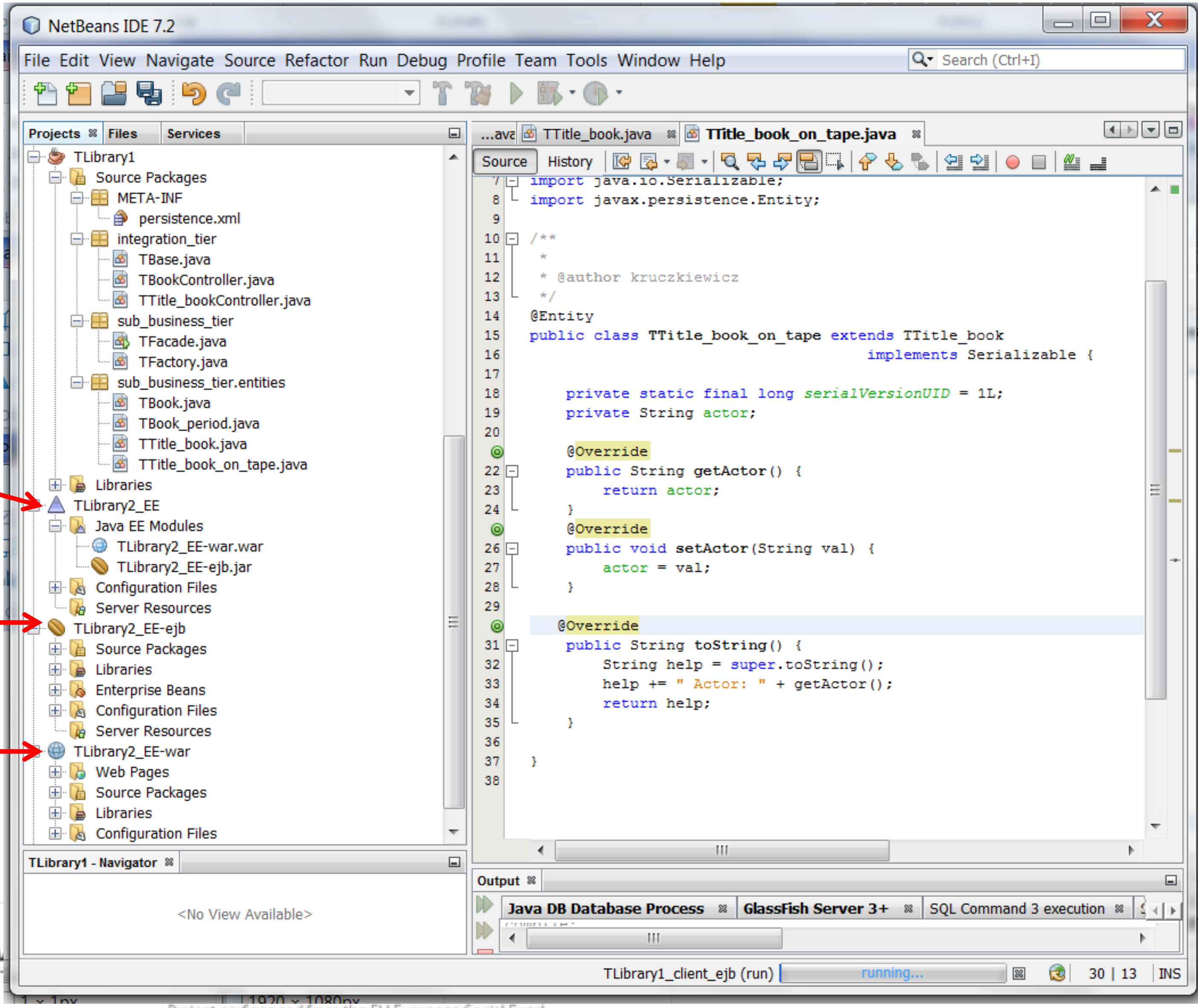
☑ Set Source Level to 6

Recommendation: Source Level 6 should be used in Java EE 6 projects.

☑ Create EJB Module: TLibrary2_EE-ejb

☑ Create Web Application Module: TLibrary2_EE-war

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)

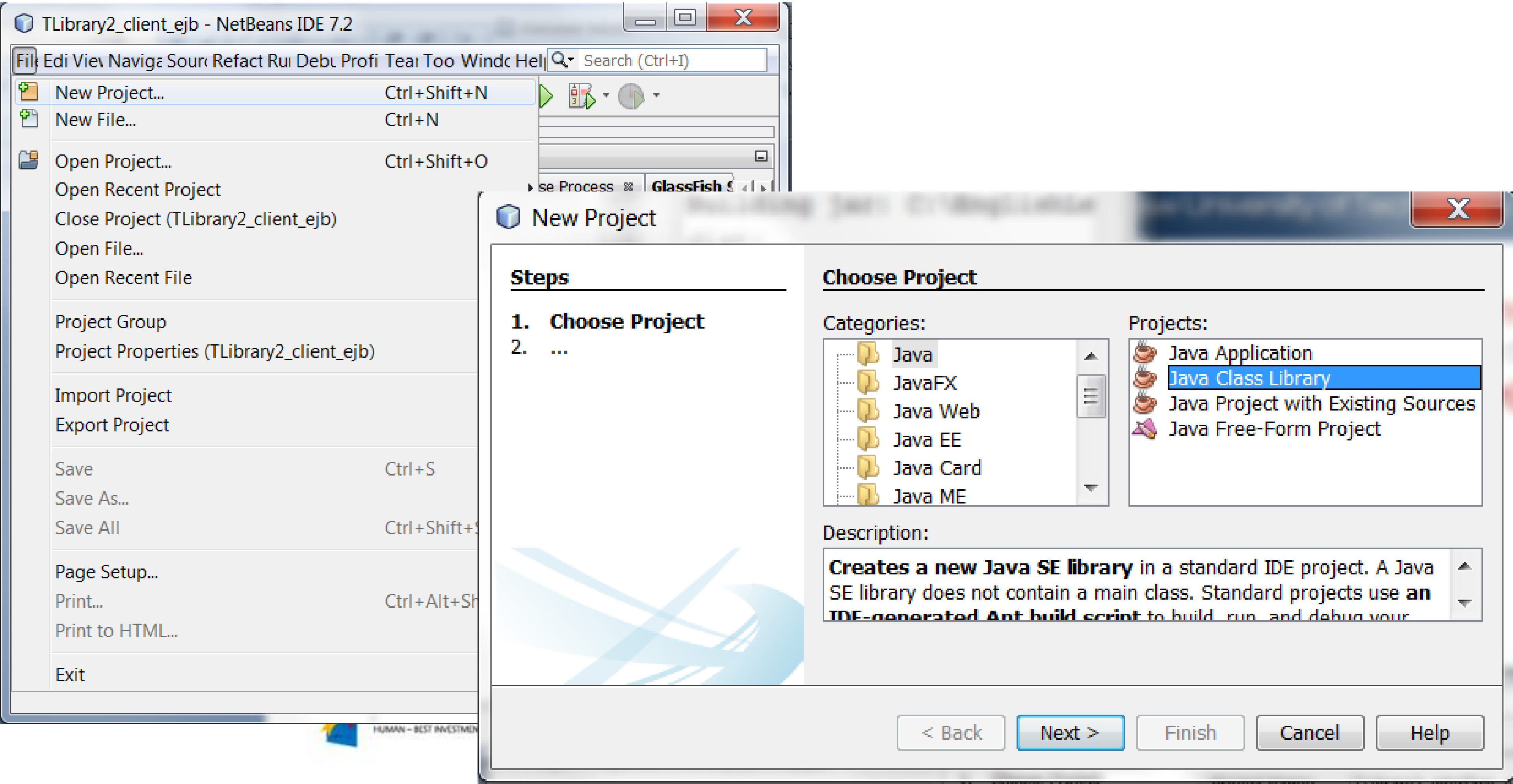


```
TLibrary1
    Source Packages
        META-INF
            persistence.xml
        integration_tier
            TBase.java
            TBookController.java
            TTitle_bookController.java
        sub_business_tier
            TFacade.java
            TFactory.java
        sub_business_tier.entities
            TBook.java
            TBook_period.java
            TTitle_book.java
            TTitle_book_on_tape.java
    Libraries
TLibrary2_EE
    Java EE Modules
        TLibrary2_EE-war.war
        TLibrary2_EE-ejb.jar
    Configuration Files
    Server Resources
TLibrary2_EE-ejb
    Source Packages
    Libraries
    Enterprise Beans
    Configuration Files
    Server Resources
TLibrary2_EE-war
    Web Pages
    Source Packages
    Libraries
    Configuration Files
```

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

/**
 *
 * @author kruczkiewicz
 */
@Entity
public class TTitle_book_on_tape extends TTitle_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;
    }

}
```

TLibrary1_client_ejb (run) running... 30 | 13 INS

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**)



TLibrary2_client_ejb - NetBeans IDE 7.2

New Project

Steps

1. **Choose Project**
2. ...

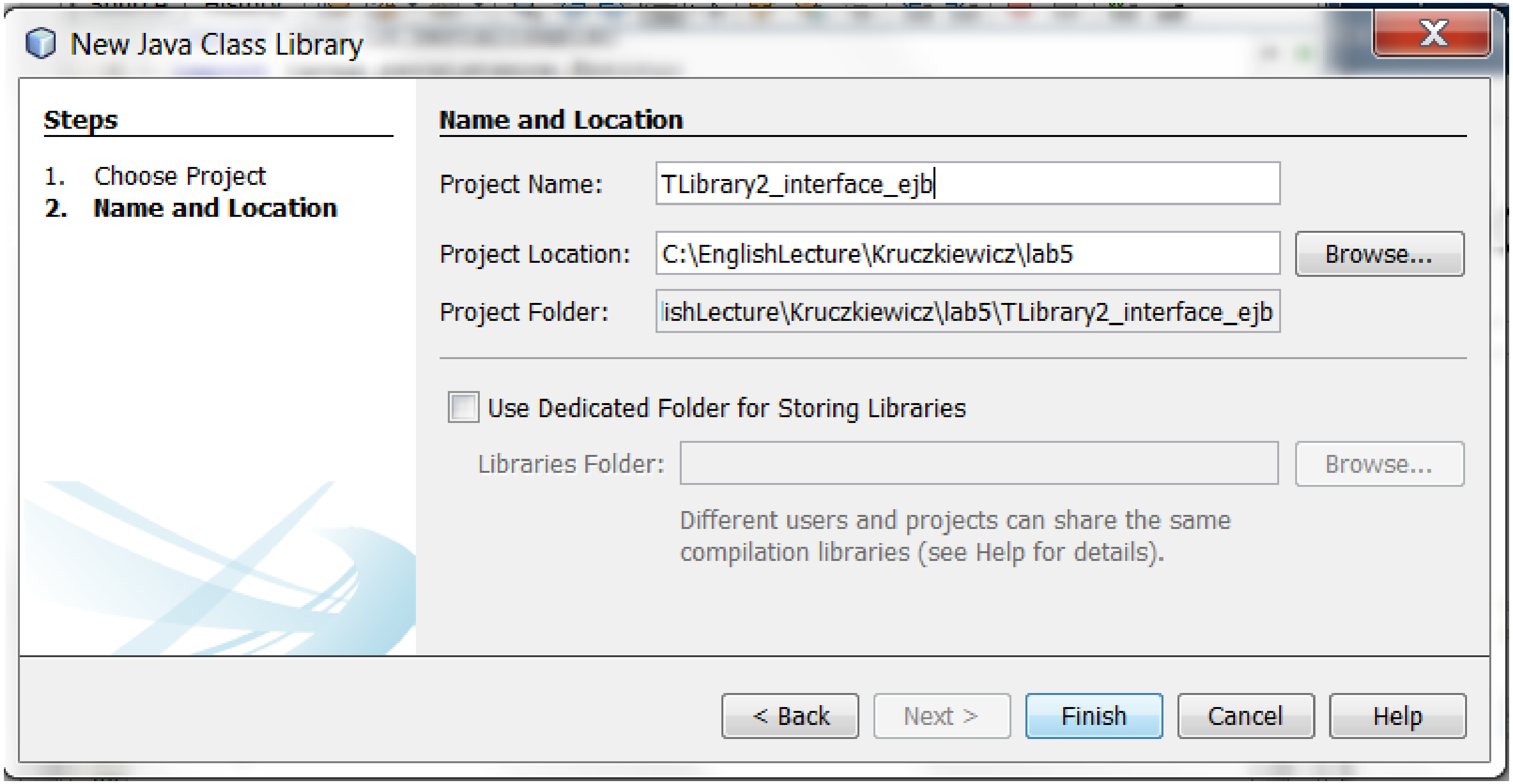
Choose Project

Categories: Java

Projects: Java Class Library

Description:

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



New Java Class Library

Steps

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

Name and Location

Project Name: TLibrary2_interface_ejb

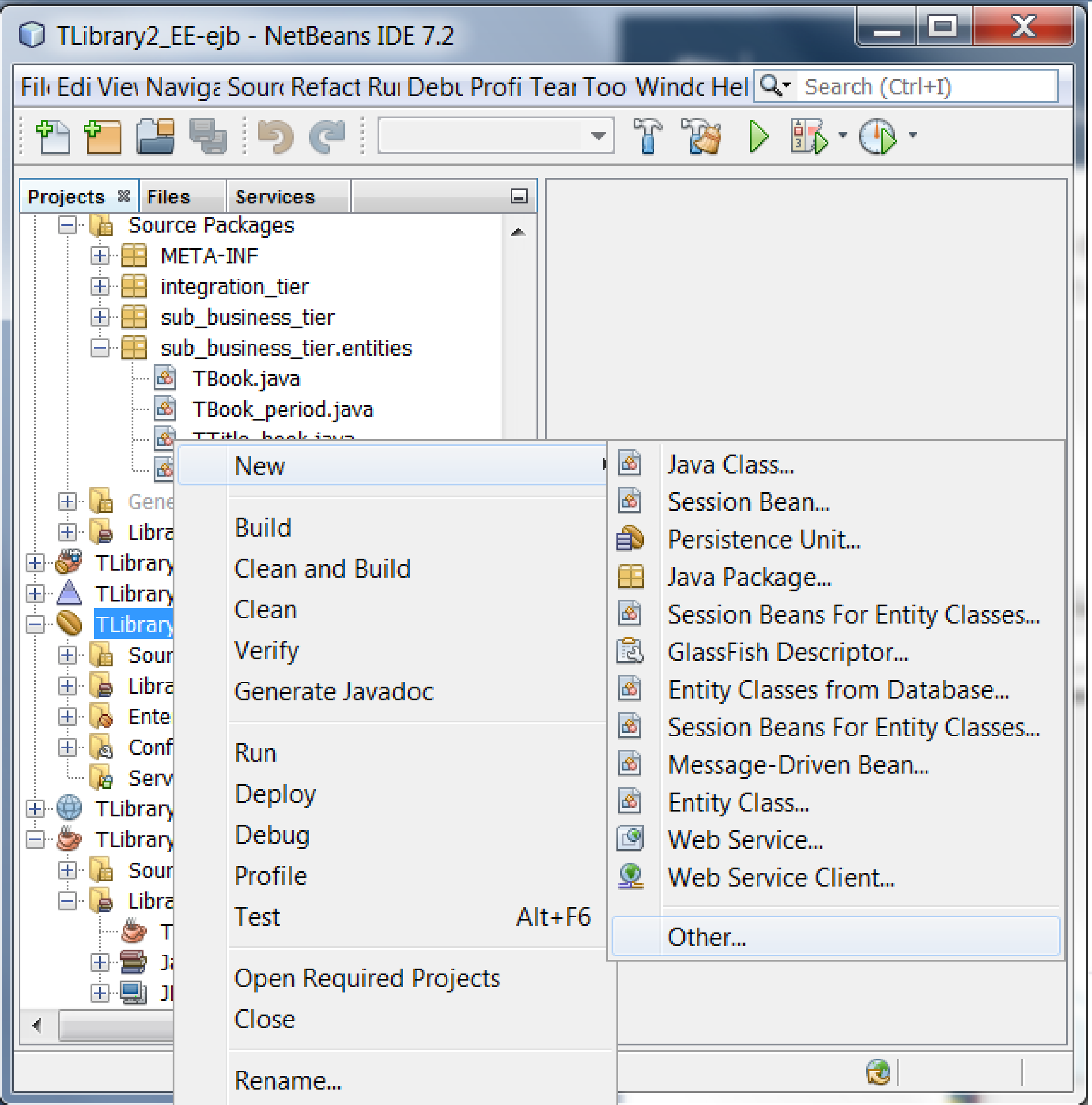
Project Location: C:\EnglishLecture\Kruczkiewicz\lab5

Project Folder: ishLecture\Kruczkiewicz\lab5\TLibrary2_interface_ejb

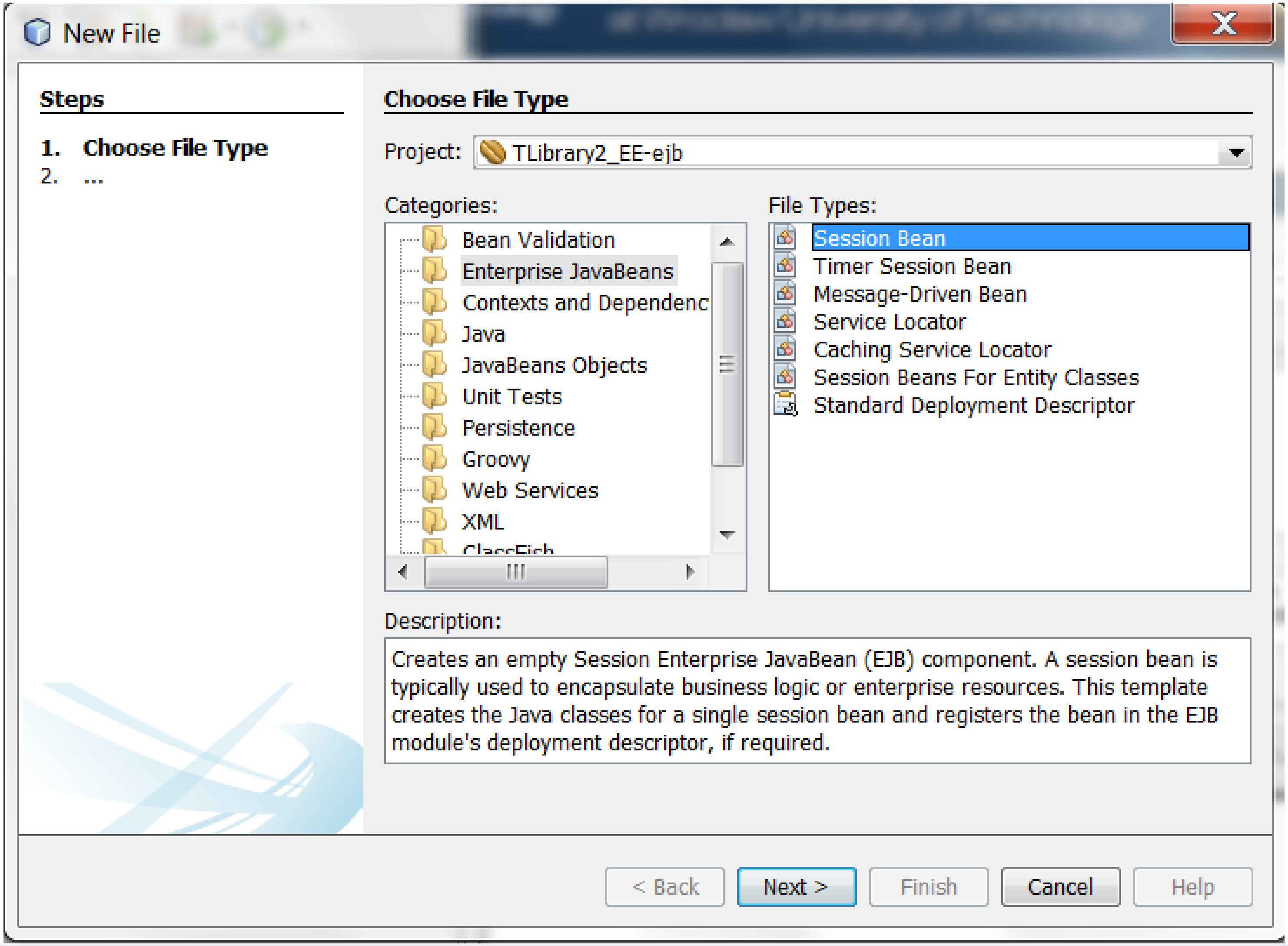
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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



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

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
- GlassFish

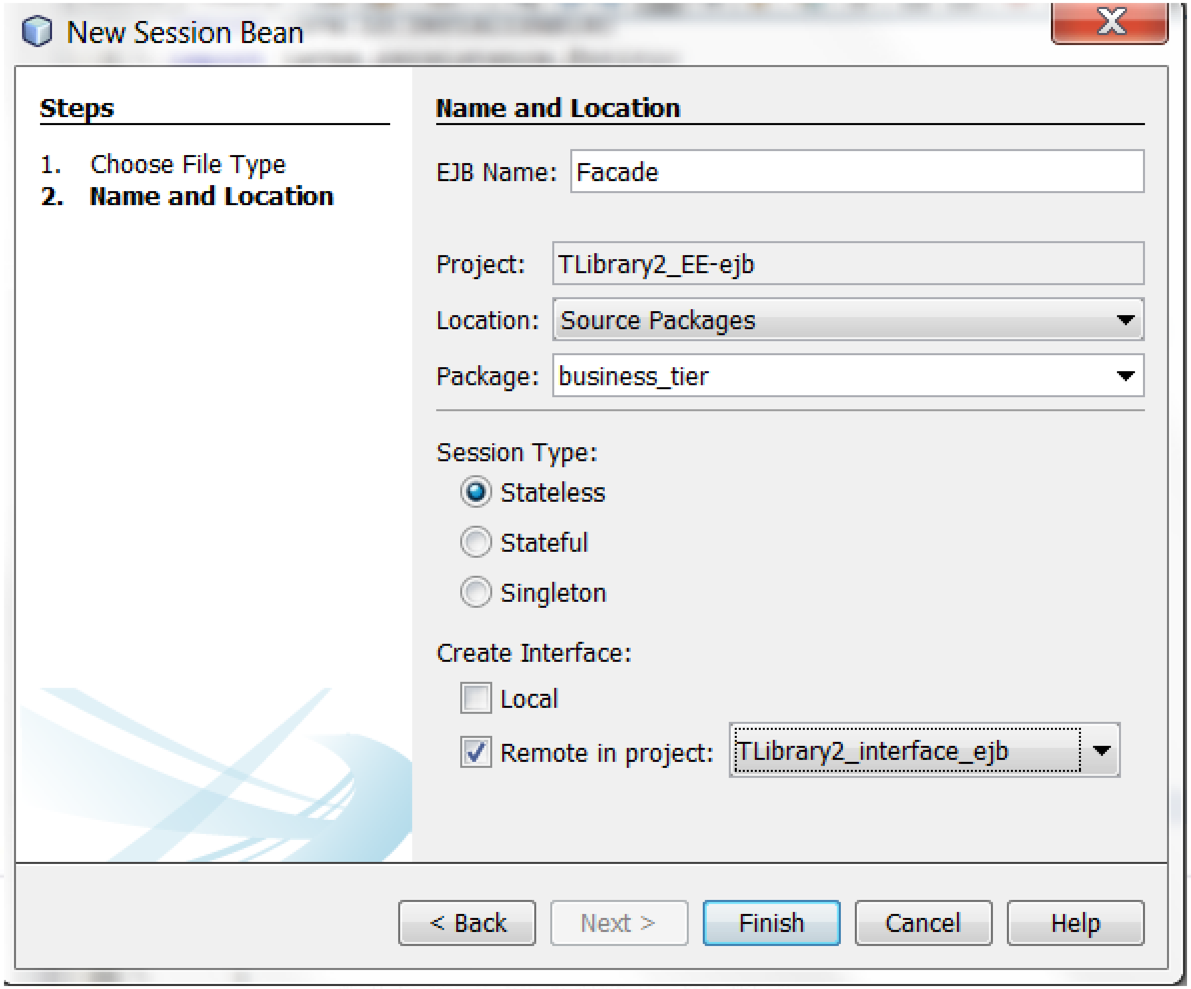
File Types:

- 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.

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: Facade

Project: TLibrary2_EE-ejb

Location: Source Packages

Package: business_tier

Session Type:

● Stateless

○ Stateful

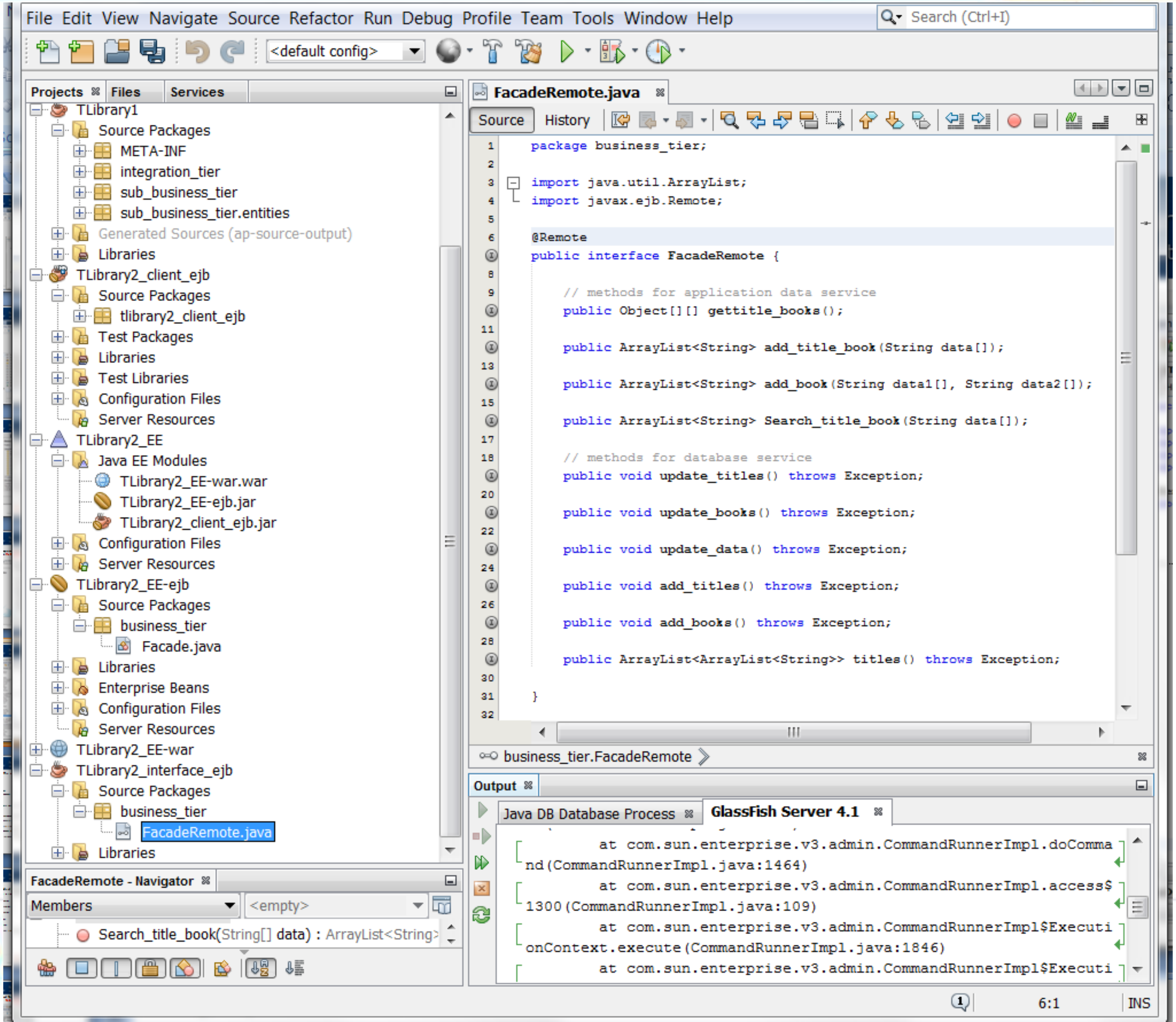
○ Singleton

Create Interface:

□ Local

☑ Remote in project: TLibrary2_interface_ejb

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



```
package business_tier;

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

@Remote
public interface FacadeRemote {

    // methods for application data service
    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[]);

    // methods for database service
    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;

}
```

```
        at com.sun.enterprise.v3.admin.CommandRunnerImpl.doComma
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
```

4.4. The code of FacadeRemote inteface

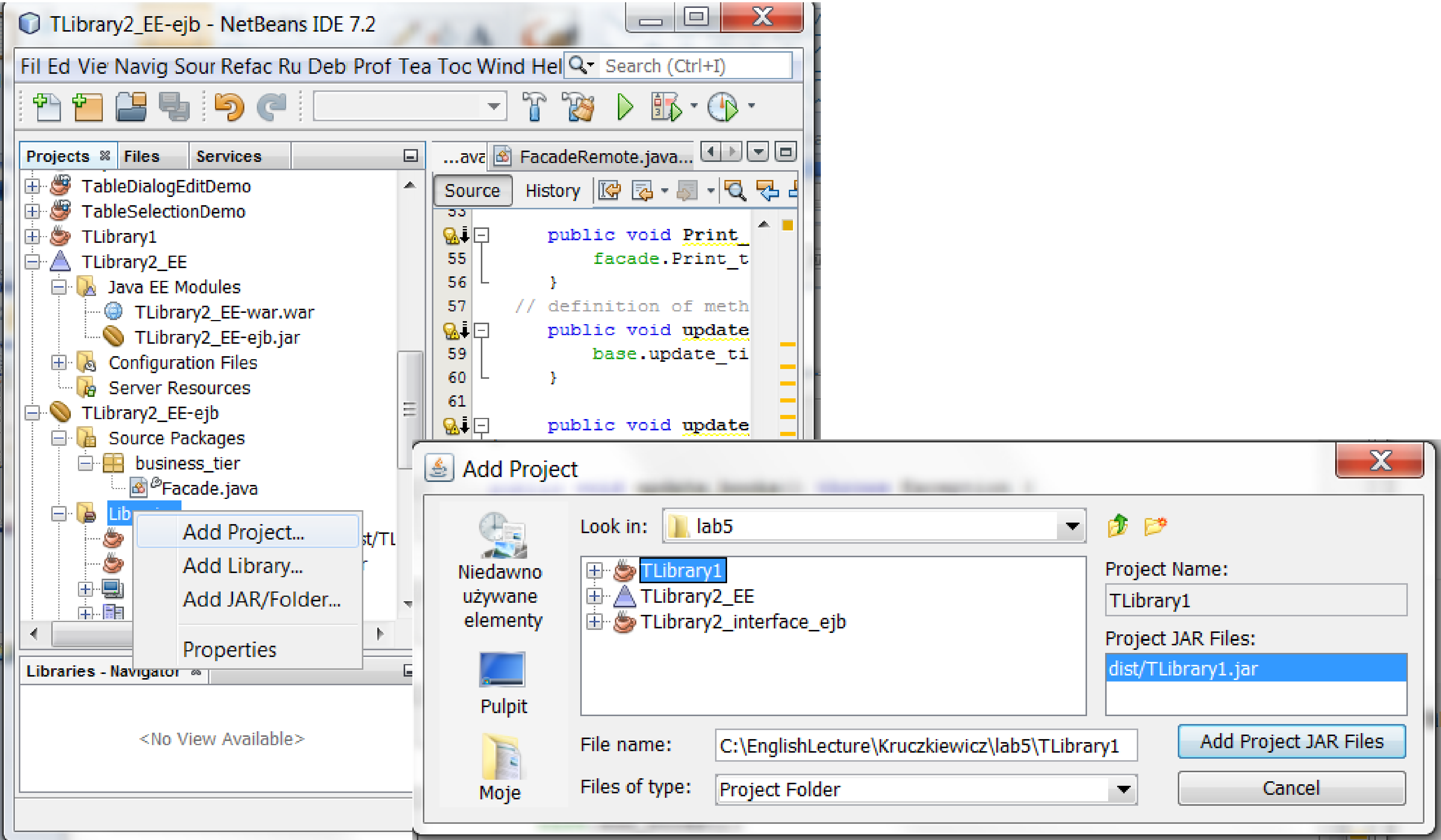
```
package business_tier;


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 inteface - continuation

// methods for database service – fromTBase 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



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

    public void update
```

Look in: lab5

Project Name: TLibrary1

Project JAR Files: dist/TLibrary1.jar

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

Files of type: Project Folder

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);
```

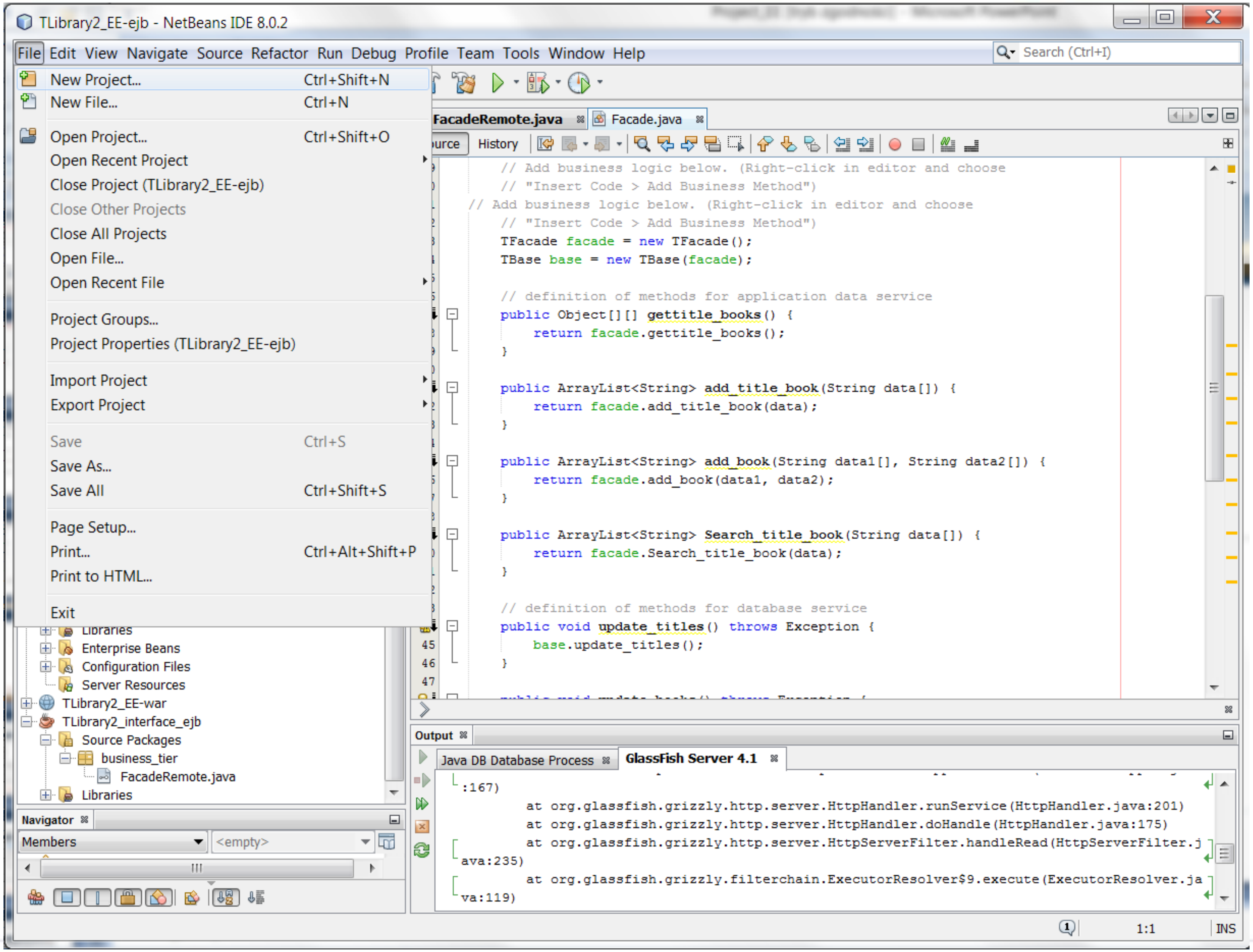
The code of Facade class - continuation

```
// 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();
   }
}
```

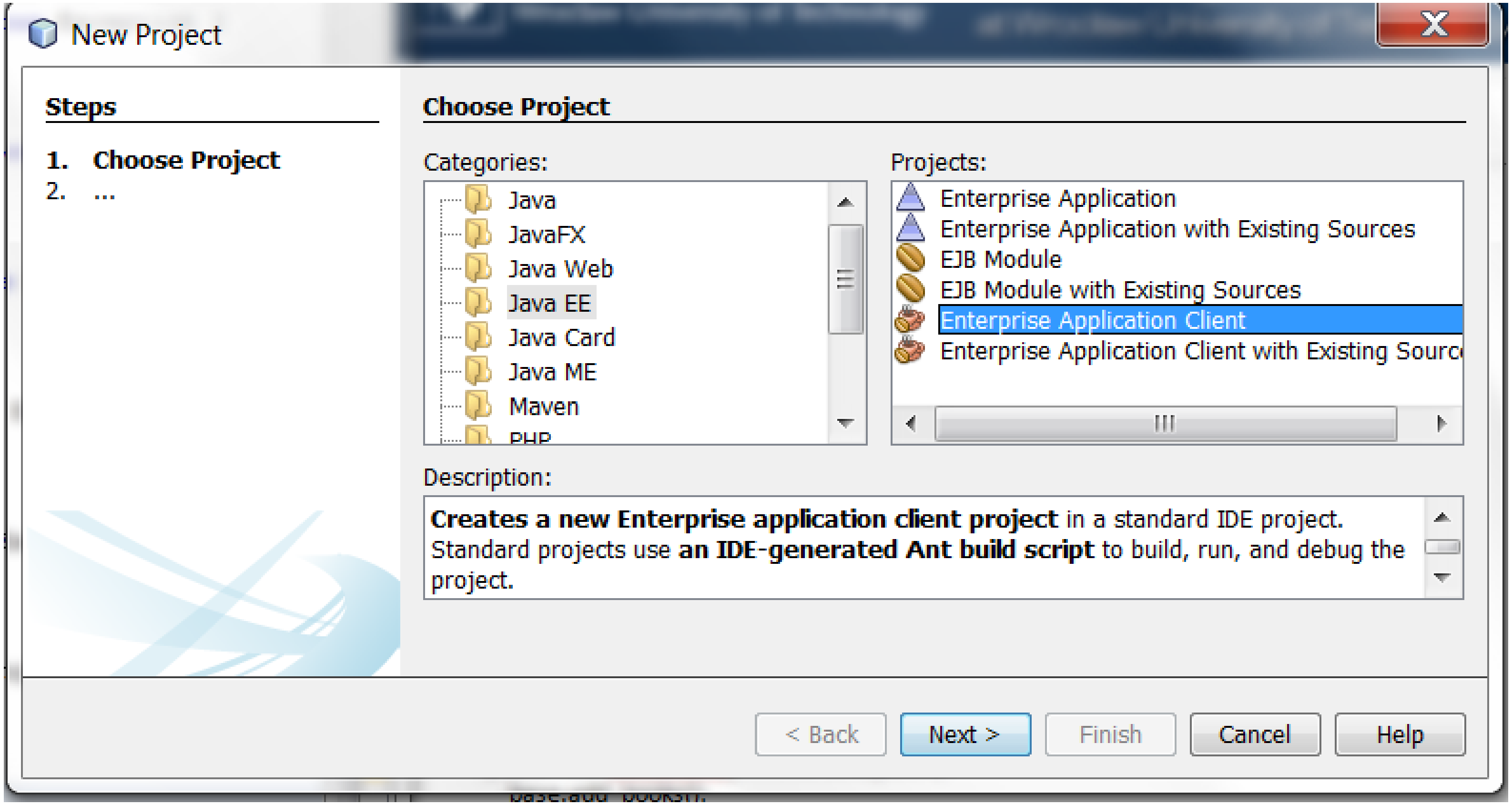
5. Creation of the EE client tier



```
    // 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();
    }
```

```
:167)
        at org.glassfish.grizzly.http.server.HttpHandler.runService(HttpHandler.java:201)
        at org.glassfish.grizzly.http.server.HttpHandler.doHandle(HttpHandler.java:175)
        at org.glassfish.grizzly.http.server.HttpServerFilter.handleRead(HttpServerFilter.j
ava:235)
        at org.glassfish.grizzly.filterchain.ExecutorResolver$9.execute(ExecutorResolver.ja
va:119)
```

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



New Project

Steps

1. **Choose Project**
2. ...

Choose Project

Categories:

- Java
- JavaFX
- Java Web
- Java EE
- Java Card
- Java ME
- Maven
- PHP

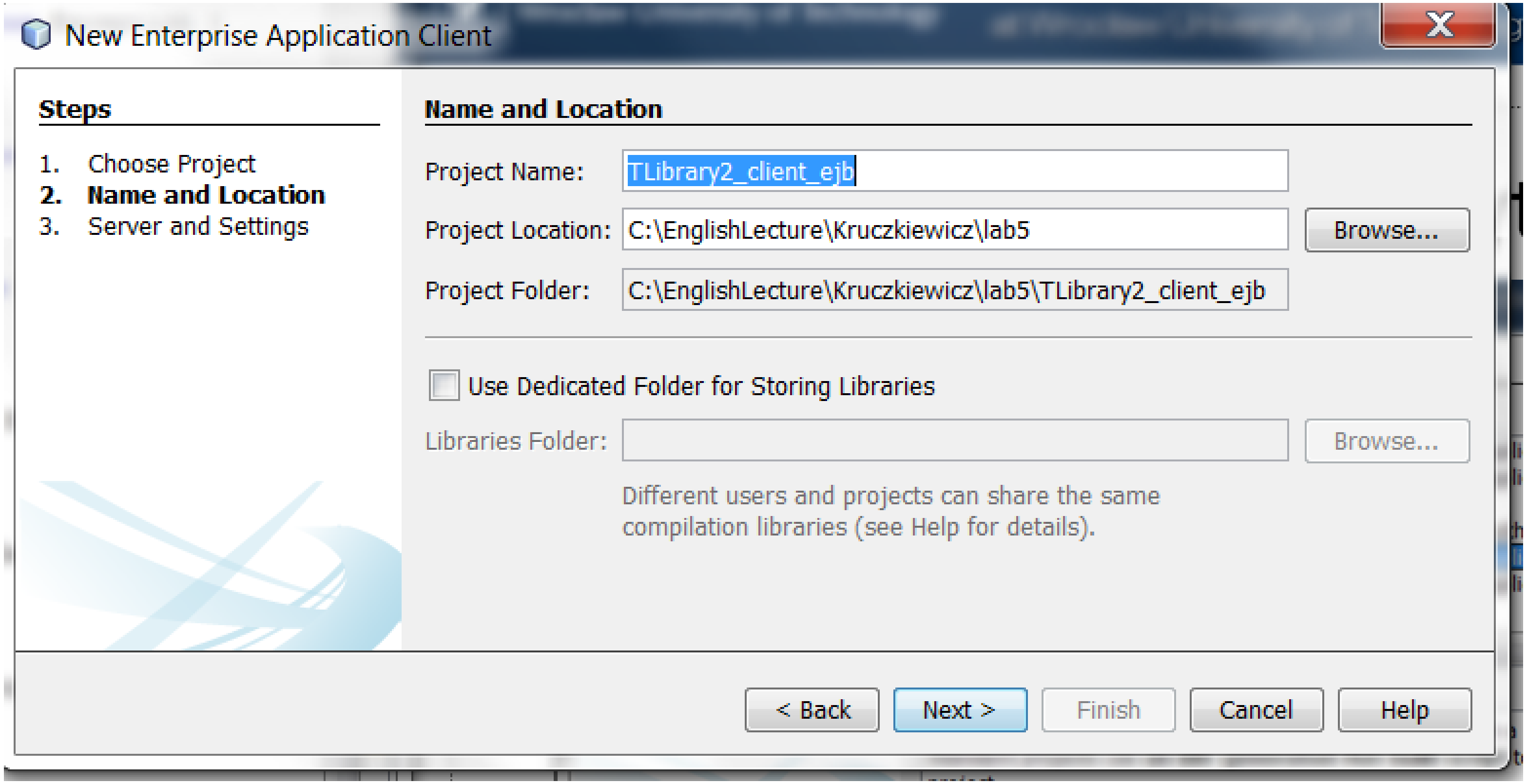
Projects:

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

Description:

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.

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



New Enterprise Application Client

Name and Location

Project Name: TLibrary2_client_ejb

Project Location: C:\EnglishLecture\Kruczkiewicz\lab5

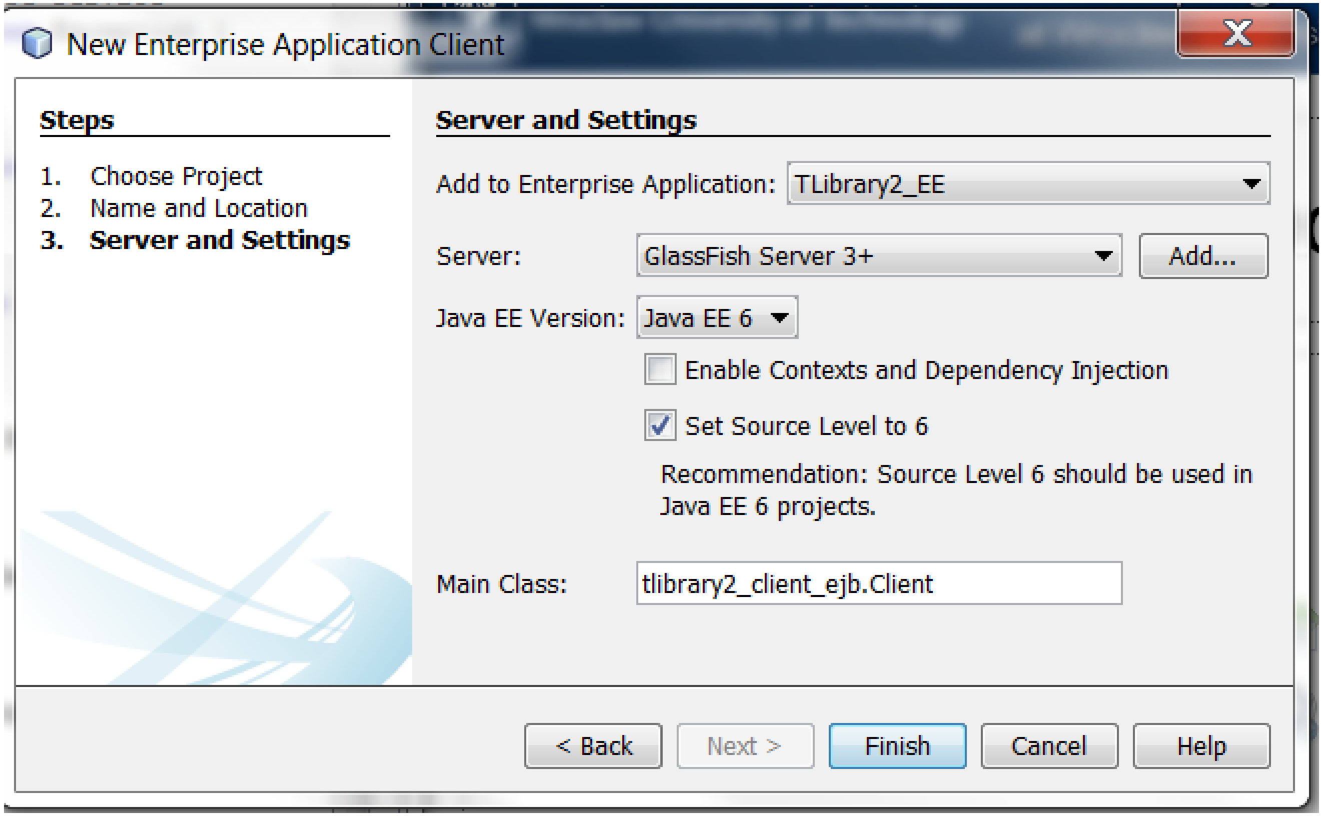
Project Folder: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_client_ejb

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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

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



New Enterprise Application Client

Server and Settings

Add to Enterprise Application: TLibrary2_EE

Server: GlassFish Server 3+

Java EE Version: Java EE 6

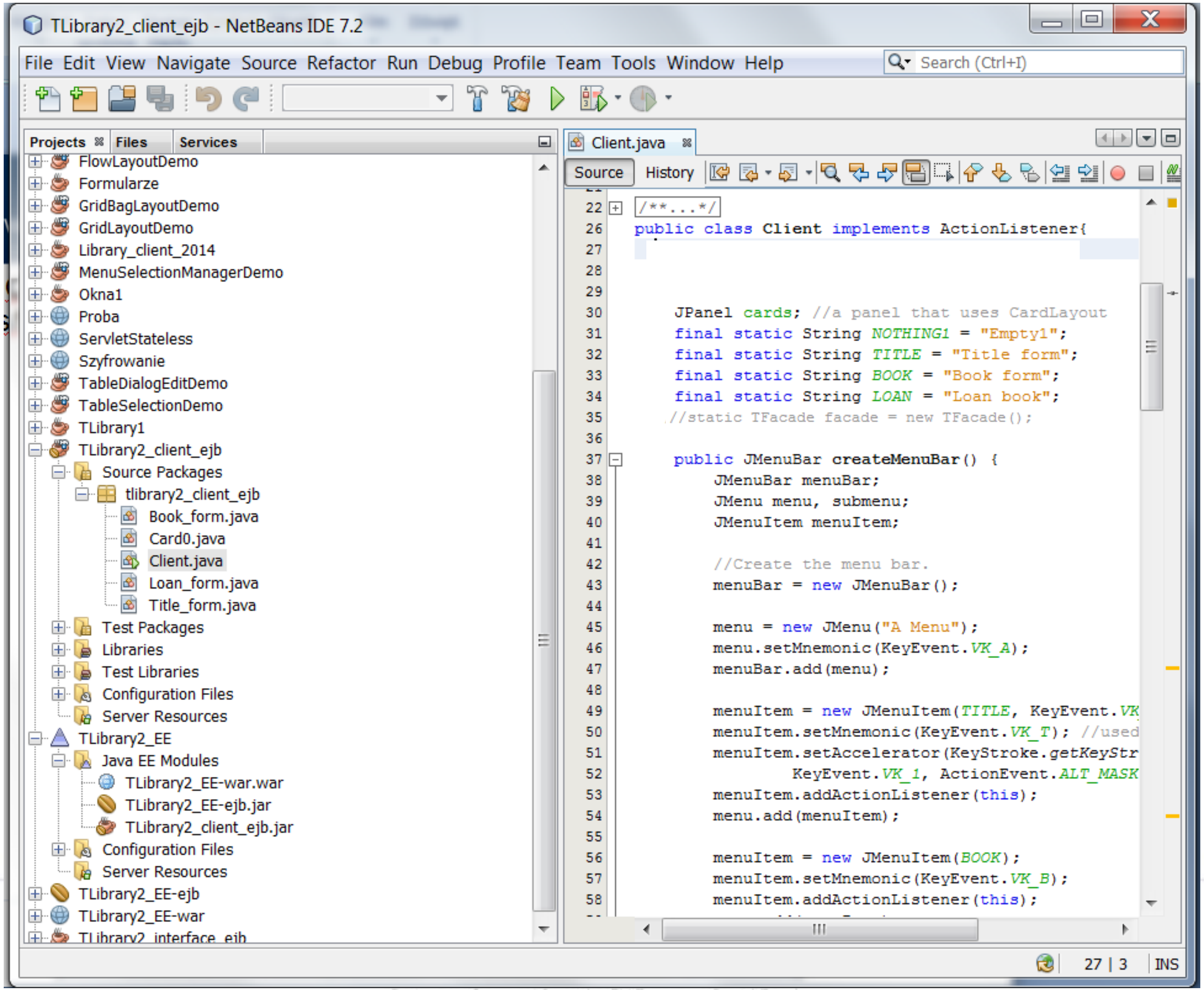
□ Enable Contexts and Dependency Injection

☑ Set Source Level to 6

Recommendation: Source Level 6 should be used in Java EE 6 projects.

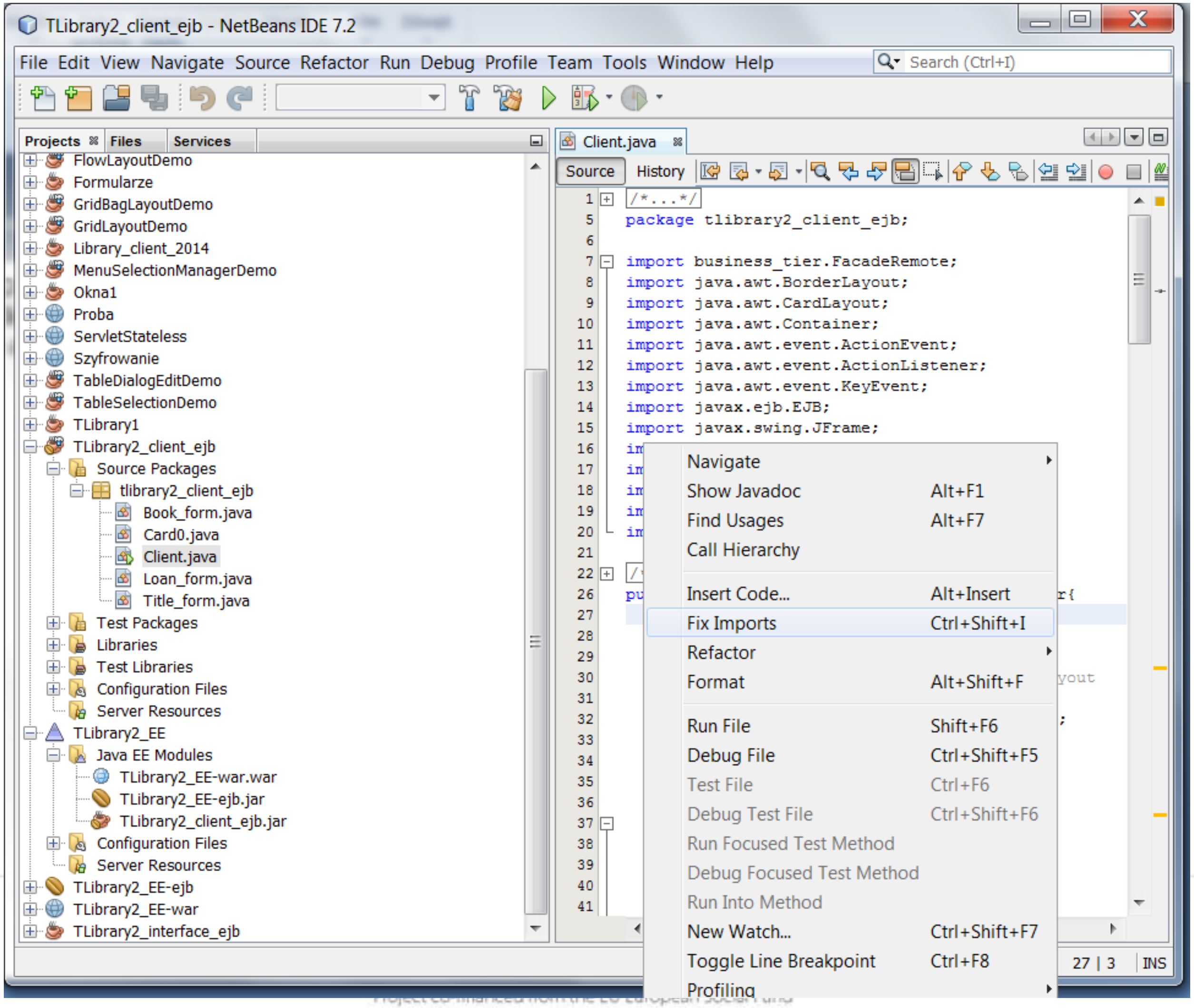
Main Class: tlibrary2_client_ejb.Client

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



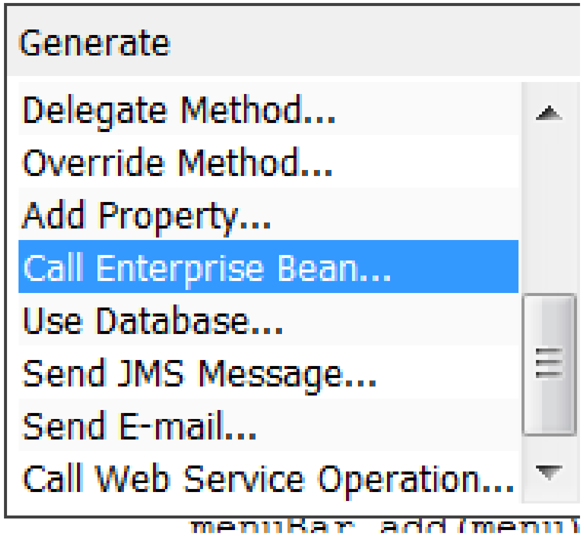
```java
/**...*/
public class Client implements ActionListener{



     JPanel cards; //a panel that uses CardLayout
     final static String NOTHING1 = "Empty1";
     final static String TITLE = "Title form";
     final static String BOOK = "Book form";
     final static String LOAN = "Loan book";
    //static TFacade facade = new TFacade();

     public JMenuBar createMenuBar() {
         JMenuBar menuBar;
         JMenu menu, submenu;
         JMenuItem menuItem;

         //Create the menu bar.
         menuBar = new JMenuBar();

         menu = new JMenu("A Menu");
         menu.setMnemonic(KeyEvent.VK_A);
         menuBar.add(menu);

         menuItem = new JMenuItem(TITLE, KeyEvent.VK
         menuItem.setMnemonic(KeyEvent.VK_T); //used
         menuItem.setAccelerator(KeyStroke.getKeyStr
                   KeyEvent.VK_1, ActionEvent.ALT_MASK
         menuItem.addActionListener(this);
         menu.add(menuItem);

         menuItem = new JMenuItem(BOOK);
         menuItem.setMnemonic(KeyEvent.VK_B);
         menuItem.addActionListener(this);
```

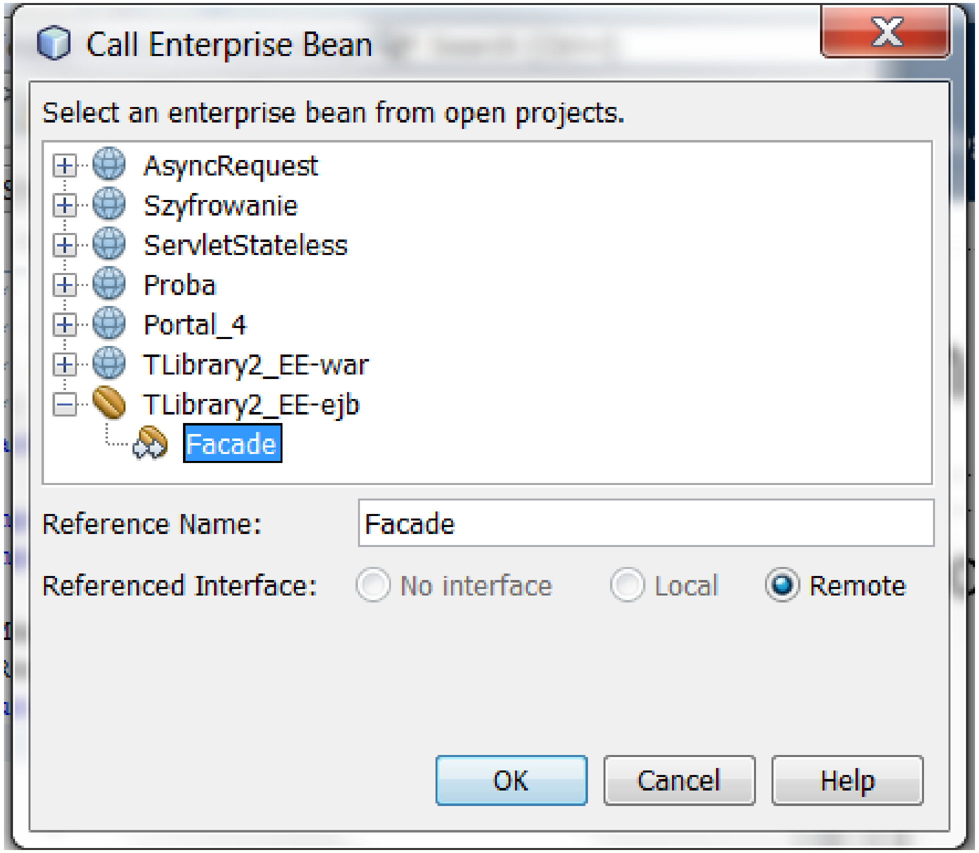
5.5. Creation of connection from the Facade Session Bean of Library_EJB1-ejb module to the Client class of the **TLibrary2_client_ejb** project - right click the edytor window with Client class and choose the Insert code item.



```
/*...*/
package tlibrary2_client_ejb;

import business_tier.FacadeRemote;
import java.awt.BorderLayout;
import java.awt.CardLayout;
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import javax.ejb.EJB;
import javax.swing.JFrame;
im
im
im
im
im

/
pu                                                  r{
```

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





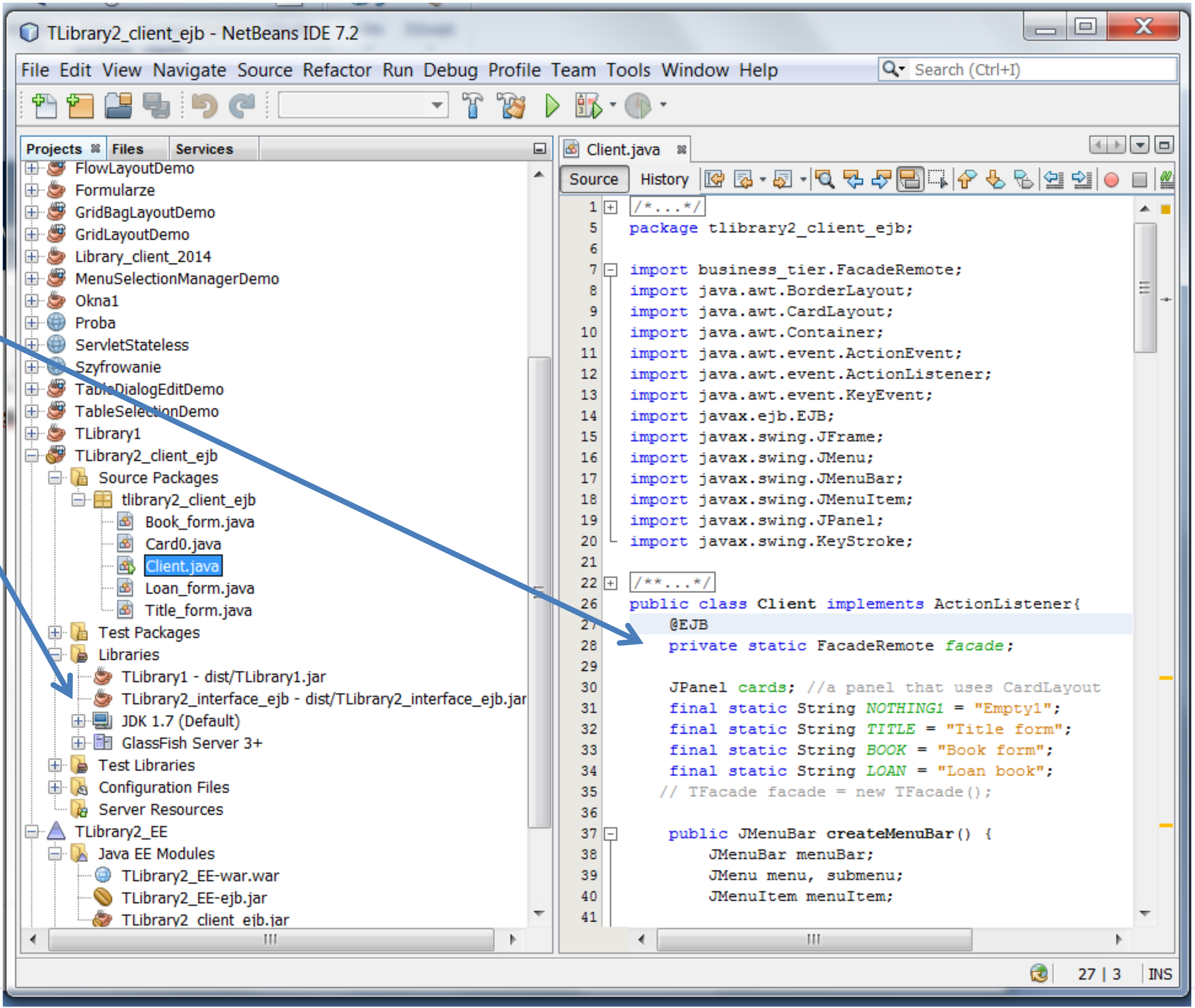
Select an enterprise bean from open projects.

- AsyncRequest
- Szyfrowanie
- ServletStateless
- Proba
- Portal_4
- TLibrary2_EE-war
- TLibrary2_EE-ejb
  - Facade

Reference Name: Facade

Referenced Interface: ○ No interface ○ Local ● Remote

5.7. The result



```
/*...*/
package tlibrary2_client_ejb;

import business_tier.FacadeRemote;
import java.awt.BorderLayout;
import java.awt.CardLayout;
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import javax.ejb.EJB;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.KeyStroke;

/**...*/
public class Client implements ActionListener{
    @EJB
    private static FacadeRemote facade;

    JPanel cards; //a panel that uses CardLayout
    final static String NOTHING1 = "Empty1";
    final static String TITLE = "Title form";
    final static String BOOK = "Book form";
    final static String LOAN = "Loan book";
  // TFacade facade = new TFacade();

    public JMenuBar createMenuBar() {
        JMenuBar menuBar;
        JMenu menu, submenu;
        JMenuItem menuItem;

```

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);
    }
```

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

Client.getFacade().add_title_book(data);
    }
```

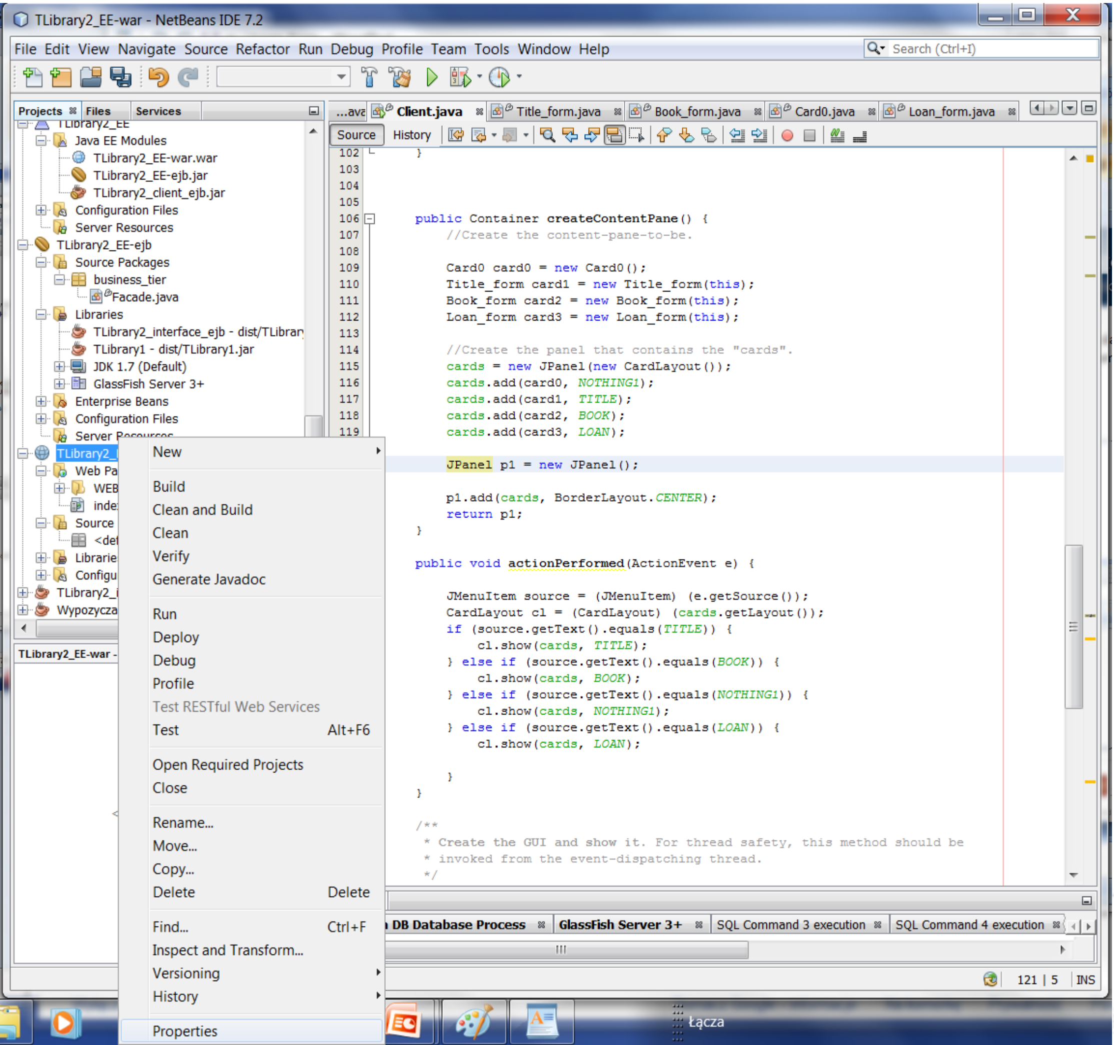
Object POJO in SE Client tier

EJB in EE Client tier

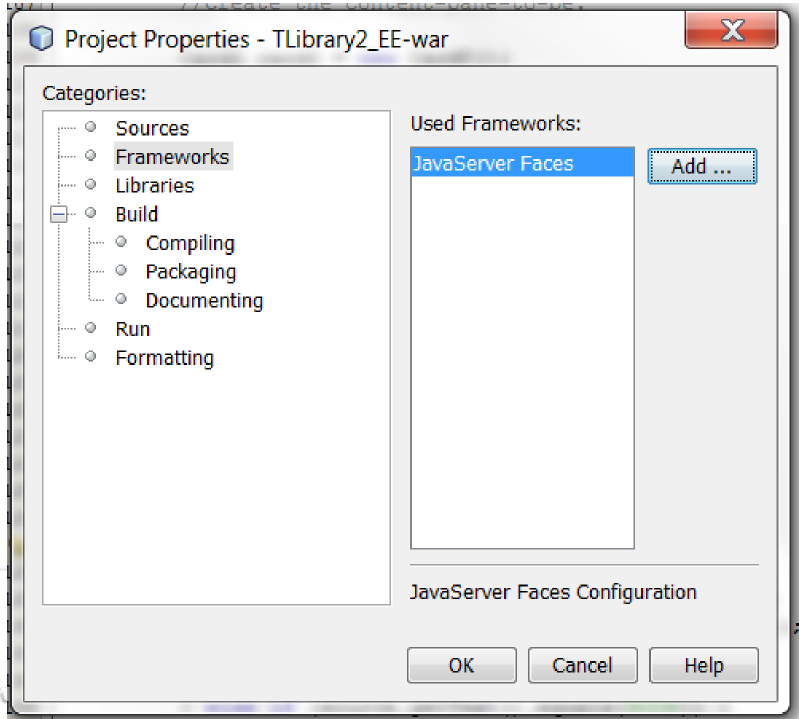
6. 0. Definition of the Web Client Tier (TLibrary2_EE-war, prepared during creation of TLibrary2_EE) project) for inserting application data into the database and displaying the data from database – change the Web Framework to the JavaServer Faces type

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



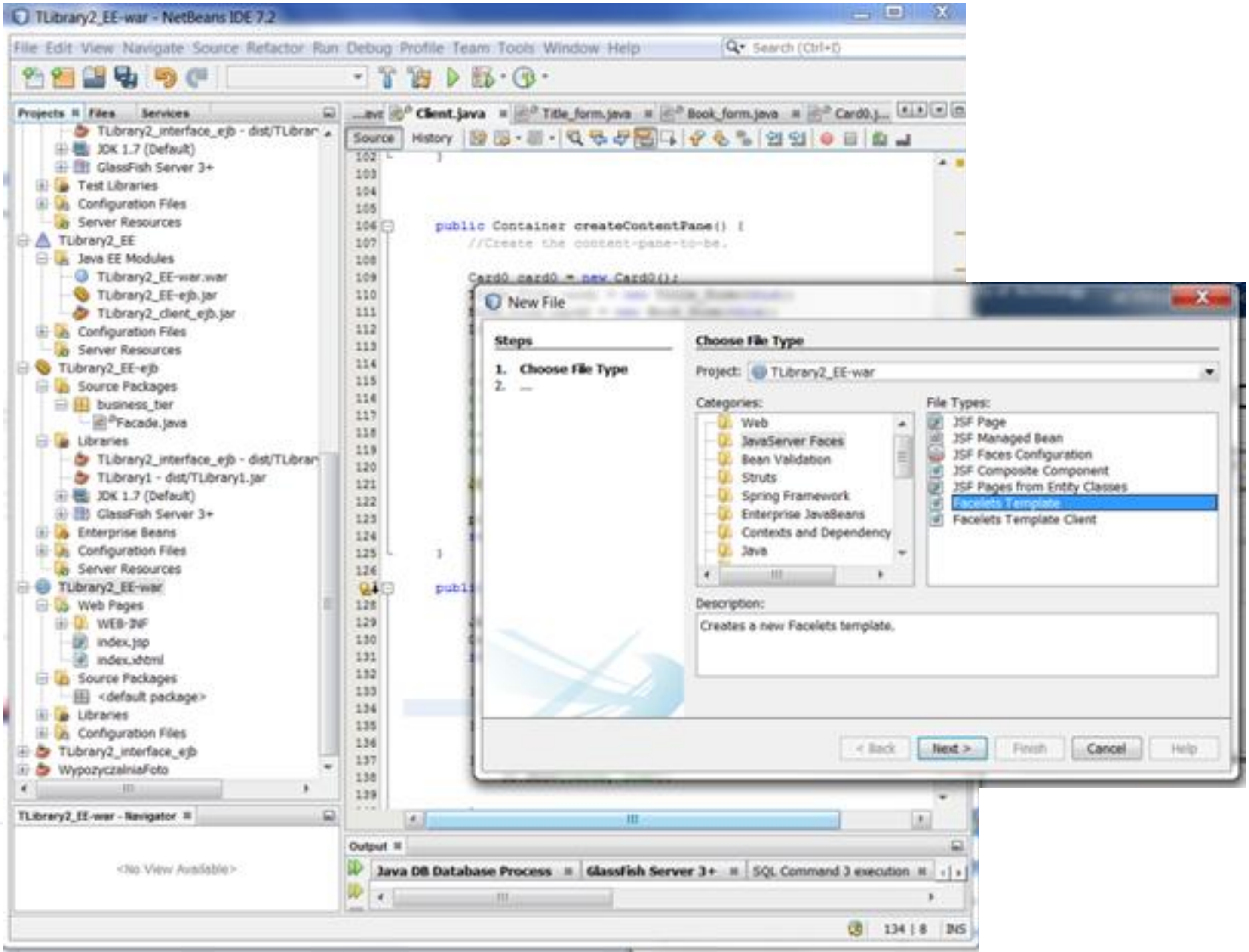
```
    }


    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.
 */
```



Used Frameworks: JavaServer Faces

JavaServer Faces Configuration

6.1. **Addition the template of web pages: R**ight 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.



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

    Card0 card0 = new Card0();
```

New File

Choose File Type

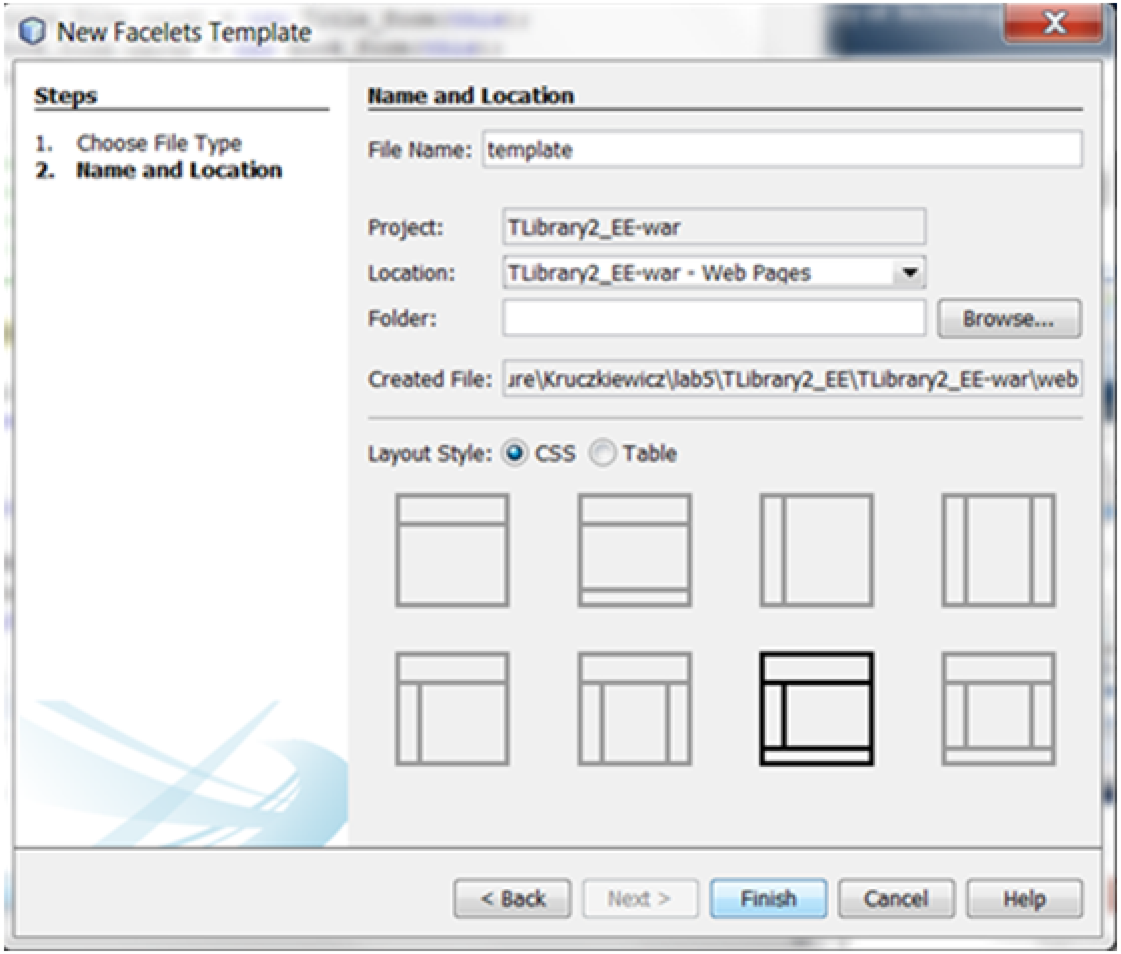
Project: TLibrary2_EE-war

Categories: JavaServer Faces

File Types: Facelets Template

Description:
Creates a new Facelets template.

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



New Facelets Template

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

Name and Location

File Name: template

Project: TLibrary2_EE-war

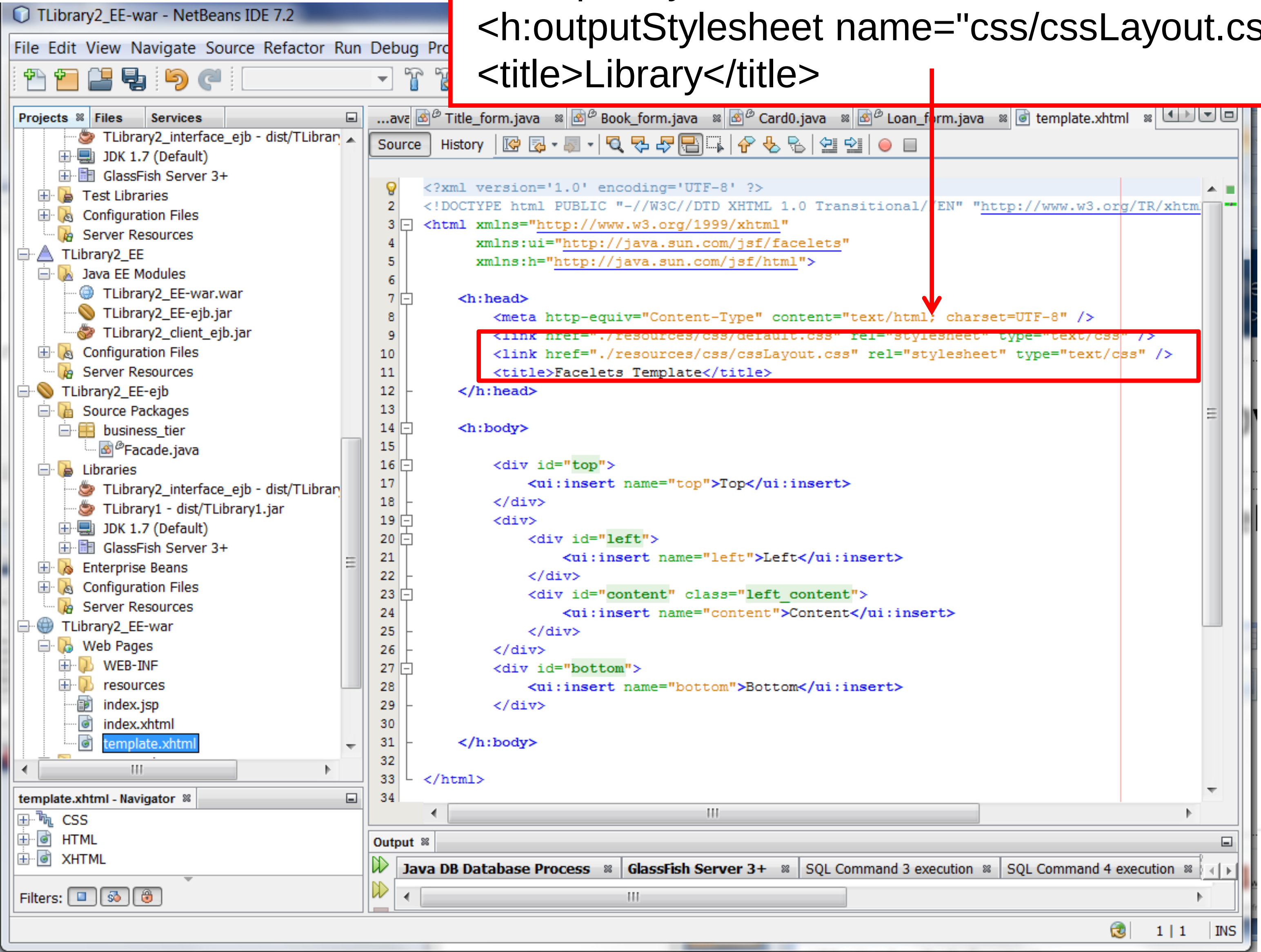
Location: TLibrary2_EE-war - Web Pages

Folder:

Created File: ure\Kruczkiewicz\lab5\TLibrary2_EE\TLibrary2_EE-war\web

Layout Style: ● CSS ○ Table

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



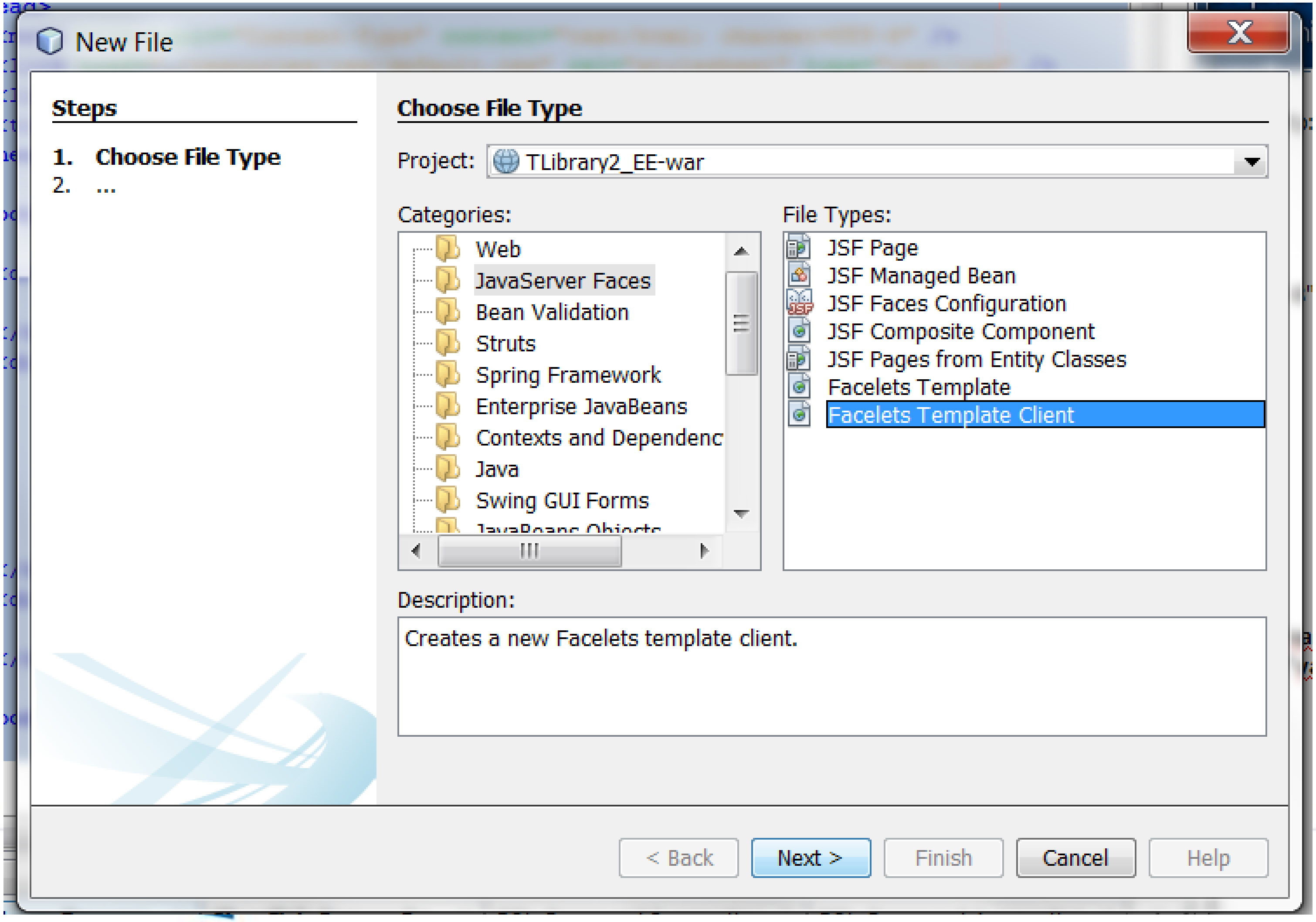
```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtm
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html">

    <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>
        <div id="bottom">
            <ui:insert name="bottom">Bottom</ui:insert>
        </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 File

Steps
1. Choose File Type
2. ...

Choose File Type

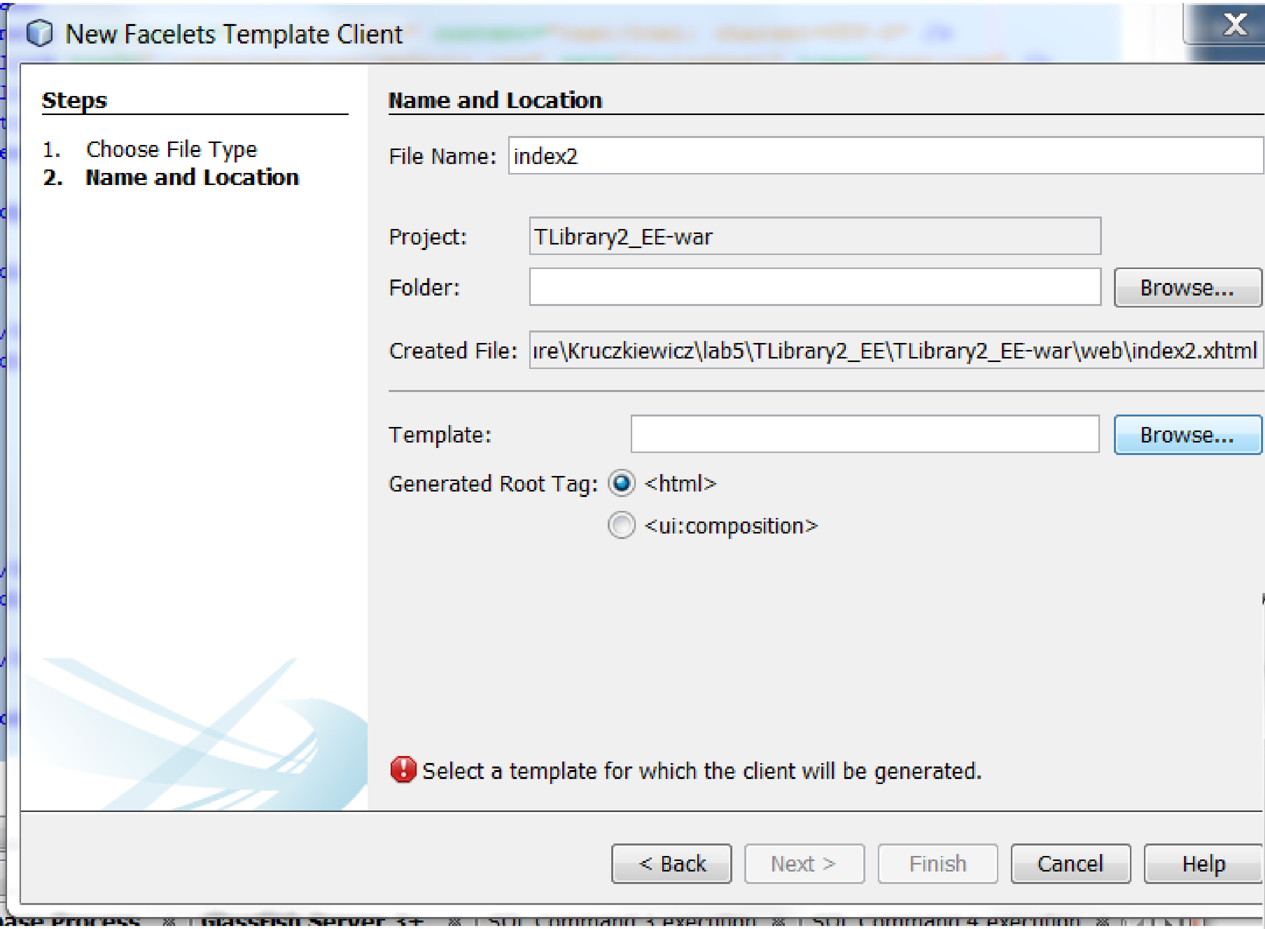
Project: TLibrary2_EE-war

Categories:
- Web
- JavaServer Faces
- Bean Validation
- Struts
- Spring Framework
- Enterprise JavaBeans
- Contexts and Dependenc
- Java
- Swing GUI Forms

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

Description:
Creates a new Facelets template client.

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



New Facelets Template Client

Steps

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

Name and Location

File Name: index2

Project: TLibrary2_EE-war

Folder:

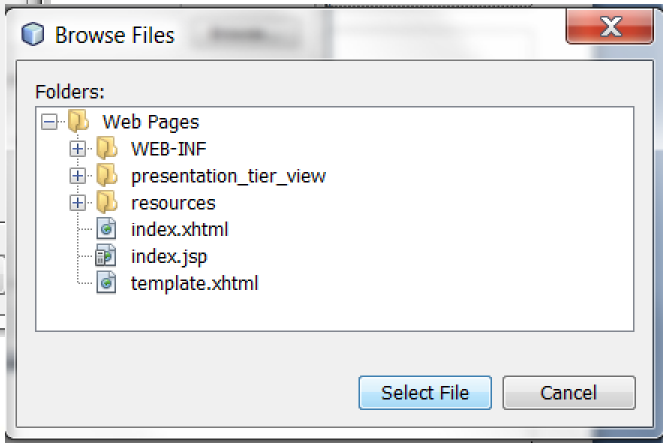
Created File: ıre\Kruczkiewicz\lab5\TLibrary2_EE\TLibrary2_EE-war\web\index2.xhtml

Template:

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

Select a template for which the client will be generated.

Below: choose the proper template



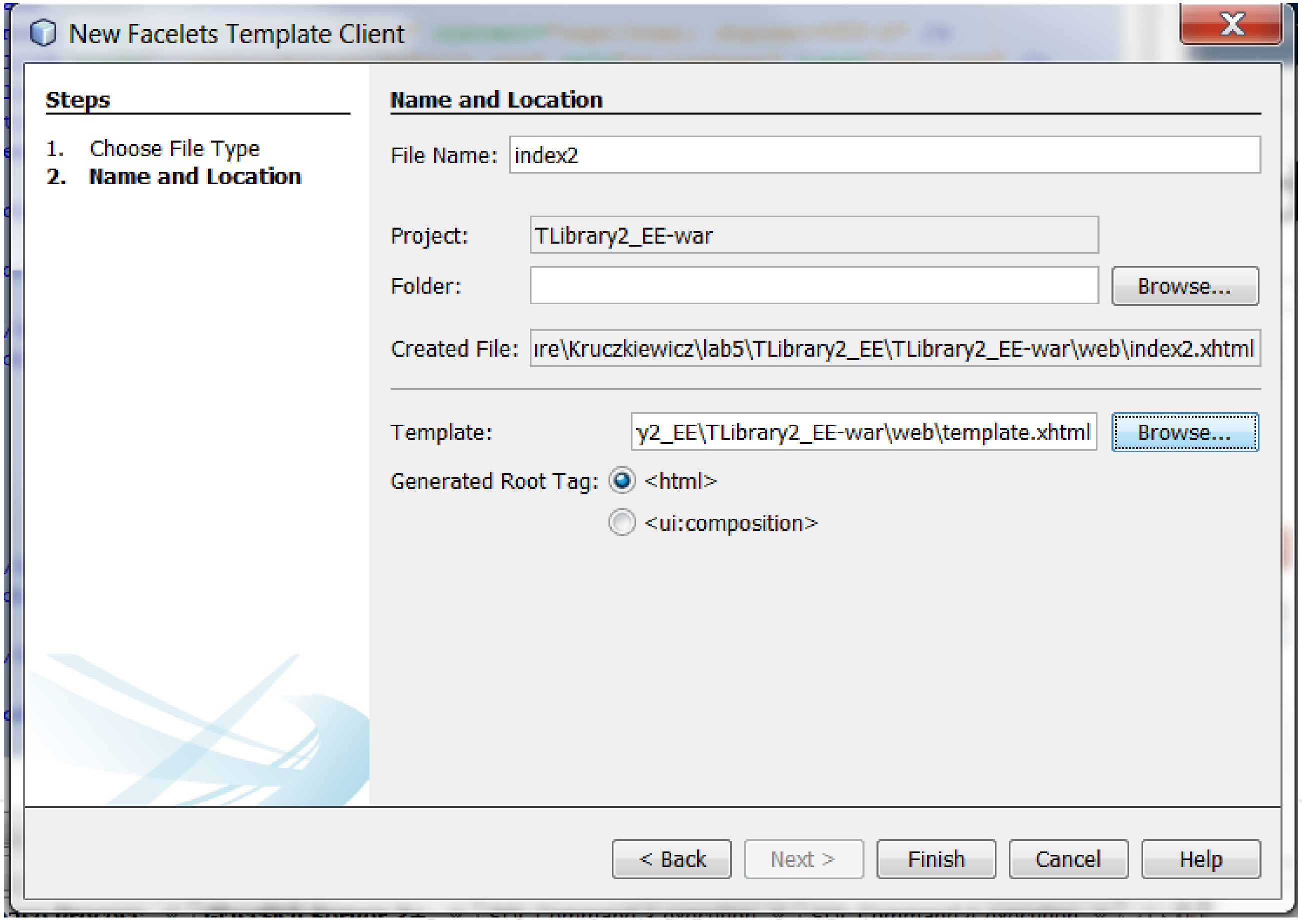
Browse Files

Folders:

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

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

Below: after choosing the proper template



New Facelets Template Client

Steps

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

Name and Location

File Name: index2

Project: TLibrary2_EE-war

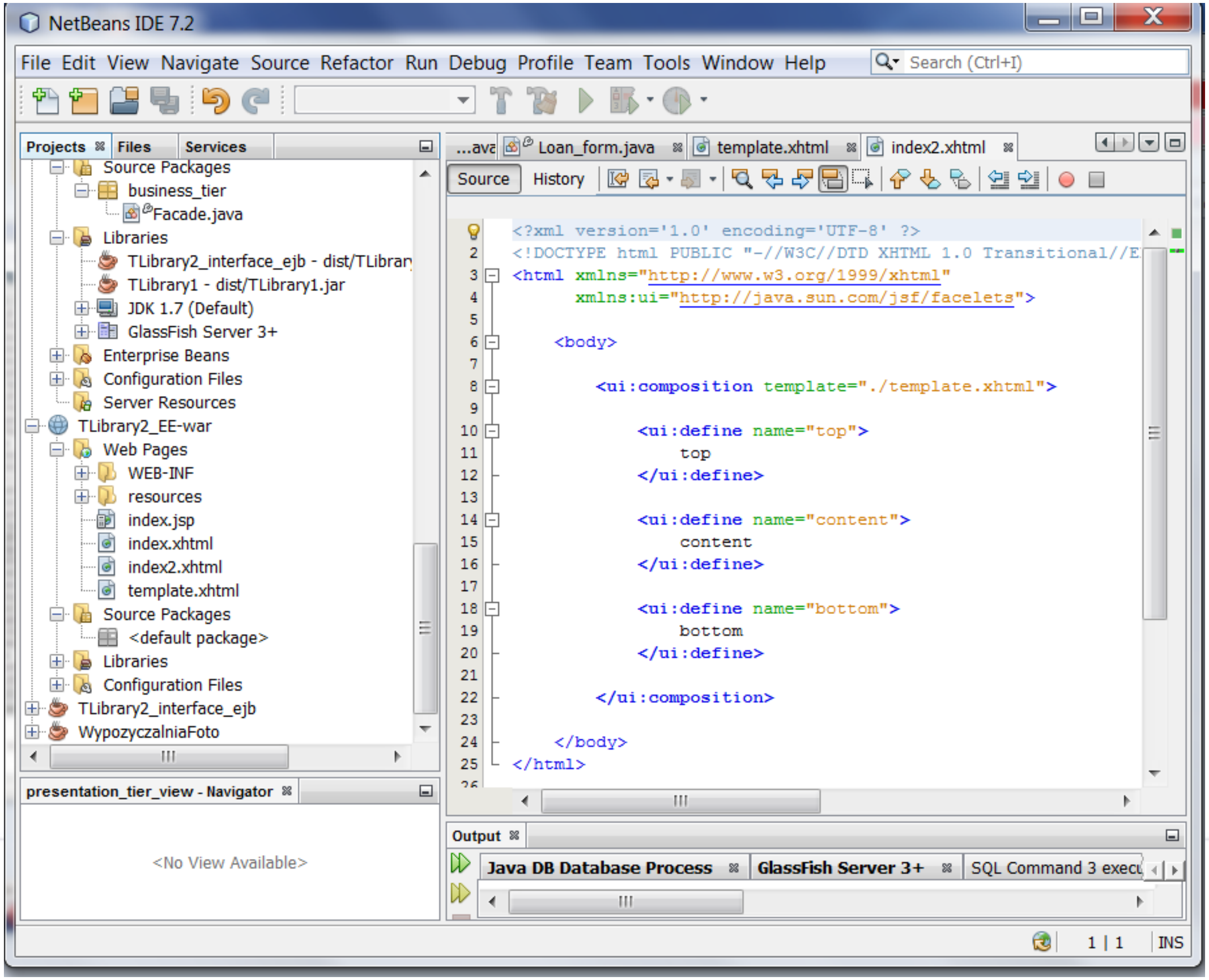
Folder:

Created File: ıre\Kruczkiewicz\lab5\TLibrary2_EE\TLibrary2_EE-war\web\index2.xhtml

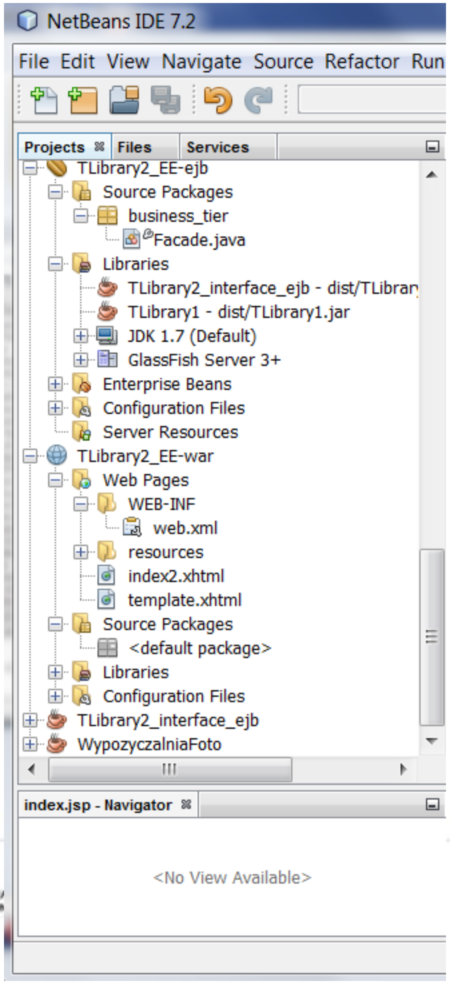
Template: y2_EE\TLibrary2_EE-war\web\template.xhtml

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

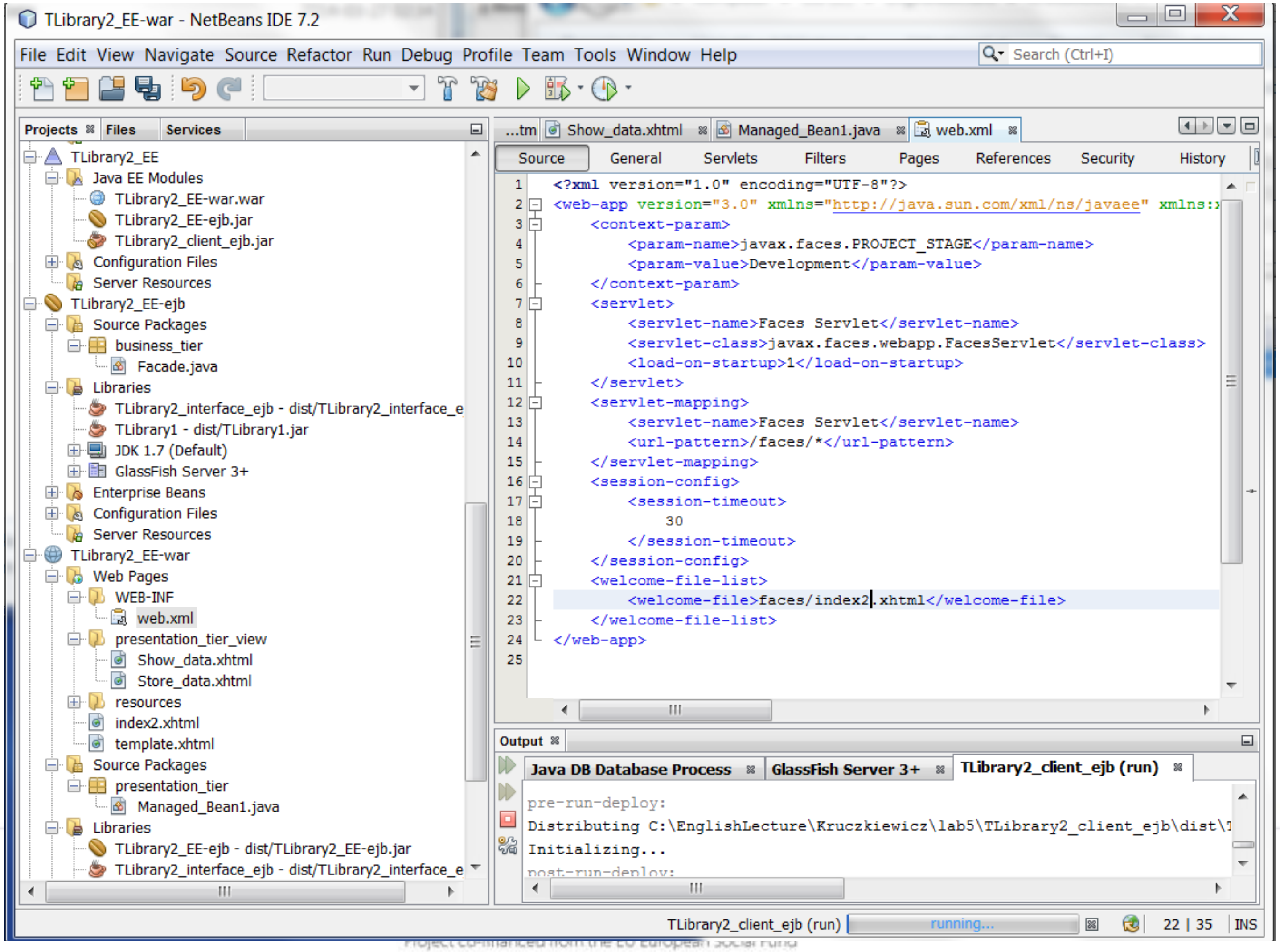
6.7. Creation the main JSF page (index2), based on the previous defined template – the result
At left: the view of the unused JSF anf JSP main pages (index.jsp and index.xhtm);
At right: after deleting unused pages - the index2.xhtml is the main JSF page



```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//E
<html xmlns="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>
```

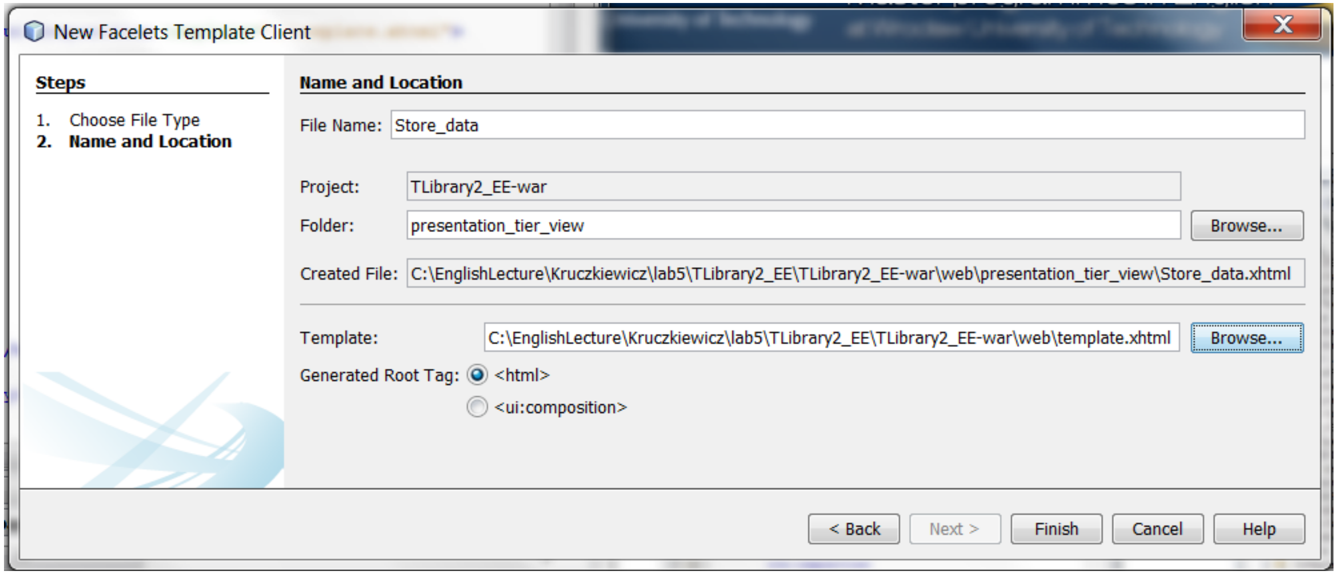


6.8. The set up the main web page as the index2.xtml – in the web.xml file (the descriptor of the web module).


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
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_interface_e
TLibrary1 - dist/TLibrary1.jar
JDK 1.7 (Default)
GlassFish Server 3+
Enterprise Beans
Configuration Files
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_e
...tm Show_data.xhtml Managed_Bean1.java web.xml
Source General Servlets Filters Pages References Security History
Output
Java DB Database Process GlassFish Server 3+ TLibrary2_client_ejb (run)
TLibrary2_client_ejb (run) running... 22 | 35 INS

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:
    <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>
```

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



New Facelets Template Client

Steps

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

Name and Location

File Name: Store_data

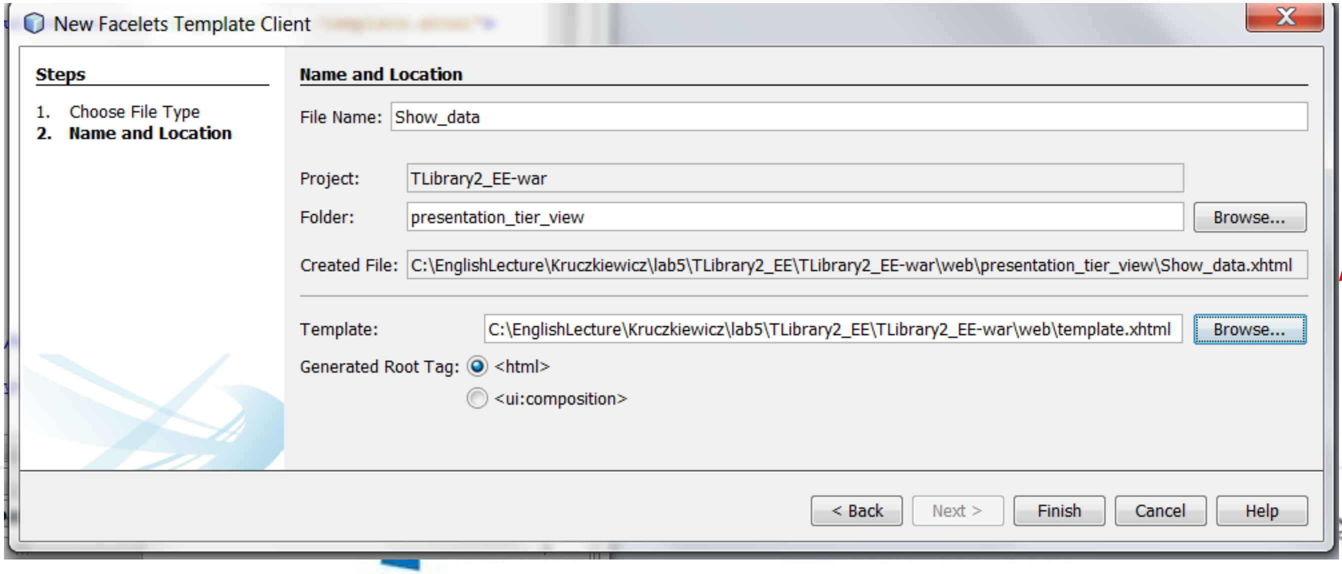
Project: TLibrary2_EE-war

Folder: presentation_tier_view

Created File: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_EE\TLibrary2_EE-war\web\presentation_tier_view\Store_data.xhtml

Template: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_EE\TLibrary2_EE-war\web\template.xhtml

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



New Facelets Template Client

Steps

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

Name and Location

File Name: Show_data

Project: TLibrary2_EE-war

Folder: presentation_tier_view

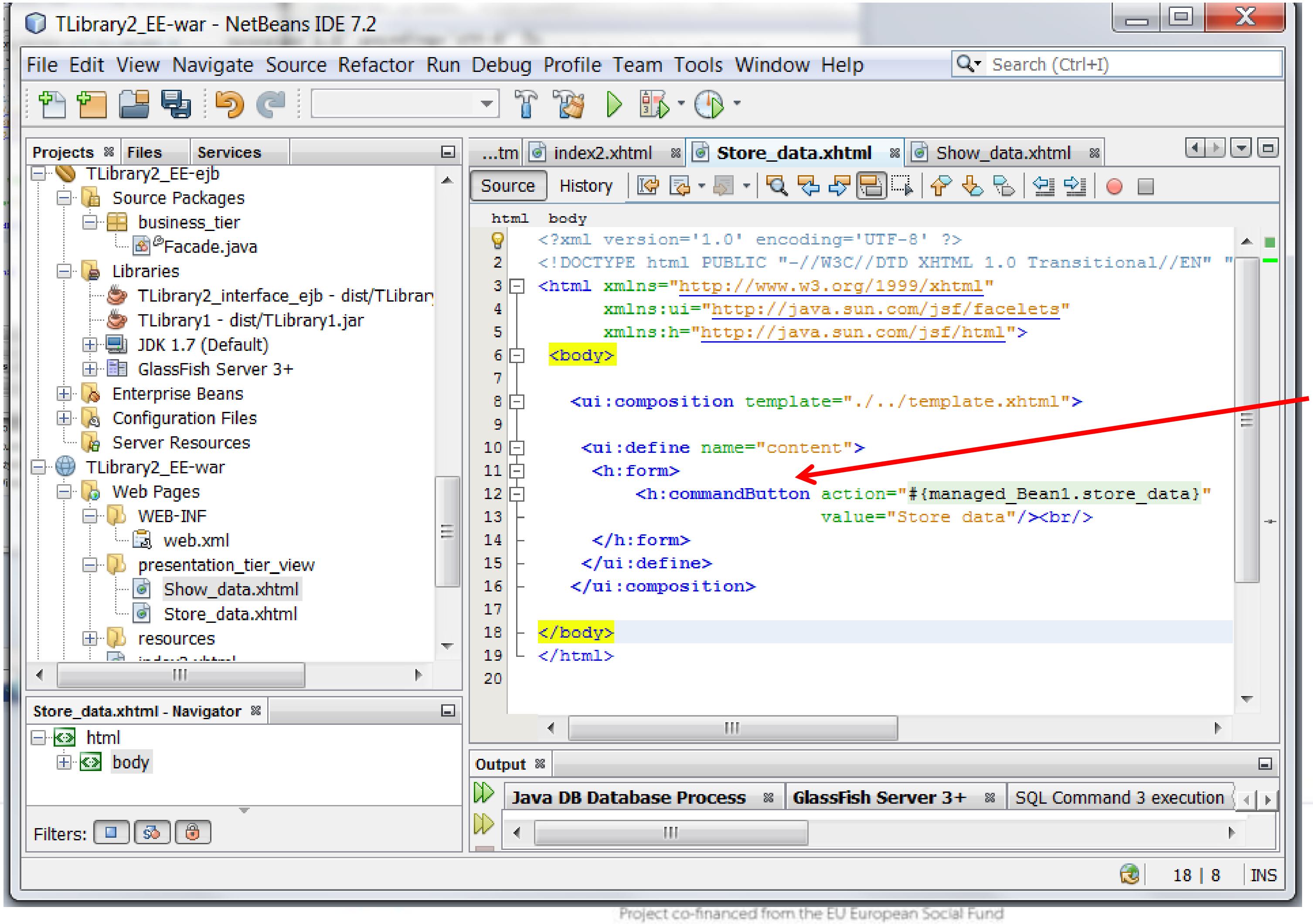
Created File: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_EE\TLibrary2_EE-war\web\presentation_tier_view\Show_data.xhtml

Template: C:\EnglishLecture\Kruczkiewicz\lab5\TLibrary2_EE\TLibrary2_EE-war\web\template.xhtml

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

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



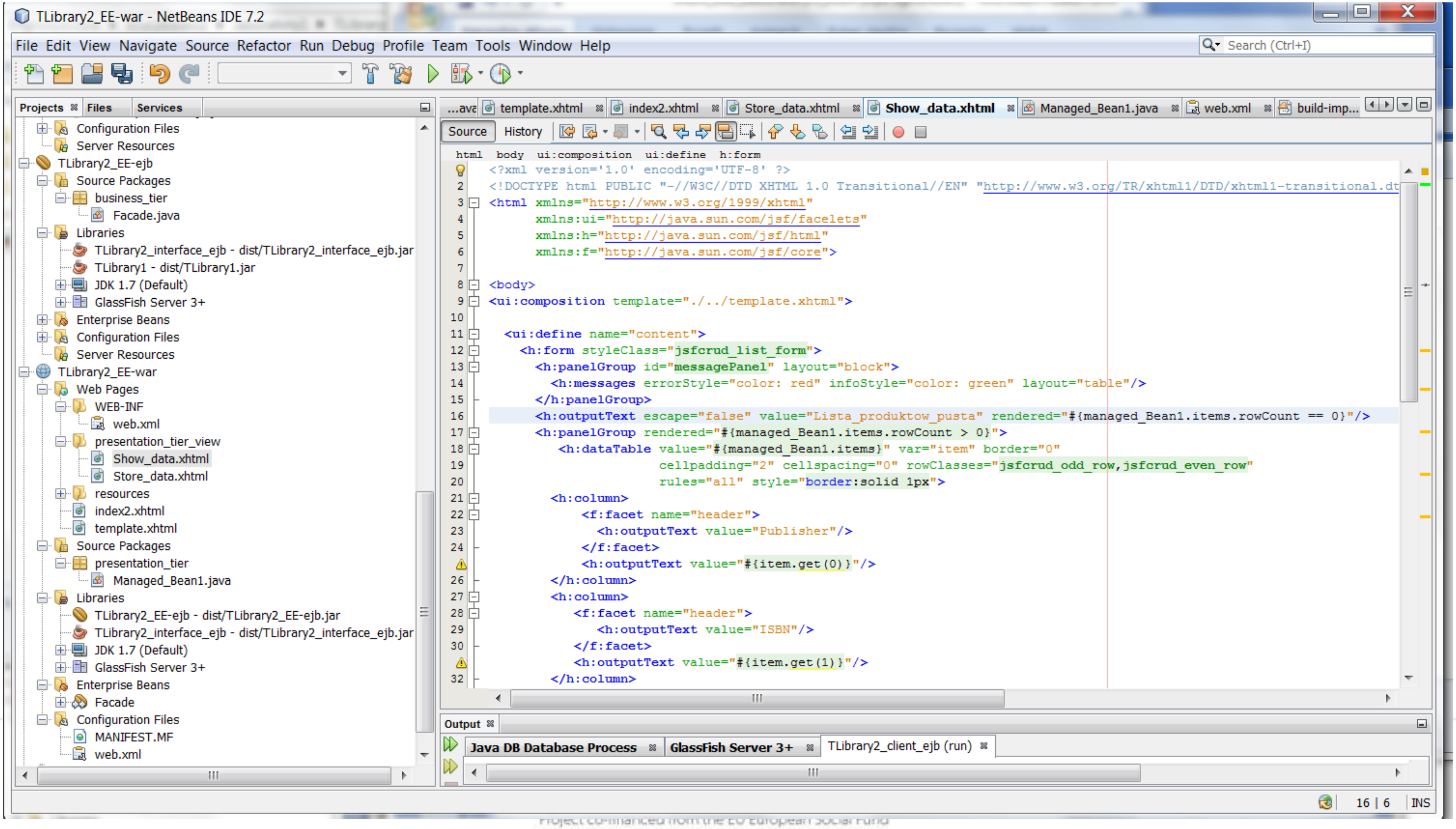
```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html">
 <body>

   <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>
```

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.

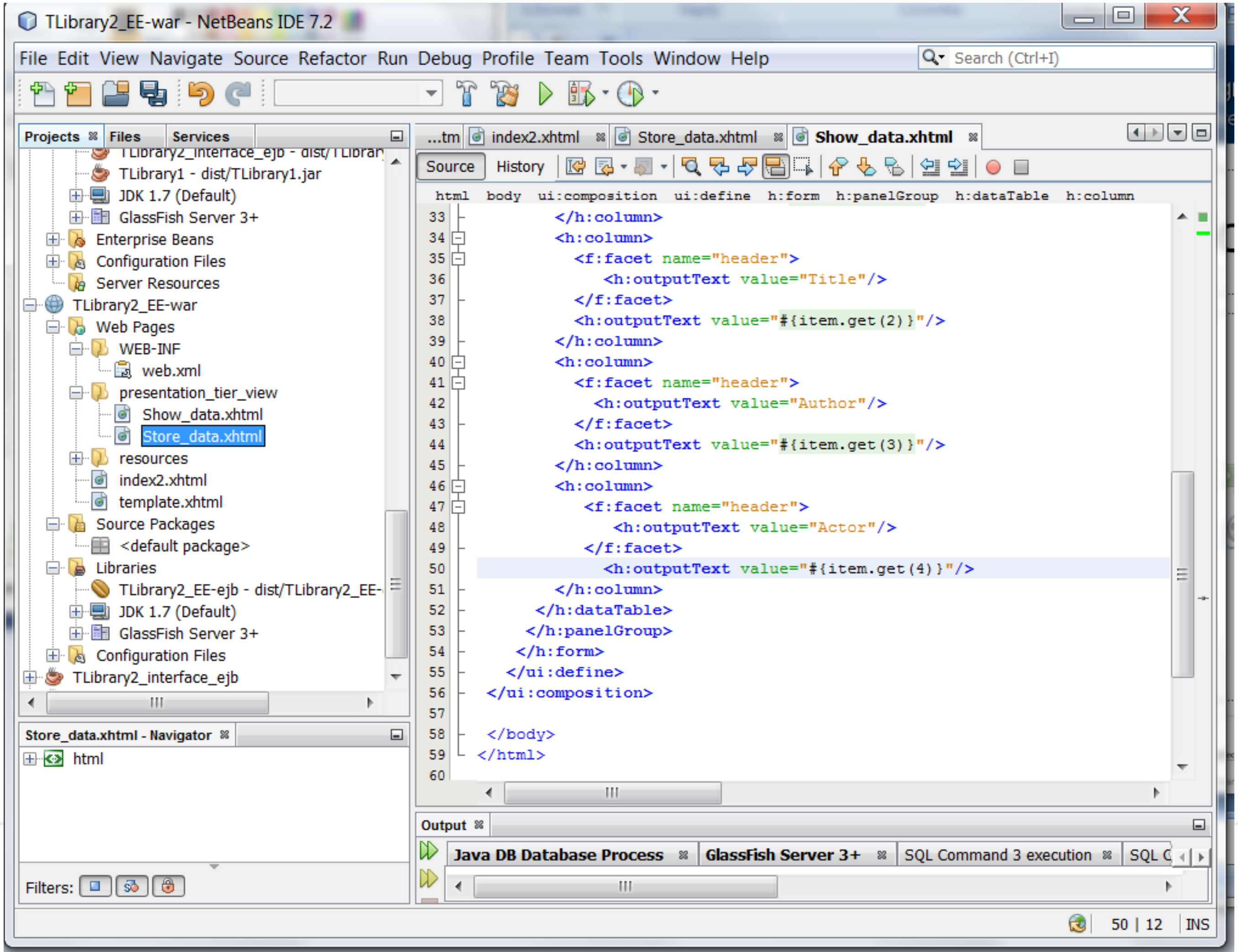
```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
  <html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:ui="http://java.sun.com/jsf/facelets"
   xmlns:h="http://java.sun.com/jsf/html">

<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



```
<?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.dt
<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">
        <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>
```

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



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

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">
```

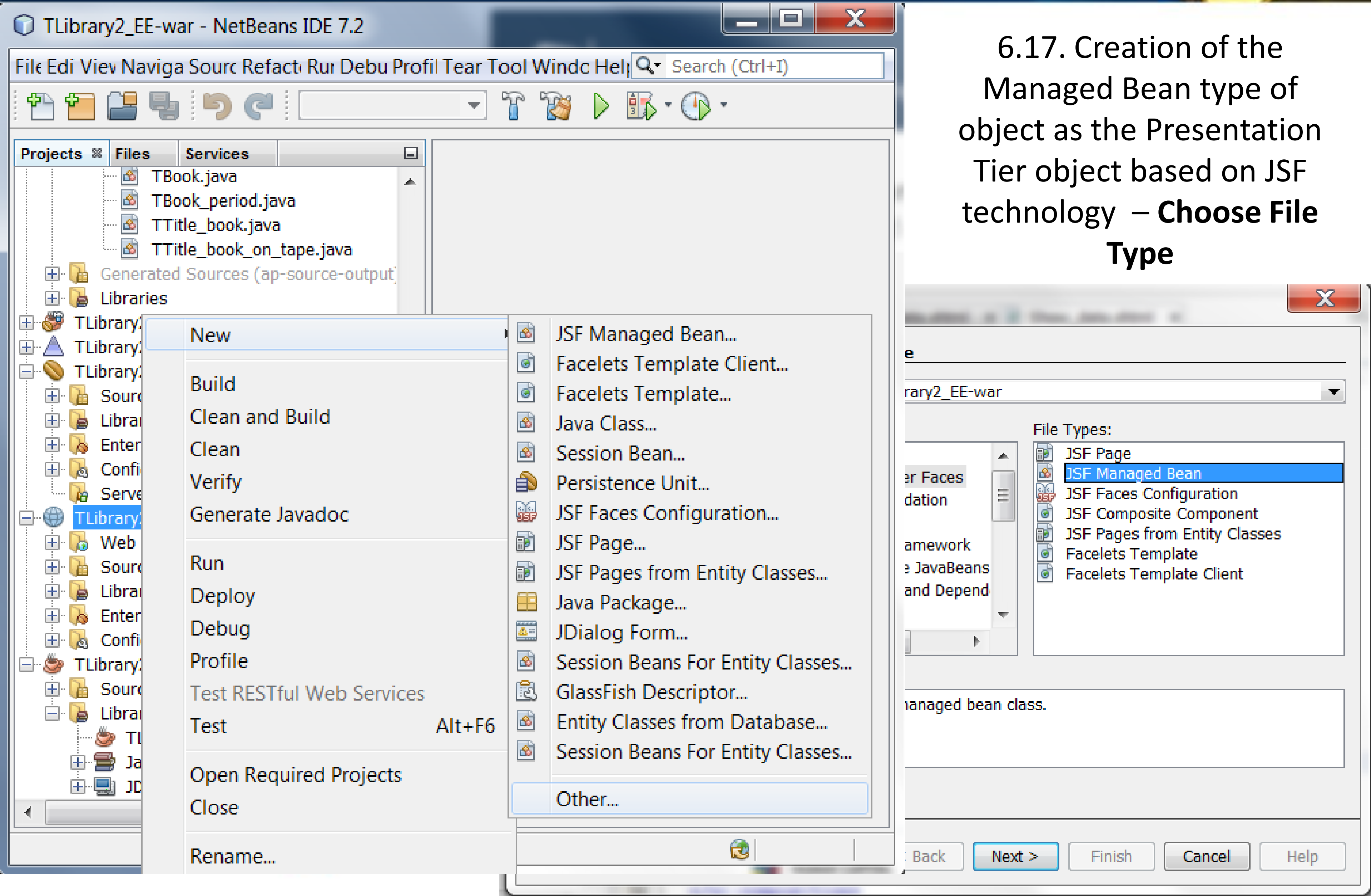
6.15. The code of the second JSF page for displaying data from the database - continuation

```
<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>
```

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

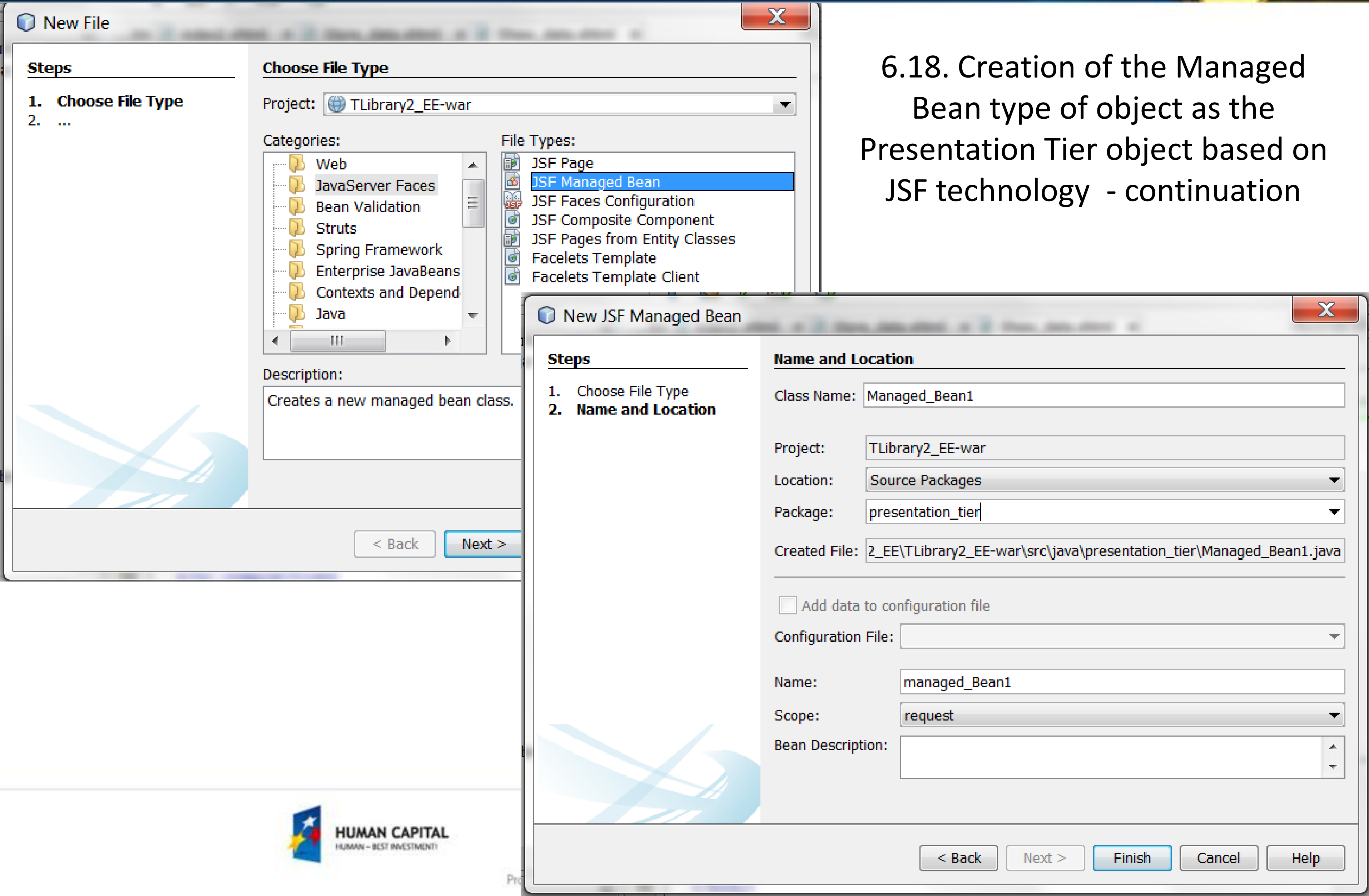


TLibrary2_EE-war - NetBeans IDE 7.2

File Types: JSF Managed Bean

…anaged bean class.

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



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

Description:

Creates a new managed bean class.

< Back | Next >

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:

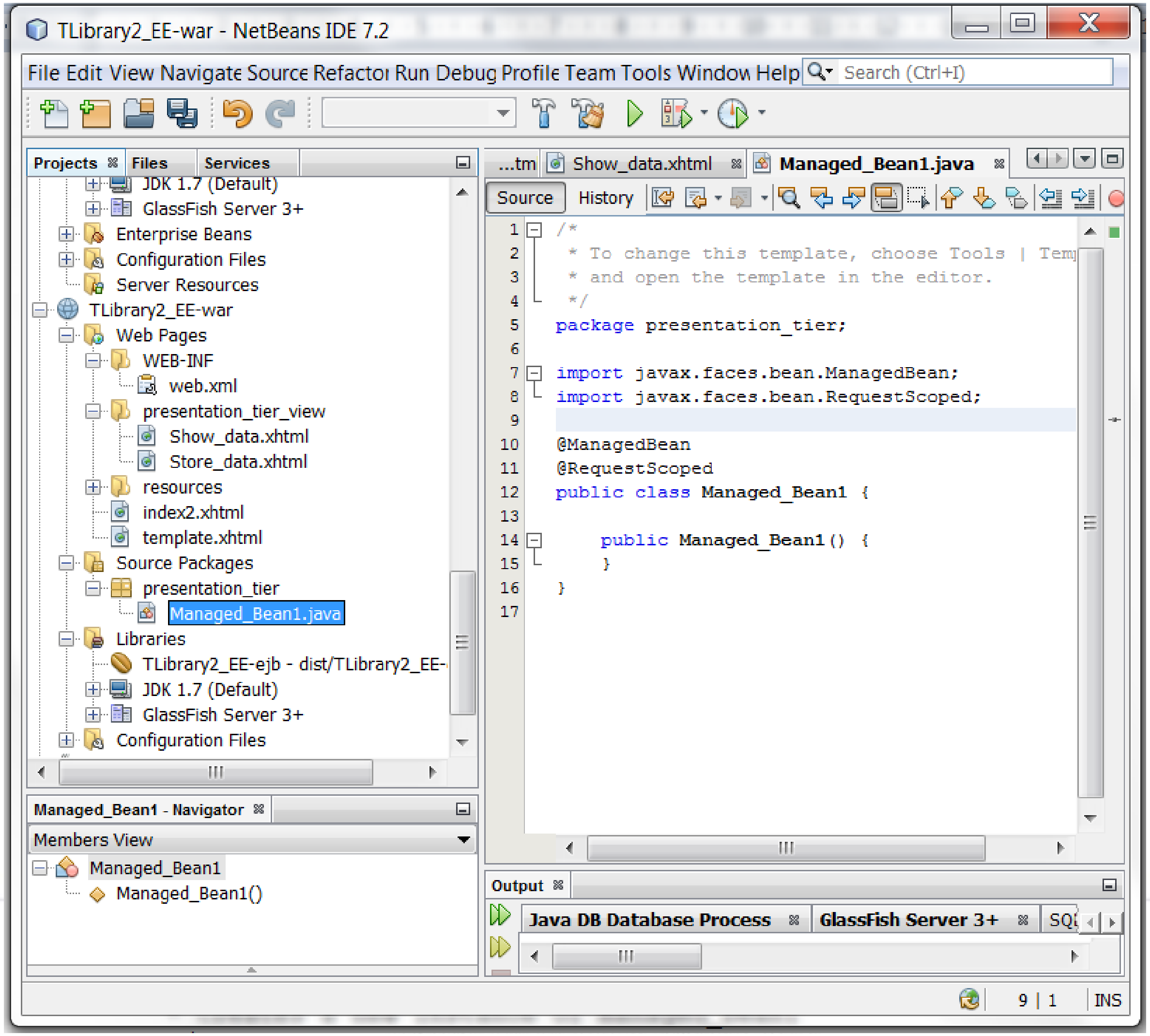
Name: managed_Bean1

Scope: request

Bean Description:

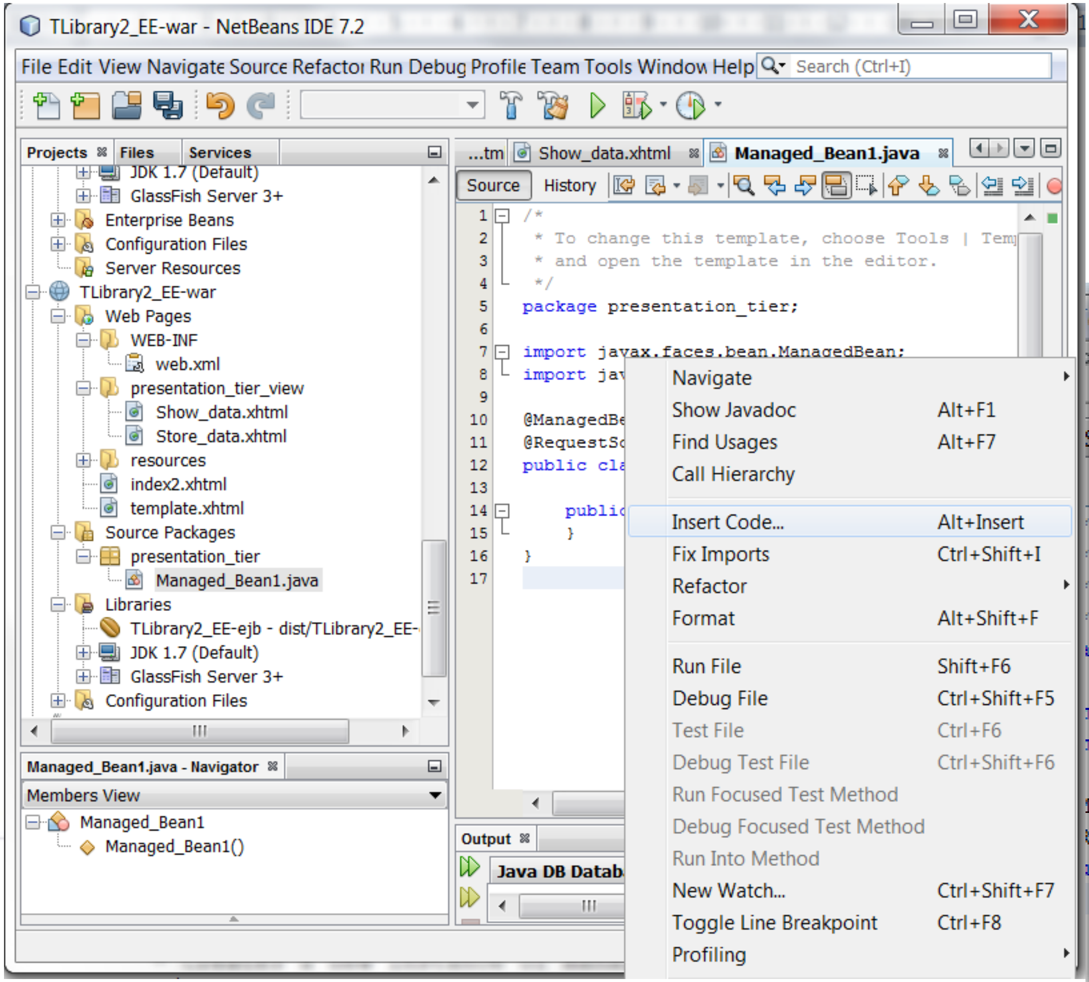
< Back | Next > | Finish | Cancel | Help

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

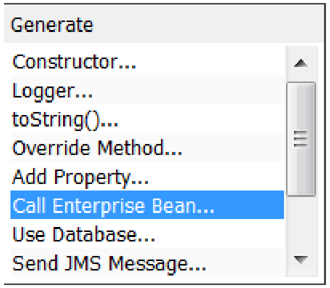


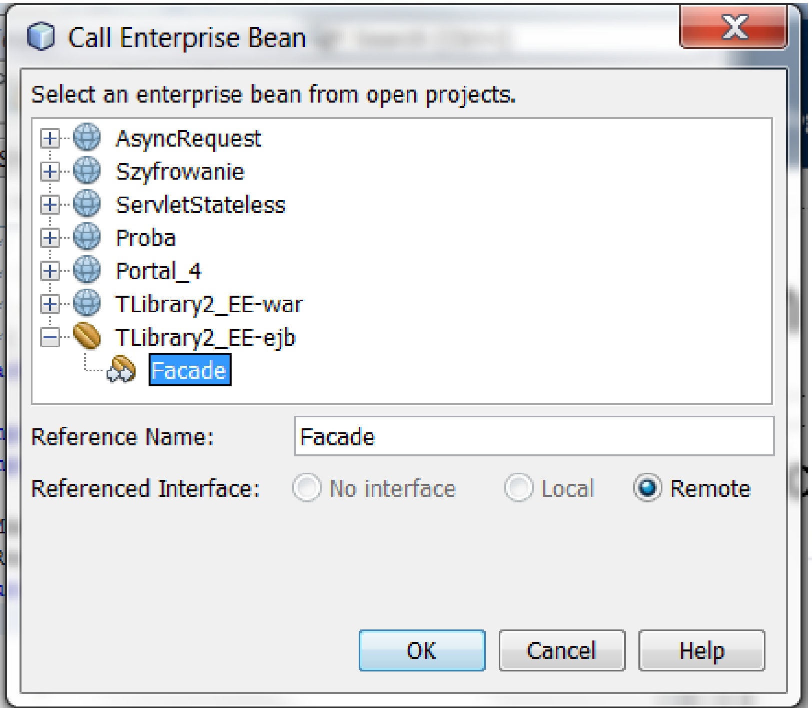
```java
/*
 * To change this template, choose Tools | Tem
 * and open the template in the editor.
 */
package presentation_tier;

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

@ManagedBean
@RequestScoped
public class Managed_Bean1 {

    public Managed_Bean1() {
    }
}
```

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).



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

import javax.faces.bean.ManagedBean;
import jav

@ManagedBe
@RequestSc
public cla

    public
    }
}
```



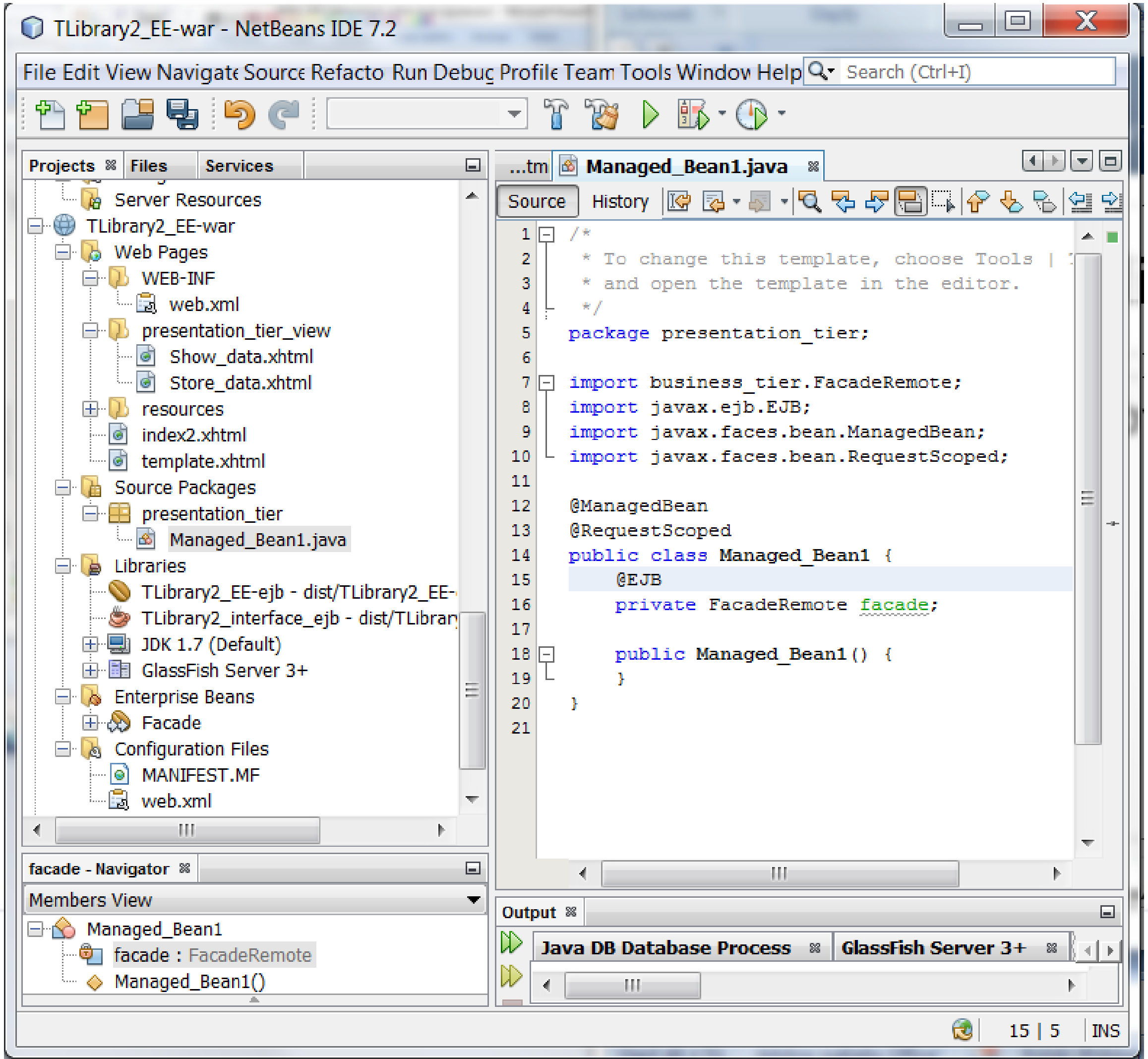


Select an enterprise bean from open projects.

Reference Name: Facade

Referenced Interface: ○ No interface ○ Local ● Remote

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



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

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

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

    public Managed_Bean1() {
    }
}
```

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;
     }
  }
```

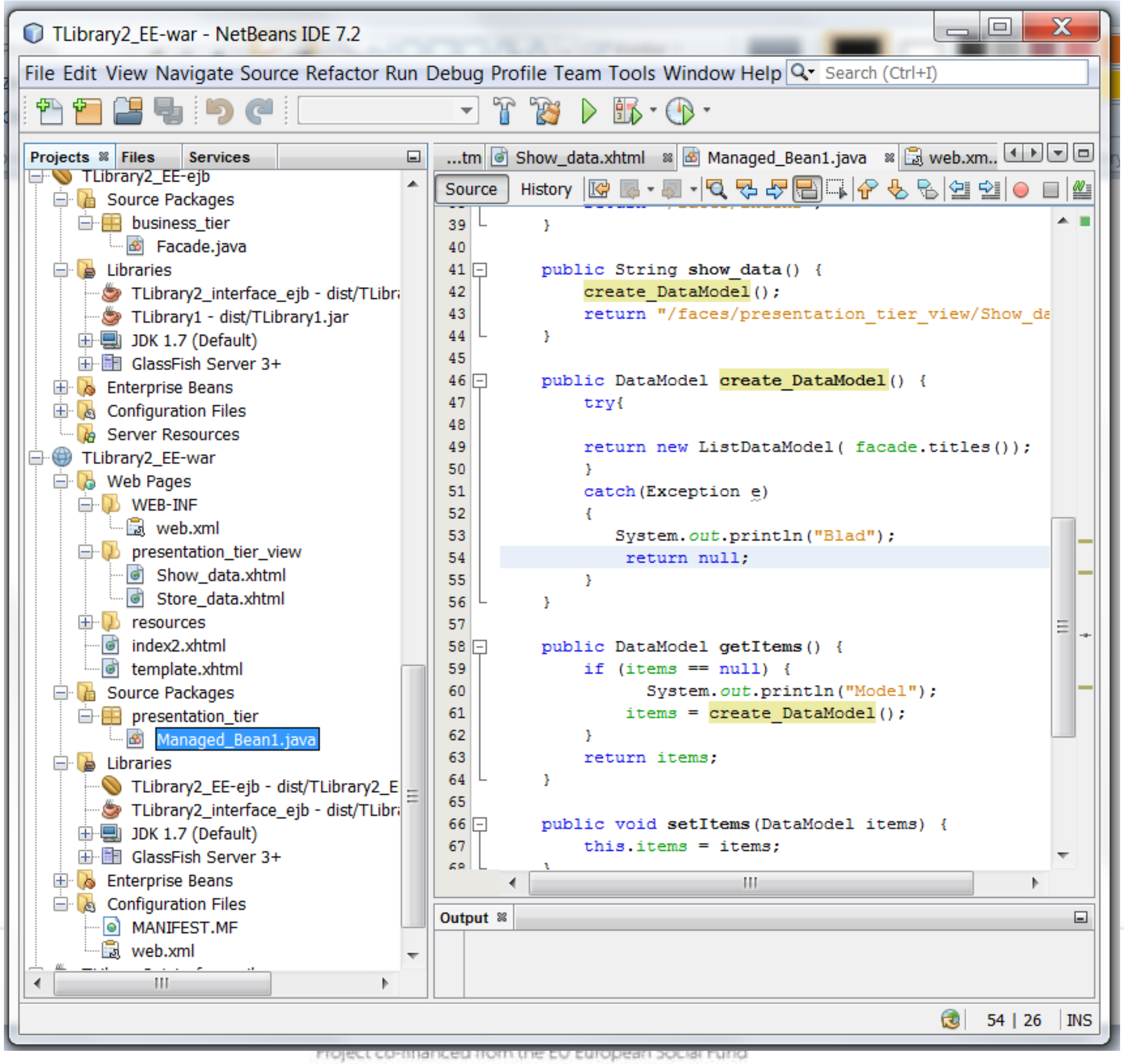
The **create_DataModel** method for creation the new data model for h:DataTable component – this based on data returned from the titles method of the Facade EJB and they are read from database by using ORM mechanism (p. 24, 4-7)

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

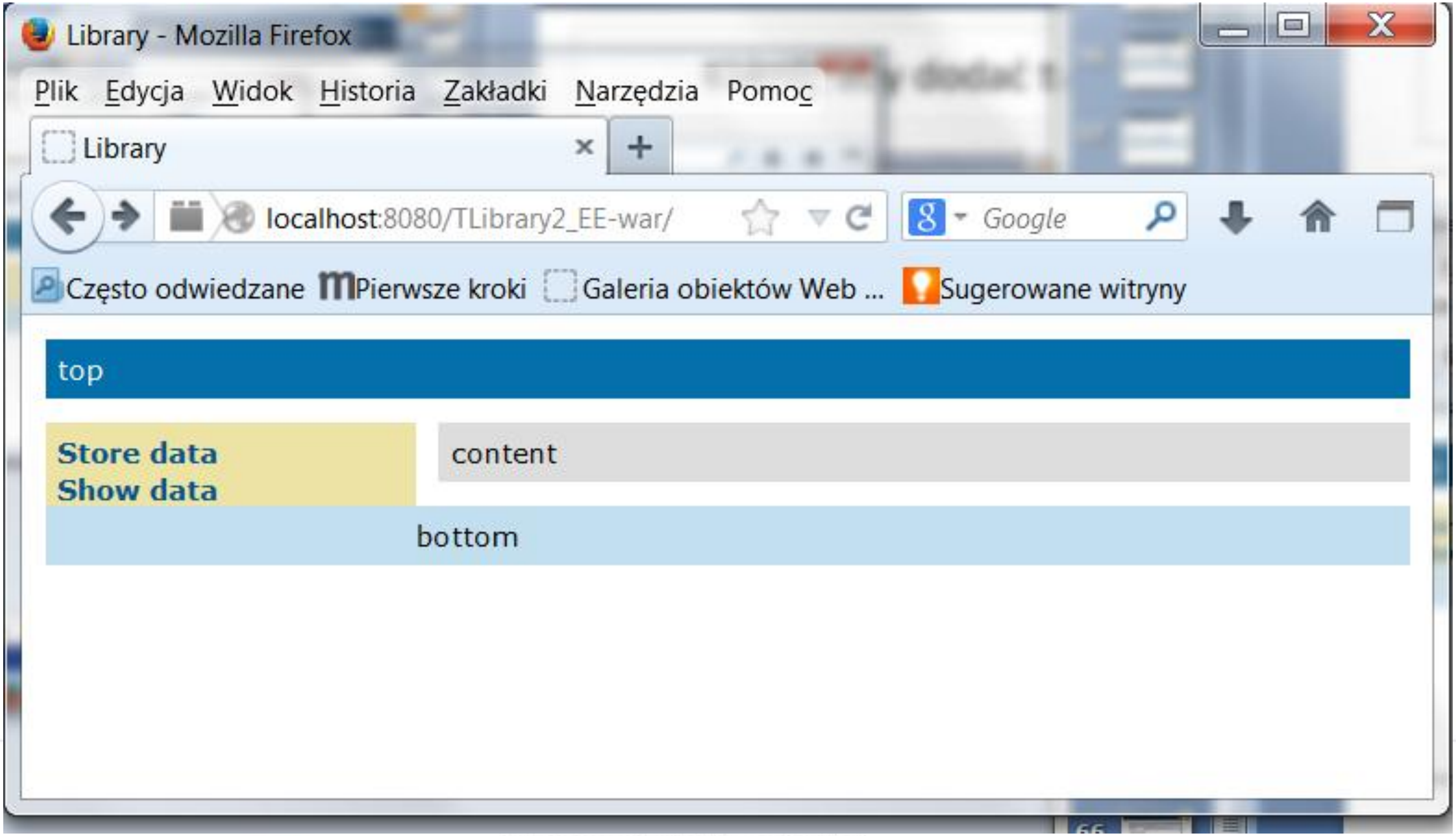


```java
    }

    public String show_data() {
        create_DataModel();
        return "/faces/presentation_tier_view/Show_da
    }

    public DataModel create_DataModel() {
        try{

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

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

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

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**

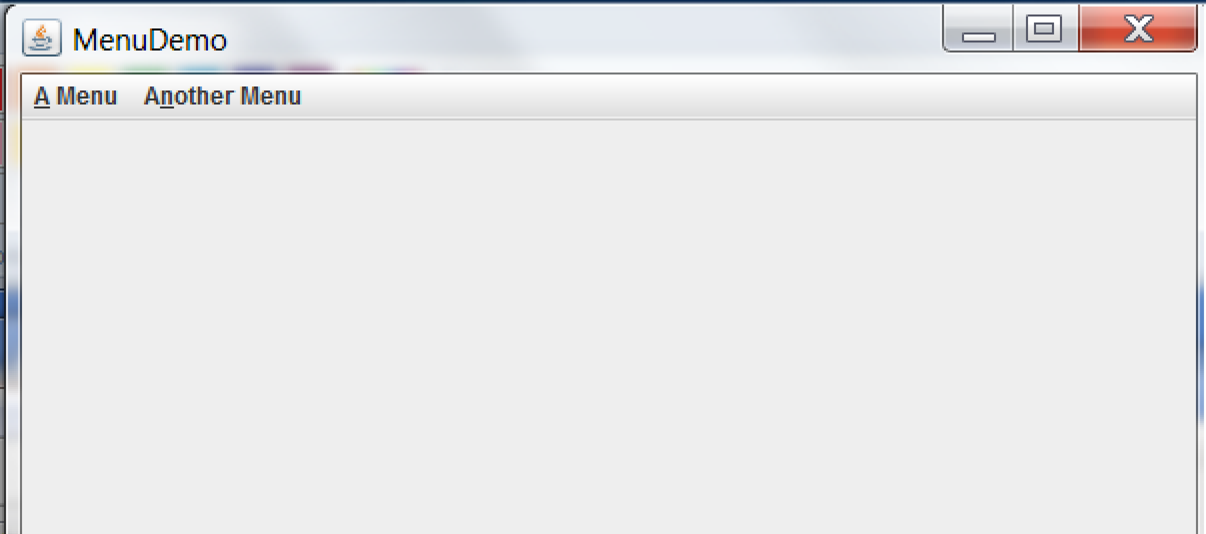

Library - Mozilla Firefox
Plik Edycja Widok Historia Zakładki Narzędzia Pomoc
Library
localhost:8080/TLibrary2_EE-war/
Google
Często odwiedzane Pierwsze kroki Galeria obiektów Web ... Sugerowane witryny

top

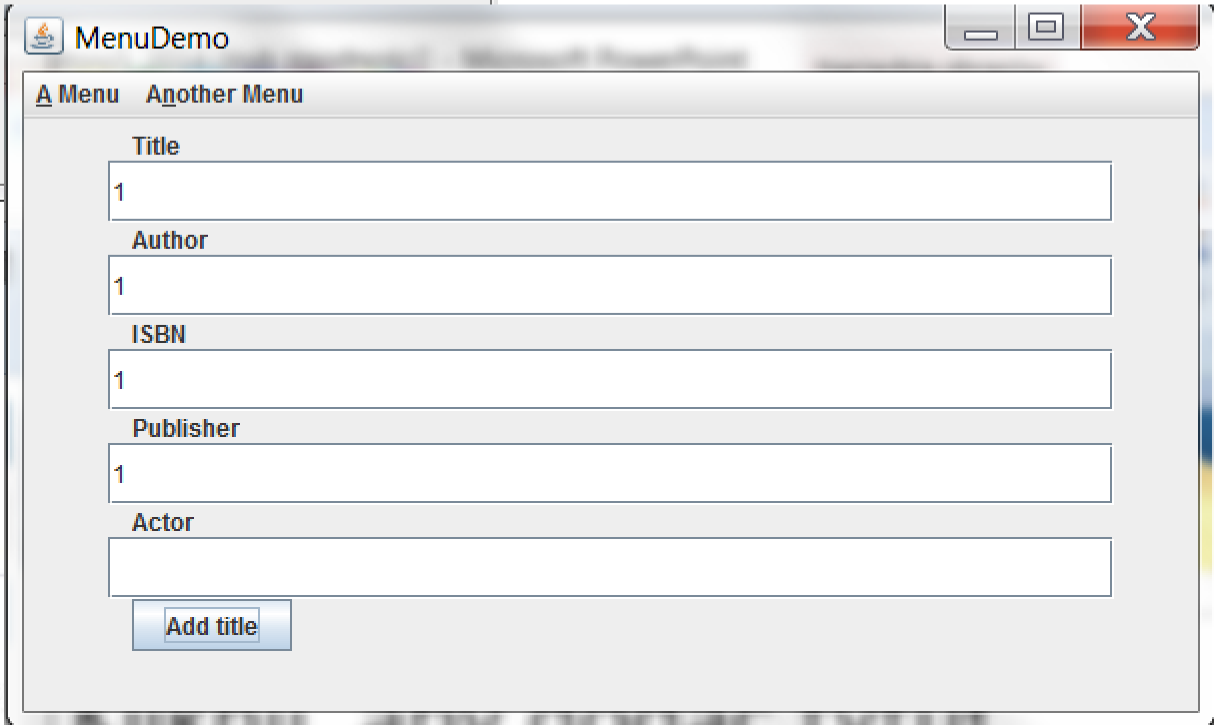
Store data
Show data

content

bottom



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.



Title: 1

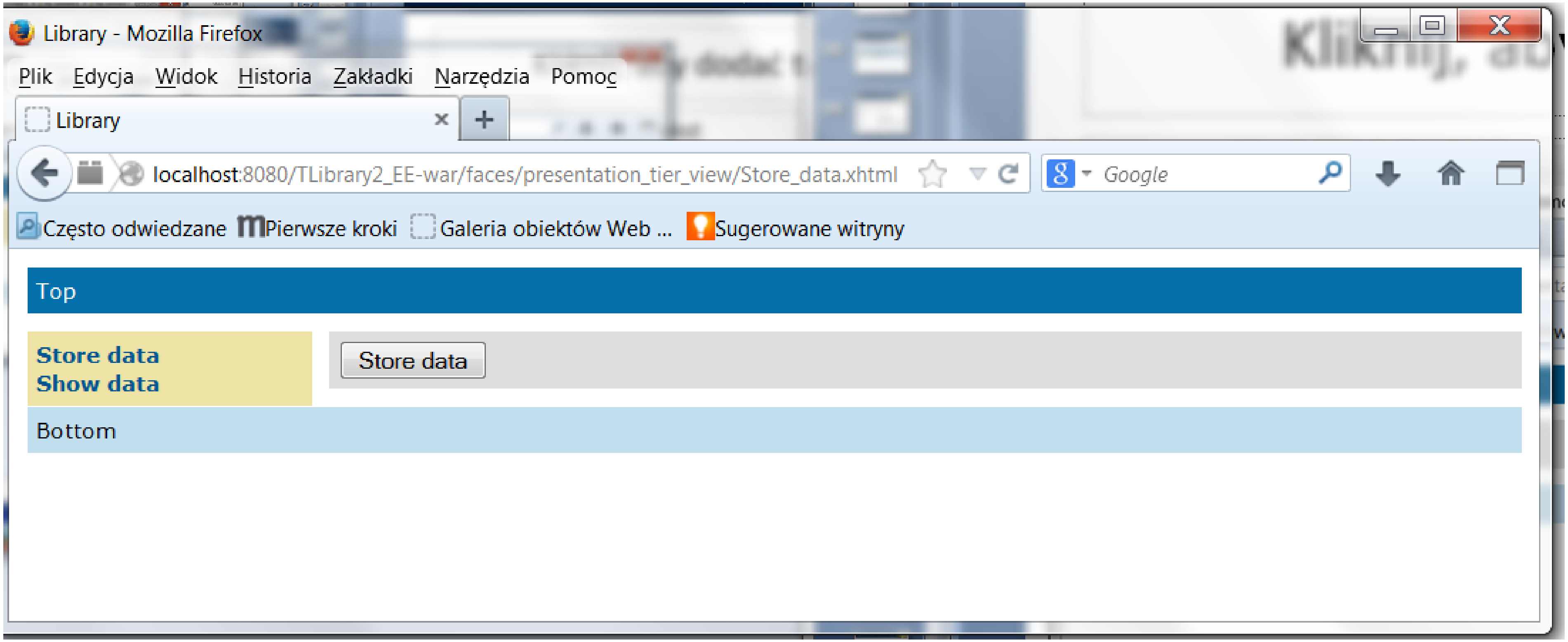
Author: 1

ISBN: 1

Publisher: 1

Actor:

7.3. The view after choose the Store data link (the left part of page) of 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/Store_data.xhtml

Google

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

Top

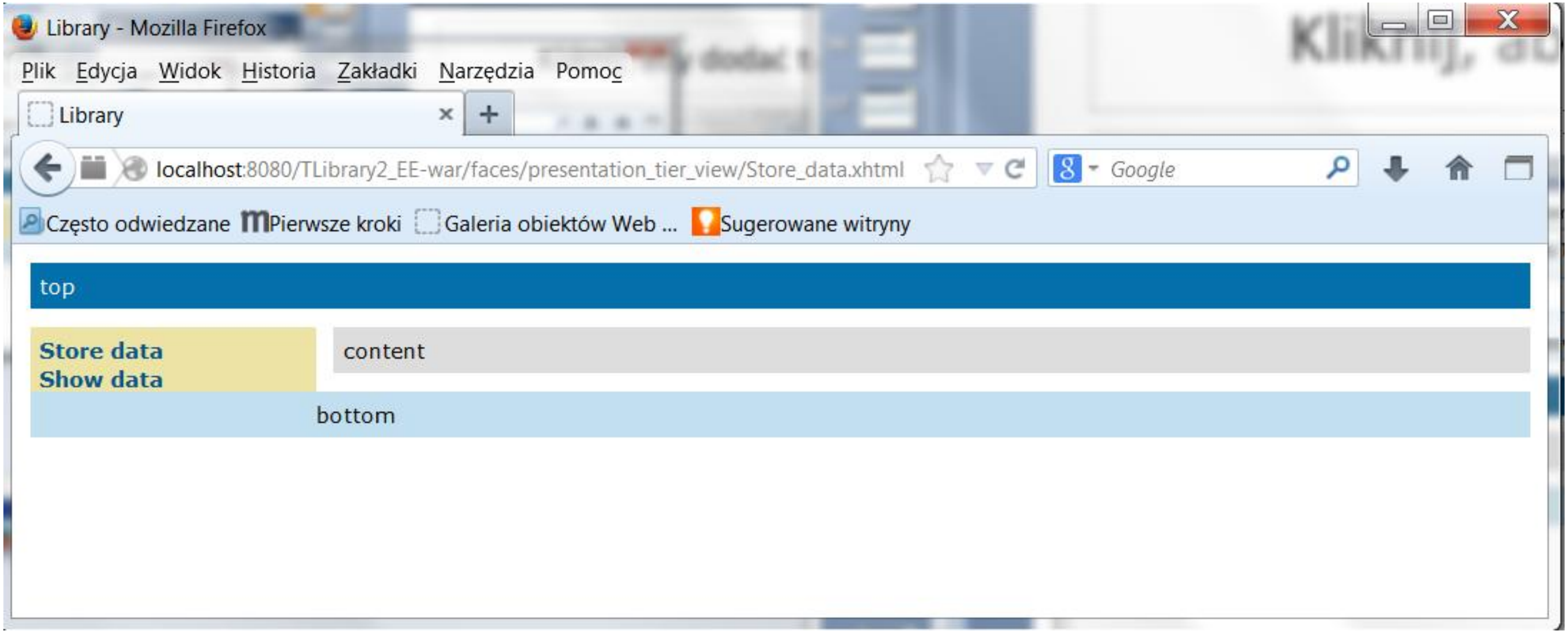
Store data
Show data

Store data

Bottom

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

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



localhost:8080/TLibrary2_EE-war/faces/presentation_tier_view/Store_data.xhtml

top

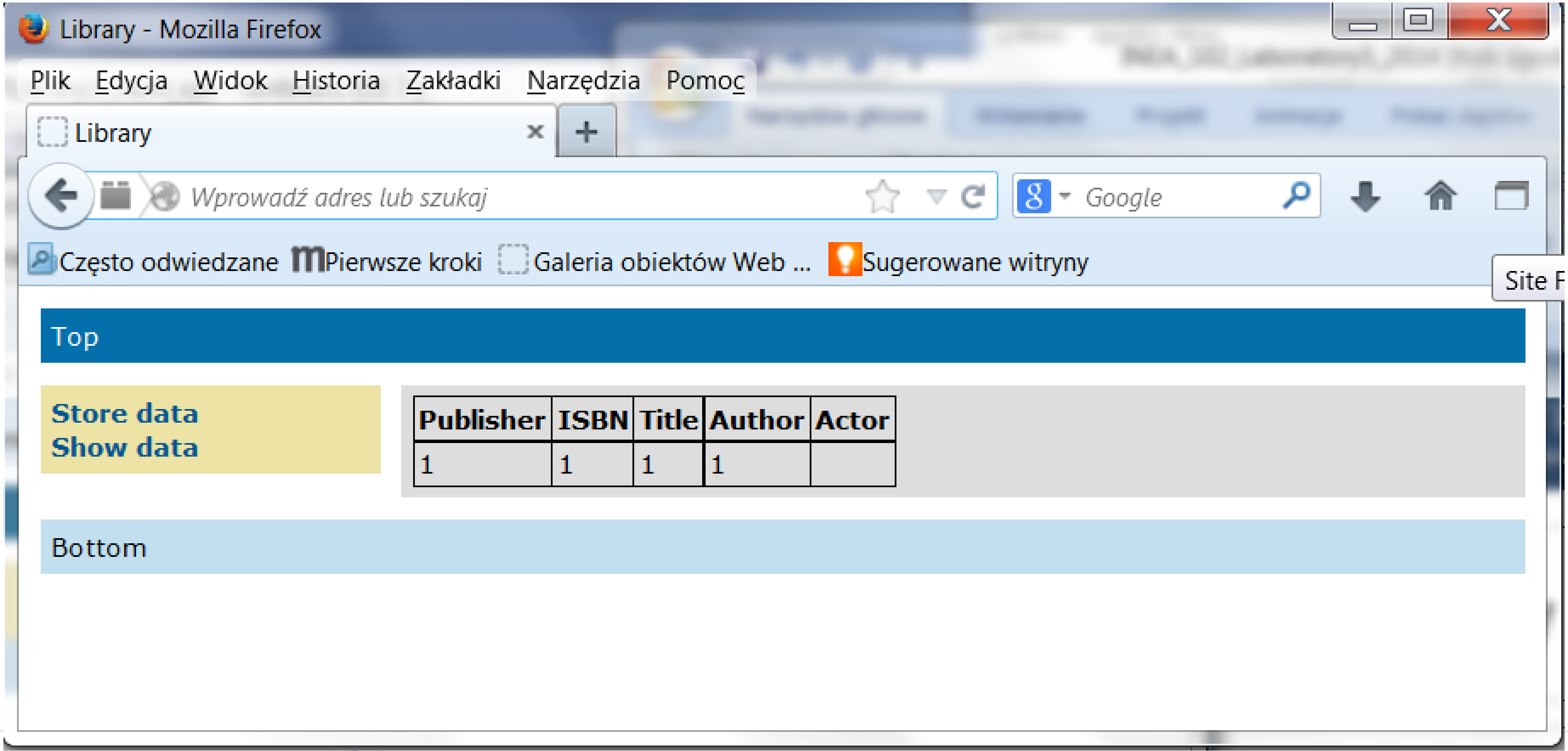
Store data
Show data

content

bottom

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

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
Wprowadź adres lub szukaj
Google
Często odwiedzane
Pierwsze kroki
Galeria obiektów Web ...
Sugerowane witryny
Site F

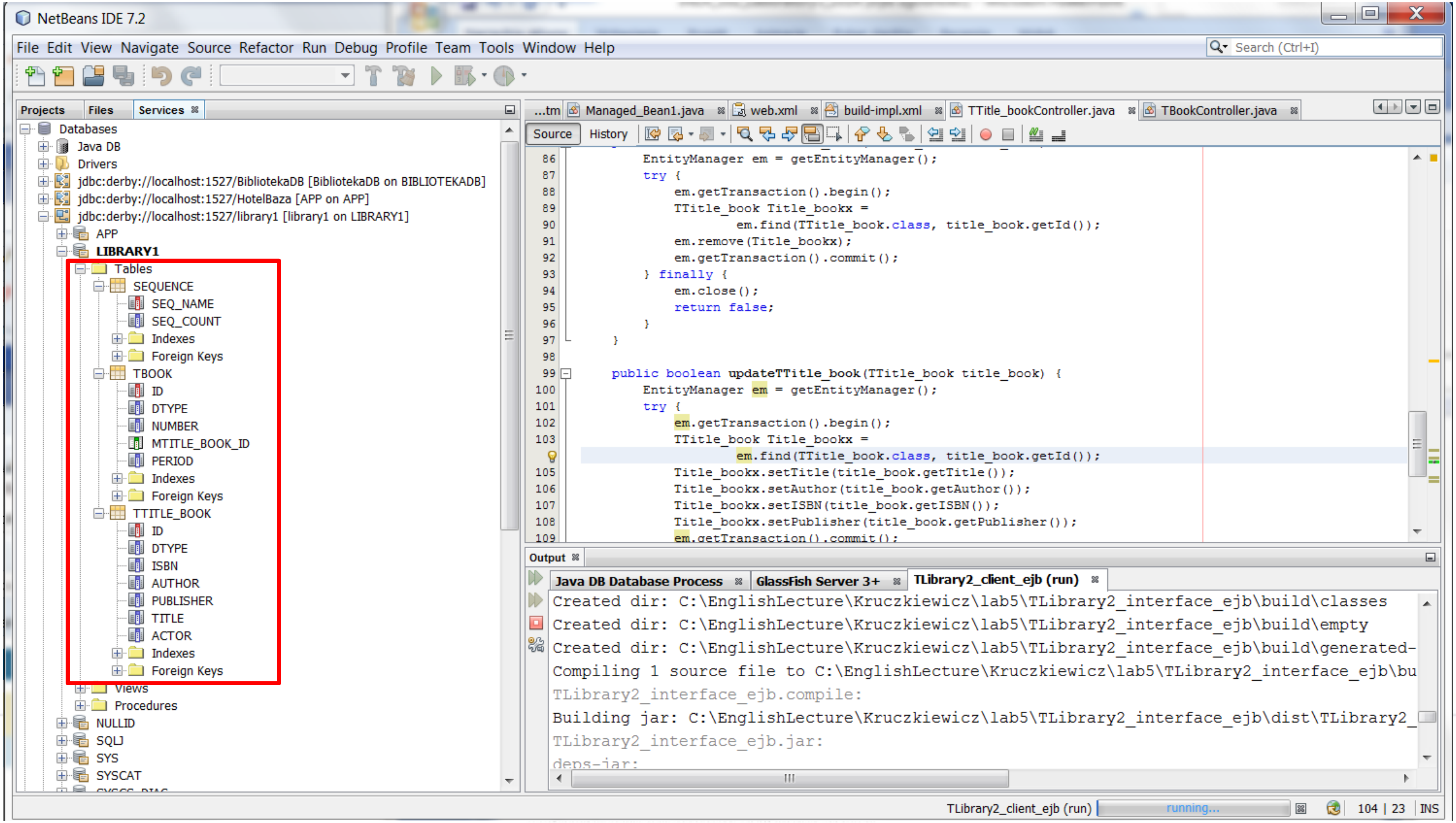
Top

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)

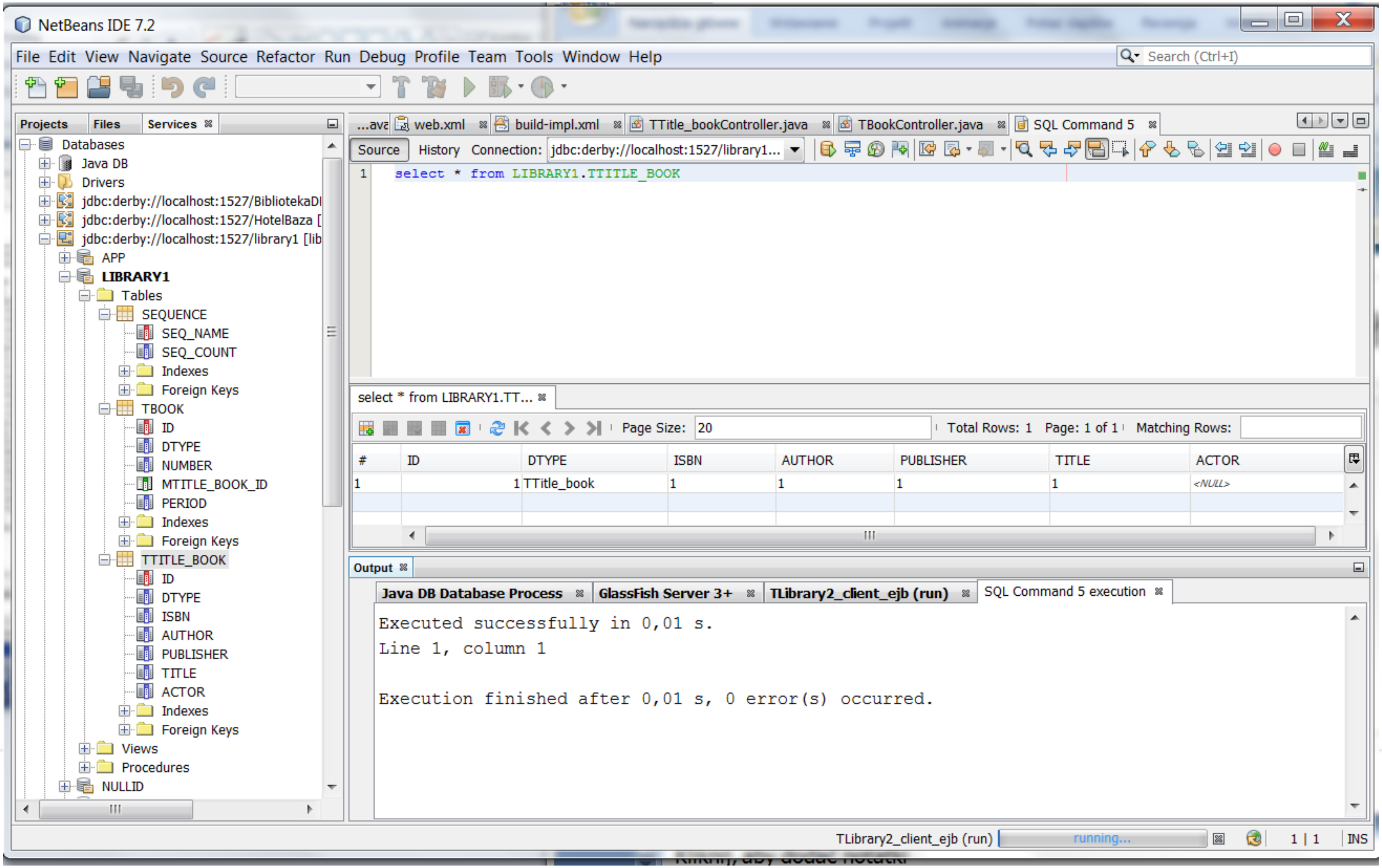


```
Databases
  Java DB
  Drivers
  jdbc:derby://localhost:1527/BibliotekaDB [BibliotekaDB on BIBLIOTEKADB]
  jdbc:derby://localhost:1527/HotelBaza [APP on APP]
  jdbc:derby://localhost:1527/library1 [library1 on LIBRARY1]
    APP
    LIBRARY1
      Tables
        SEQUENCE
          SEQ_NAME
          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
    SQLJ
    SYS
    SYSCAT
```

```java
        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();
```

```
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)

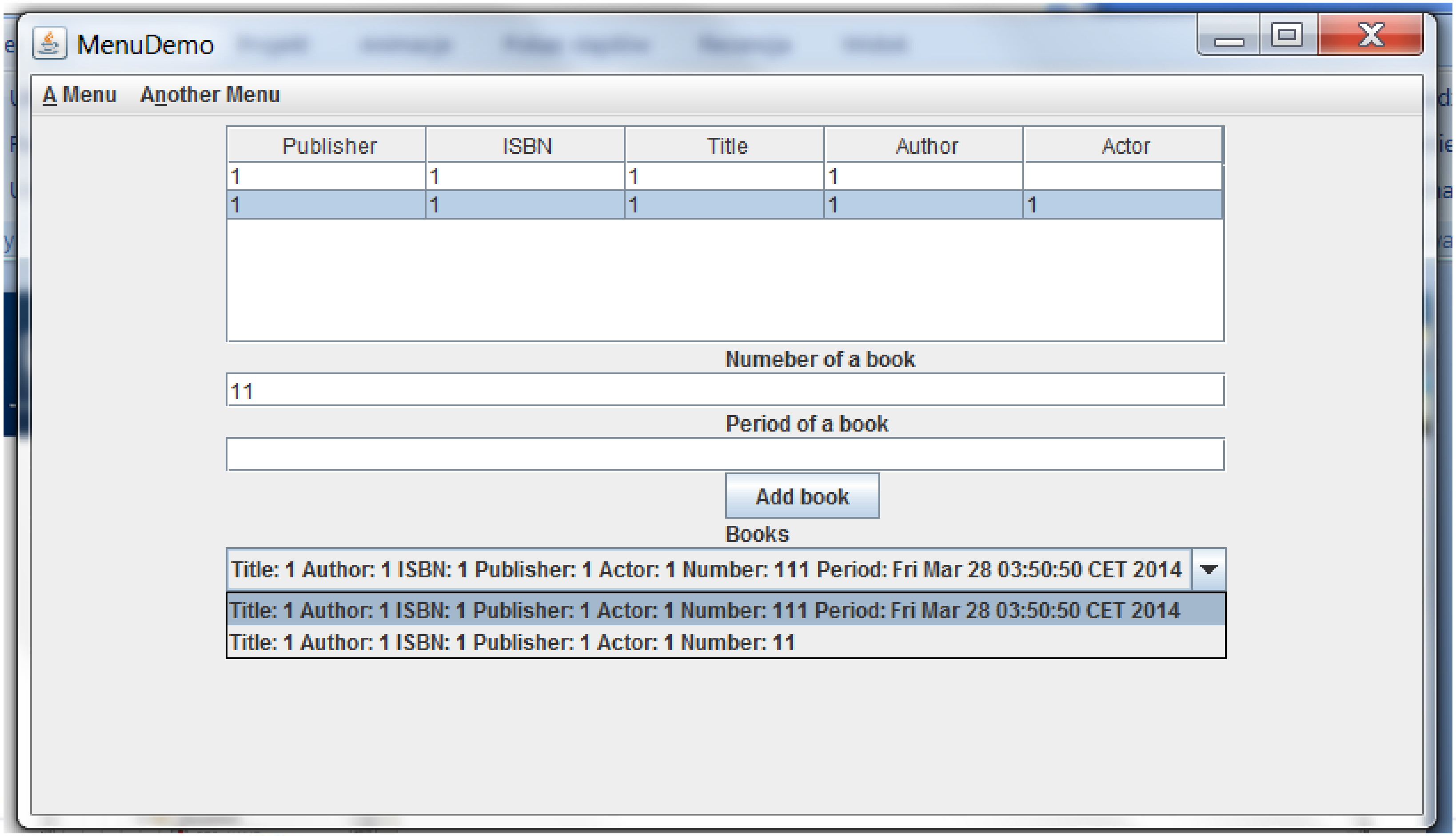


```
select * from LIBRARY1.TTITLE_BOOK
```

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

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

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

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



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

Numeber of a book

11

Period of a book

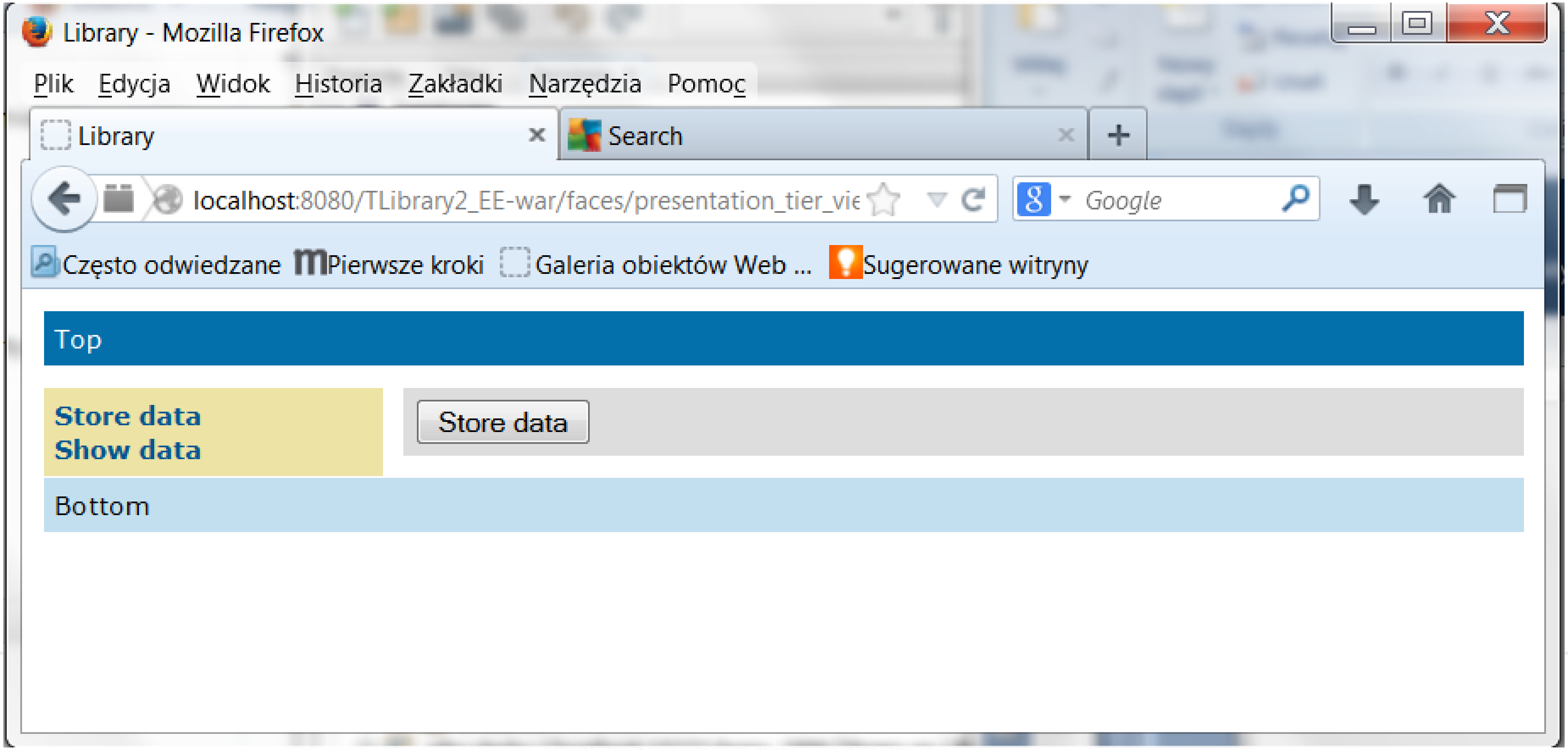
Books

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

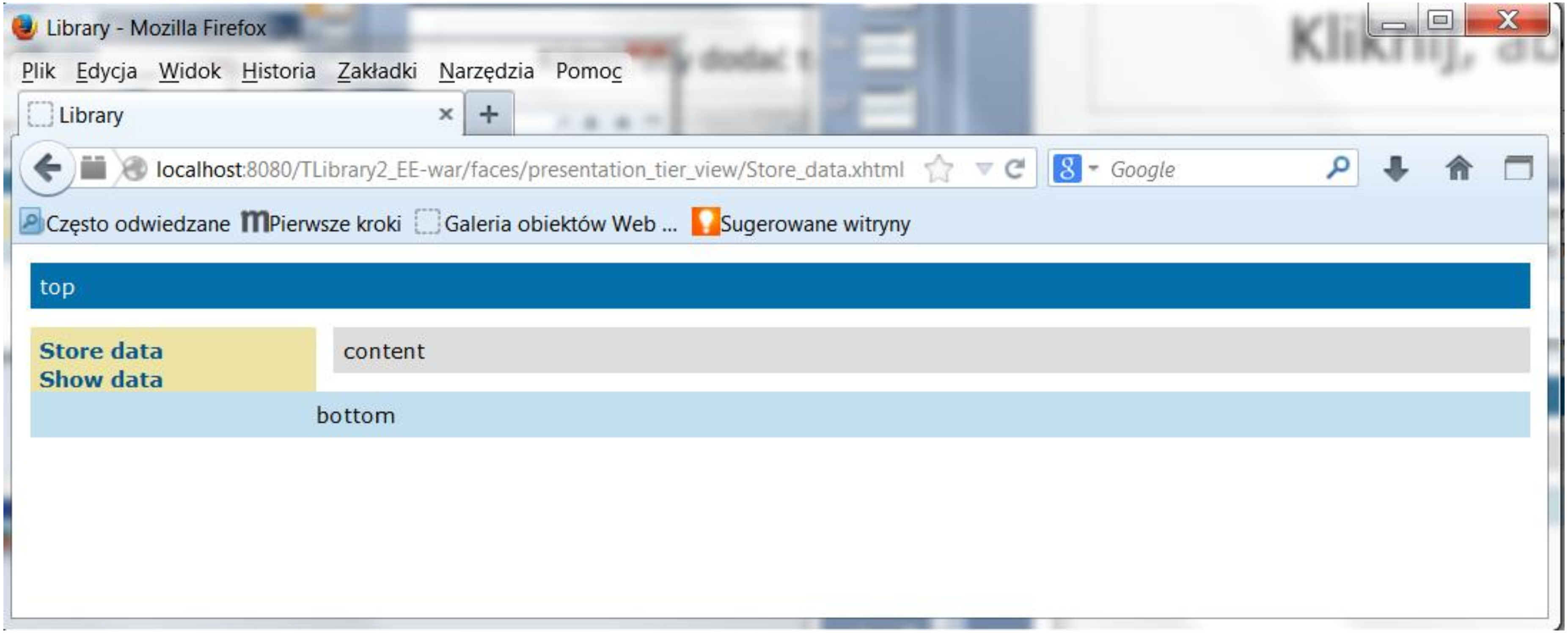
Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 11

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

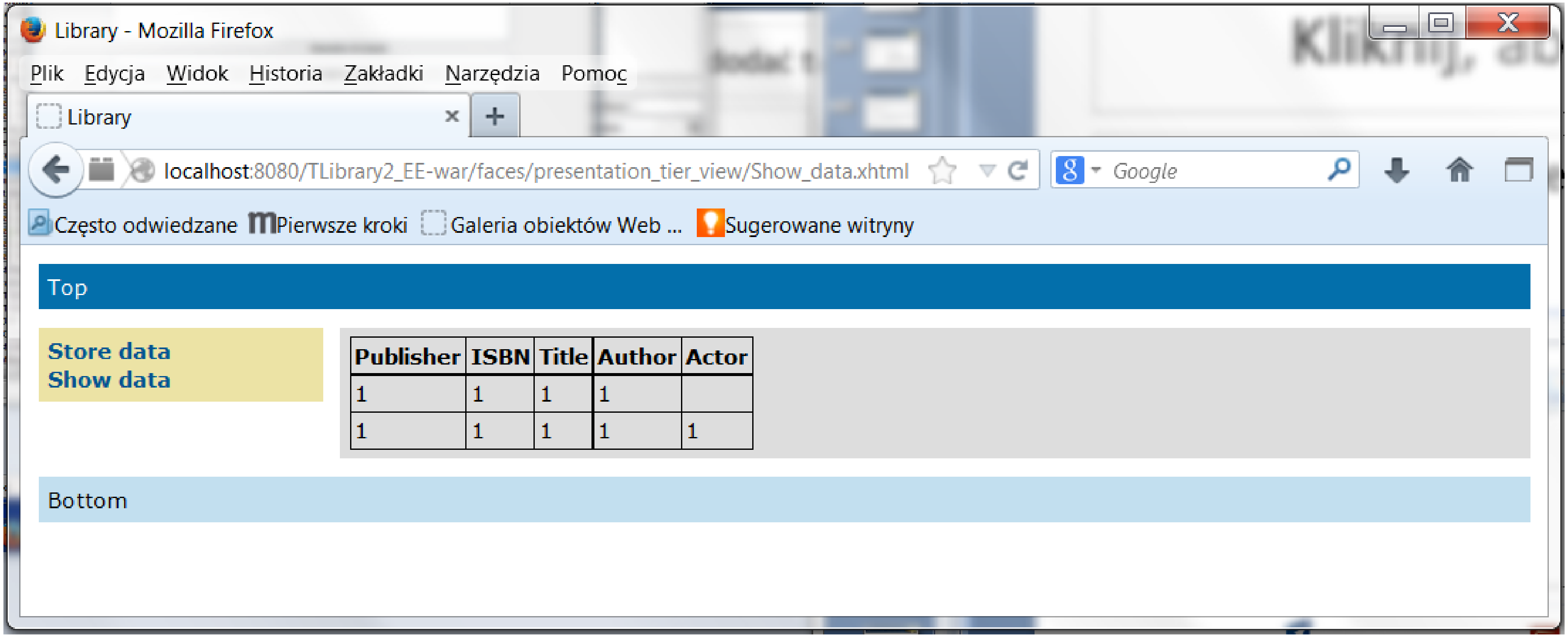

Library - Mozilla Firefox
Plik Edycja Widok Historia Zakładki Narzędzia Pomoc
Library
Search
localhost:8080/TLibrary2_EE-war/faces/presentation_tier_vie
Google
Często odwiedzane Pierwsze kroki Galeria obiektów Web ... Sugerowane witryny
Top
Store data
Show data
Store data
Bottom

Project co-financed from the EU European Social Fund

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


Library - Mozilla Firefox
Plik Edycja Widok Historia Zakładki Narzędzia Pomoc
Library
localhost:8080/TLibrary2_EE-war/faces/presentation_tier_view/Store_data.xhtml
Google
Często odwiedzane Pierwsze kroki Galeria obiektów Web ... Sugerowane witryny
top
Store data
Show data
content
bottom

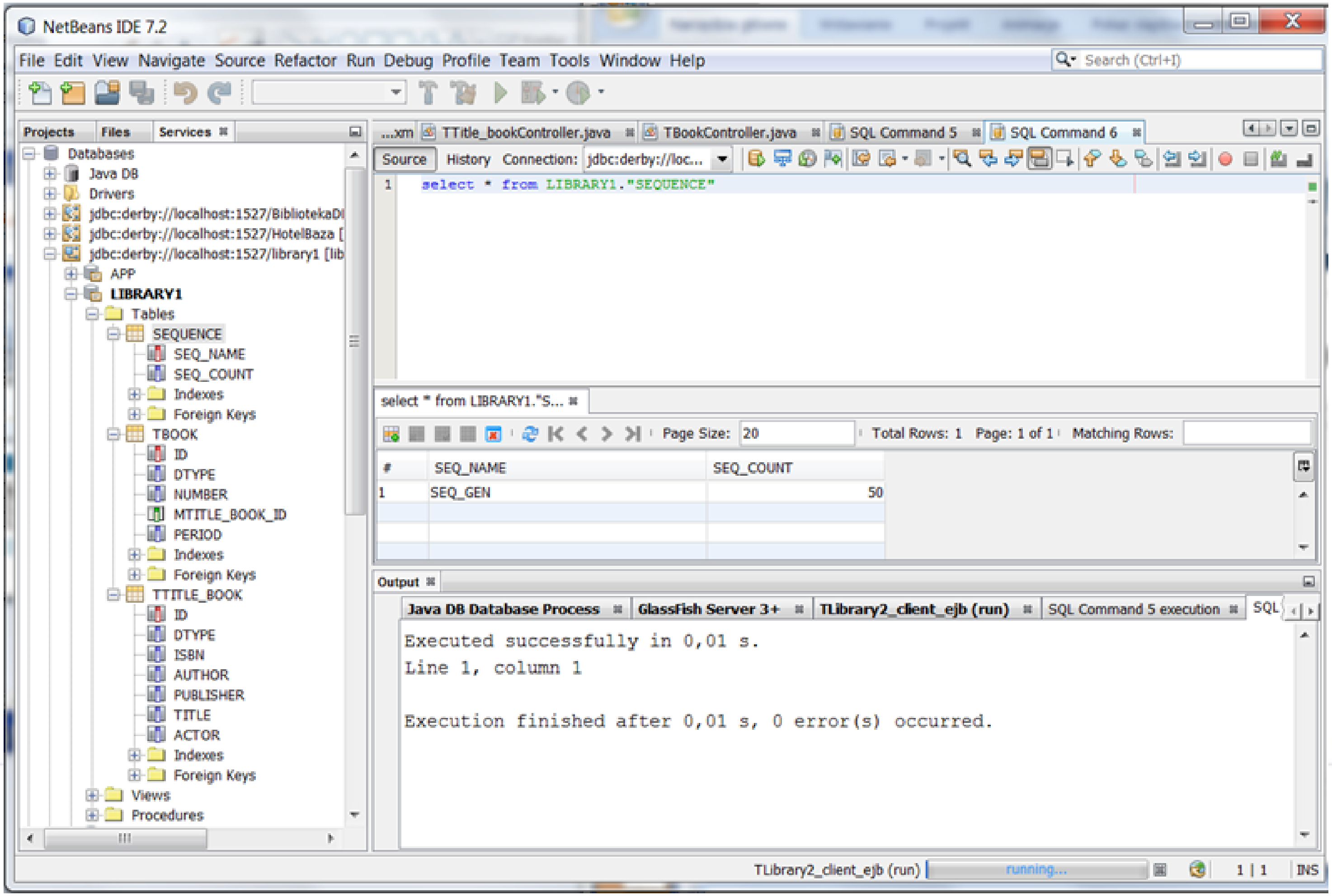
7.11. The view after choosing the Show data button (the left part of page) – i.e. after getting data from the database by the **TLibrary2_EE-war** client


Library - Mozilla Firefox
Plik Edycja Widok Historia Zakładki Narzędzia Pomoc
Library
localhost:8080/TLibrary2_EE-war/faces/presentation_tier_view/Show_data.xhtml
Google
Często odwiedzane Pierwsze kroki Galeria obiektów Web ... Sugerowane witryny
Top
Store data
Show data
Bottom

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

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

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)

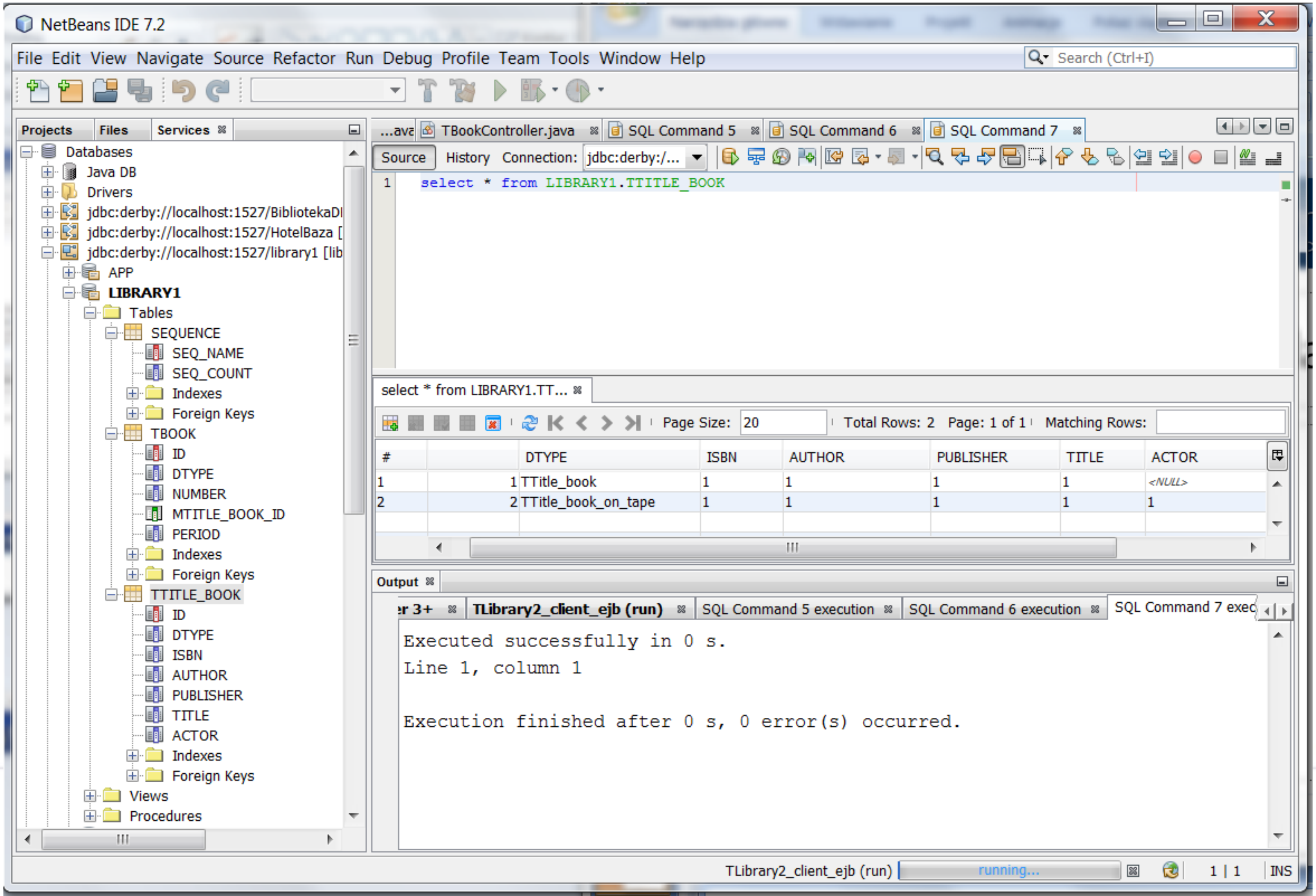


```
select * from LIBRARY1."SEQUENCE"
```

#	SEQ_NAME	SEQ_COUNT
1	SEQ_GEN	50

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

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

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)

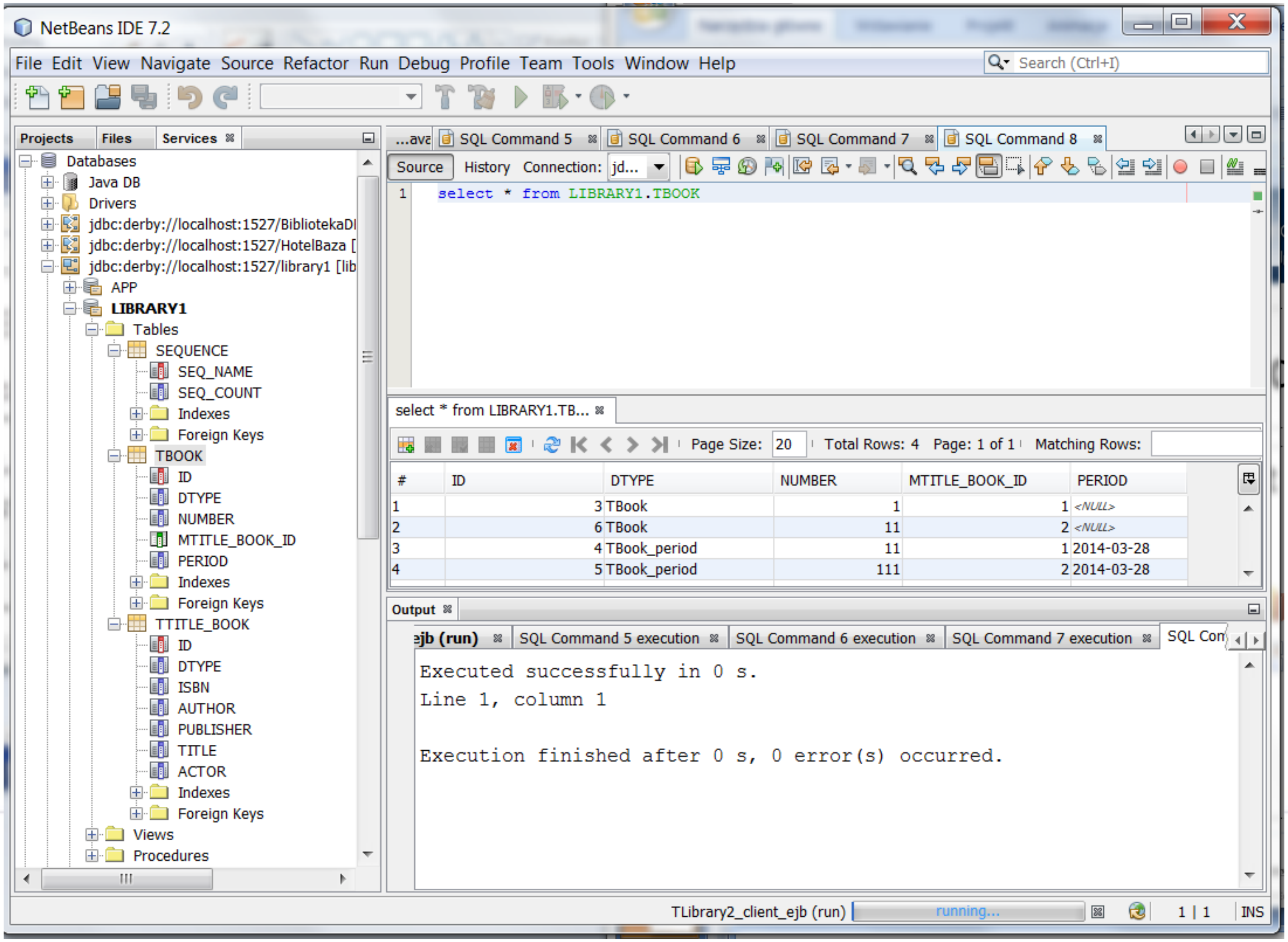


```
select * from LIBRARY1.TTITLE_BOOK
```

#		DTYPE	ISBN	AUTHOR	PUBLISHER	TITLE	ACTOR
1	1	TTitle_book	1	1	1	1	<NULL>
2	2	TTitle_book_on_tape	1	1	1	1	1

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

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)

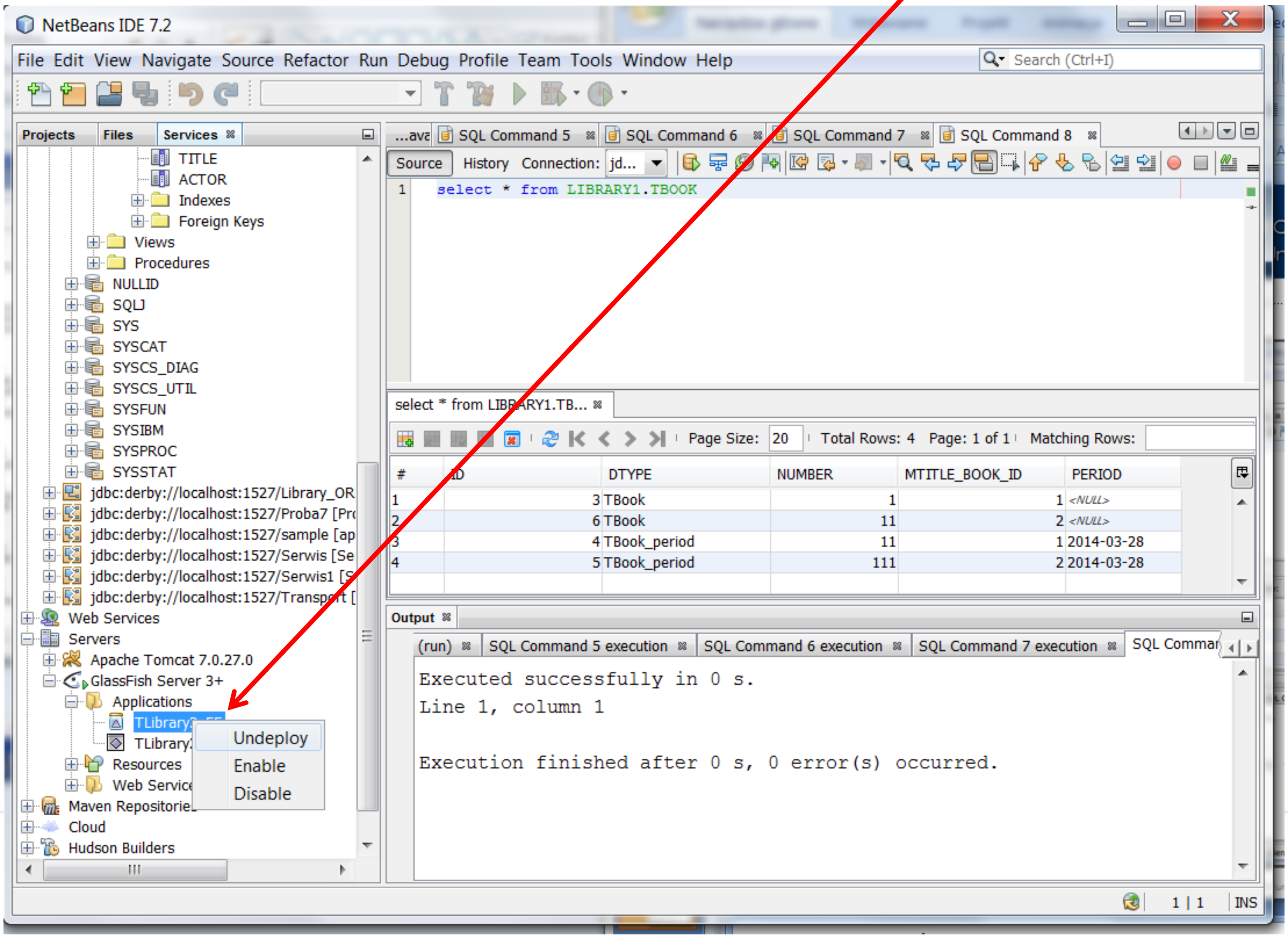

NetBeans IDE 7.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Files Services
Databases
Java DB
Drivers
jdbc:derby://localhost:1527/BibliotekaDI
jdbc:derby://localhost:1527/HotelBaza [
jdbc:derby://localhost:1527/library1 [lib
APP
LIBRARY1
Tables
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
...ava SQL Command 5 SQL Command 6 SQL Command 7 SQL Command 8
Source History Connection: jd...
select * from LIBRARY1.TB...
Page Size: 20 Total Rows: 4 Page: 1 of 1 Matching Rows:
Output
ejb (run) SQL Command 5 execution SQL Command 6 execution SQL Command 7 execution SQL Con
TLibrary2_client_ejb (run) running... 1 | 1 INS

```
select * from LIBRARY1.TBOOK
```

#	ID	DTYPE	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

```
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



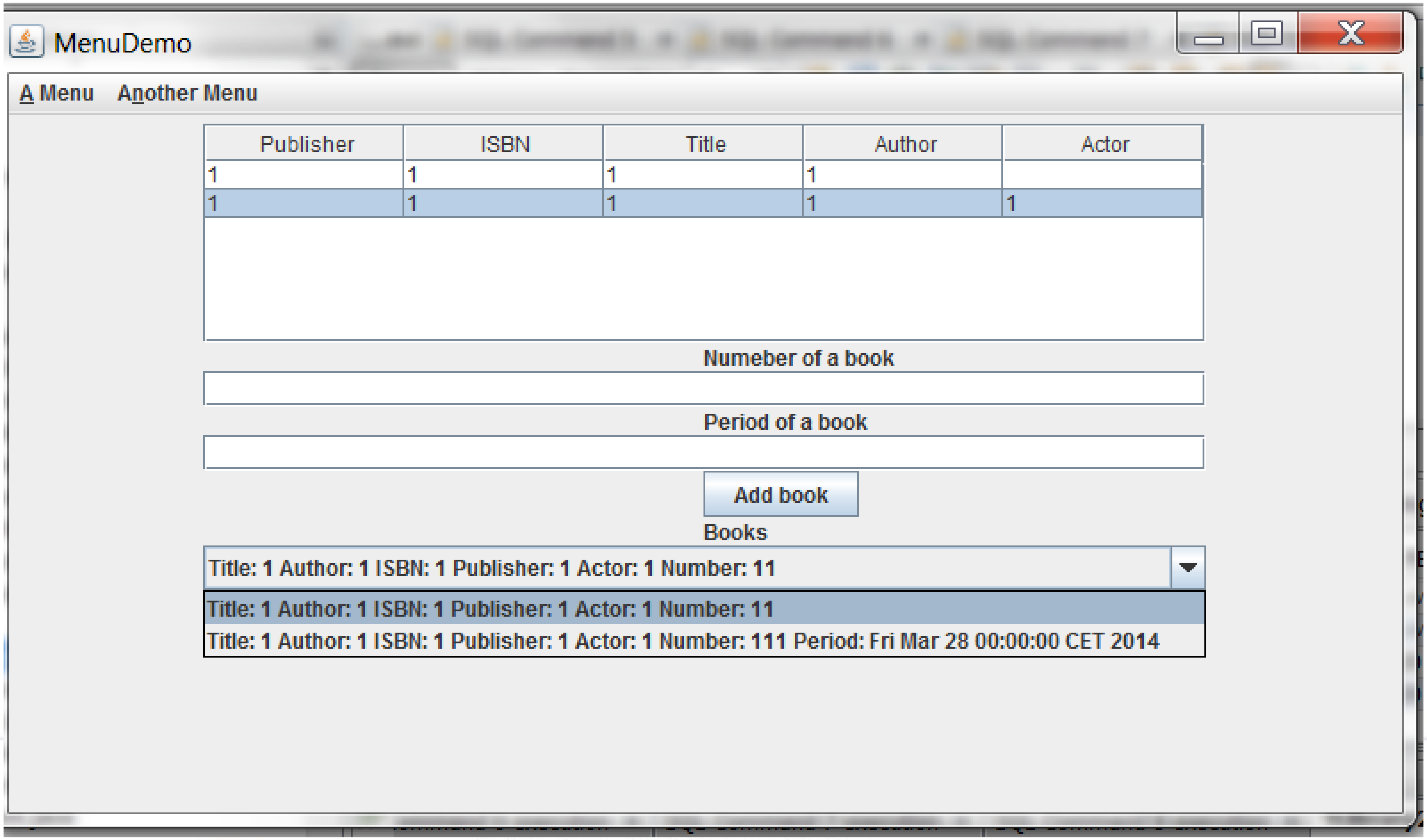
```
select * from LIBRARY1.TBOOK
```

select * from LIBRARY1.TB...

#	ID	DTYPE	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

```
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



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

Numeber of a book

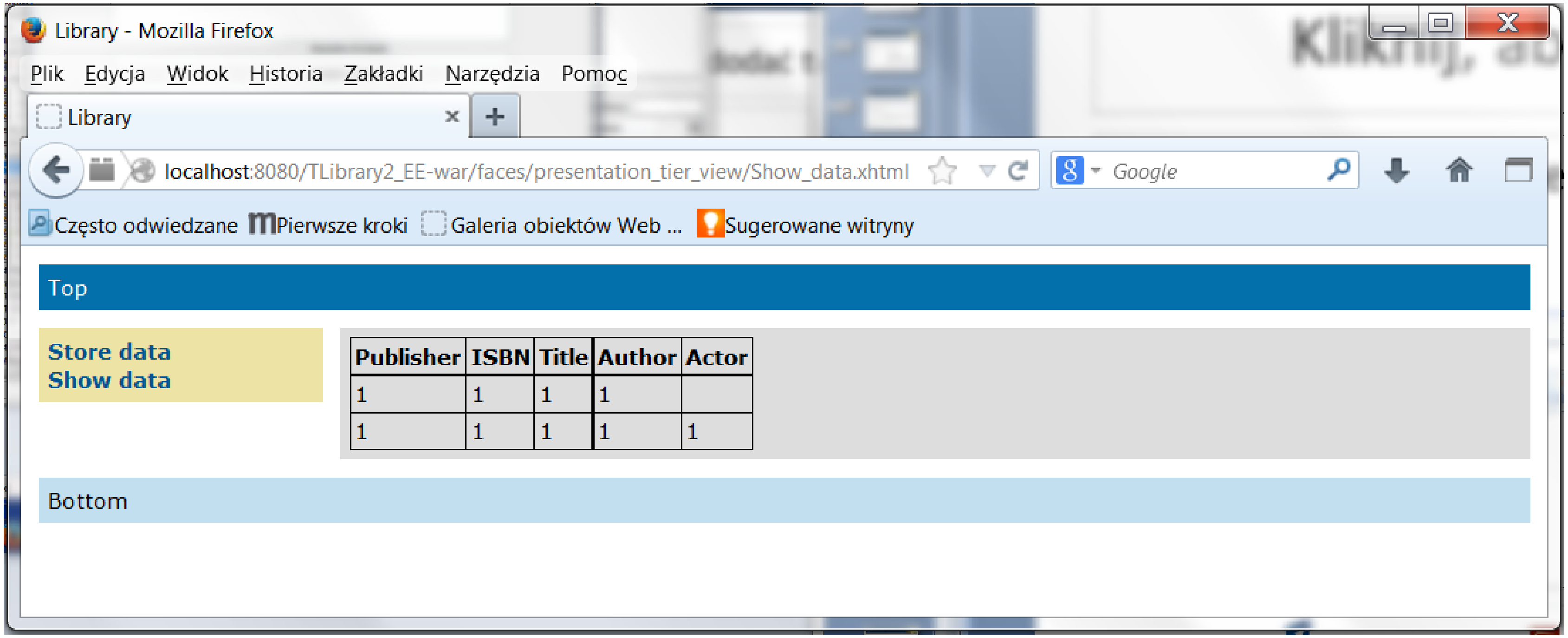
Period of a book

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.



Top

Store data
Show data

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

Bottom